Selfish Mining in Ethereum

by Cyril Grunspan

École Supérieure Ingénieurs Léonard de Vinci
12 Avenue Léonard de Vinci, 92400 Courbevoie

De Vinci Research Center

Email: `cyril.grunspan@devinci.fr`
Web: `cyrilgrunspan.fr`

June 13, 2019

Joint work with Ricardo Pérez-Marco

Talk based on the following articles

Talk based on the following articles

Selfish Mining in Ethereum (2019: arXiv:1904.13330), *Selfish Mining and Dyck Words in Bitcoin and Ethereum Networks* (proceedings of Tokenomics 2019 conference, ENS, arXiv:1904.07675)

Talk based on the following articles

Selfish Mining in Ethereum (2019: arXiv:1904.13330), *Selfish Mining and Dyck Words in Bitcoin and Ethereum Networks* (proceedings of Tokenomics 2019 conference, ENS, arXiv:1904.07675)

Foundational article: *On profitability of selfish mining* (2018)

Talk based on the following articles

Selfish Mining in Ethereum (2019: arXiv:1904.13330), *Selfish Mining and Dyck Words in Bitcoin and Ethereum Networks* (proceedings of Tokenomics 2019 conference, ENS, arXiv:1904.07675)

Foundational article: *On profitability of selfish mining* (2018)

Other articles:

On profitability of stubborn mining (2018), *On Profitability of Trailing Mining* (2018), *Bitcoin Selfish Mining and Dyck Words* (2019), *Bitcoin Selfish Mining and Dyck Words* (2019) All available on arxiv.org

Talk based on the following articles

Selfish Mining in Ethereum (2019: arXiv:1904.13330), *Selfish Mining and Dyck Words in Bitcoin and Ethereum Networks* (proceedings of Tokenomics 2019 conference, ENS, arXiv:1904.07675)

Foundational article: *On profitability of selfish mining* (2018)

Other articles:

On profitability of stubborn mining (2018), *On Profitability of Trailing Mining* (2018), *Bitcoin Selfish Mining and Dyck Words* (2019), *Bitcoin Selfish Mining and Dyck Words* (2019) All available on arxiv.org

Here: mathematical articles with theorems and proofs! Be careful with "well known results" with no proof...

Talk based on the following articles

Selfish Mining in Ethereum (2019: arXiv:1904.13330), *Selfish Mining and Dyck Words in Bitcoin and Ethereum Networks* (proceedings of Tokenomics 2019 conference, ENS, arXiv:1904.07675)

Foundational article: *On profitability of selfish mining* (2018)

Other articles:

On profitability of stubborn mining (2018), *On Profitability of Trailing Mining* (2018), *Bitcoin Selfish Mining and Dyck Words* (2019), *Bitcoin Selfish Mining and Dyck Words* (2019) All available on arxiv.org

Here: mathematical articles with theorems and proofs! Be careful with "well known results" with no proof...

Online mining simulators for Bitcoin exist and confirm our study

1 Bitcoin Protocol

2 Bitcoin Protocol

Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, October 31th 2008

3 Bitcoin Protocol

Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, October 31th 2008

1. New transactions are broadcast to all nodes.

4 Bitcoin Protocol

Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, October 31th 2008

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.

5 Bitcoin Protocol

Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, October 31th 2008

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.

6 Bitcoin Protocol

Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, October 31th 2008

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. **When a node finds a proof-of-work, it broadcasts the block to all nodes.**

7 Bitcoin Protocol

Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, October 31th 2008

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. **When a node finds a proof-of-work, it broadcasts the block to all nodes.**
5. Nodes accept the block only if all transactions in it are valid and not already spent.

8 Bitcoin Protocol

Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, October 31th 2008

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. **When a node finds a proof-of-work, it broadcasts the block to all nodes.**
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

9 Bitcoin Protocol

Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, October 31th 2008

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. **When a node finds a proof-of-work, it broadcasts the block to all nodes.**
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it

10 Bitcoin Protocol

Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, October 31th 2008

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. **When a node finds a proof-of-work, it broadcasts the block to all nodes.**
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it

A miner should never mine secretly and never withholds his blocks

11 Genesis

12 Genesis

Old question in the Bitcoin community

13 Genesis

Old question in the Bitcoin community

RHorning, *Mining cartel attack*, Bitcointalk forum, December 12, 2010

14 Genesis

Old question in the Bitcoin community

RHorning, *Mining cartel attack*, Bitcointalk forum, December 12, 2010

A cartel that would only recognize blocks generated by each other.

15 Genesis

Old question in the Bitcoin community

RHorning, *Mining cartel attack*, Bitcointalk forum, December 12, 2010

A cartel that would only recognize blocks generated by each other.

Meni Rosenfeld, *Analysis of Bitcoin Pooled Mining Reward Systems*, technical report, December 22, 2011

16 Genesis

Old question in the Bitcoin community

RHorning, *Mining cartel attack*, Bitcointalk forum, December 12, 2010

A cartel that would only recognize blocks generated by each other.

Meni Rosenfeld, *Analysis of Bitcoin Pooled Mining Reward Systems*, technical report, December 22, 2011

Ittay Eyal, Emin Gun Sirer, Majority is not Enough: Bitcoin Mining is Vulnerable, November 1 2013, Financial Cryptography 2014

17 Genesis

Old question in the Bitcoin community

RHorning, *Mining cartel attack*, Bitcointalk forum, December 12, 2010

A cartel that would only recognize blocks generated by each other.

Meni Rosenfeld, *Analysis of Bitcoin Pooled Mining Reward Systems*, technical report, December 22, 2011

Ittay Eyal, Emin Gun Sirer, Majority is not Enough: Bitcoin Mining is Vulnerable, November 1 2013, Financial Cryptography 2014

Lear Bahack, *Theoretical Bitcoin Attacks with less than Half of the Computational Power*, December 25 2013, technical report

18 Model Parameters

19 Model Parameters

Selfish Miner S with relative hashrate $q < \frac{1}{2}$

20 Model Parameters

Selfish Miner S with relative hashrate $q < \frac{1}{2}$

Honest miners H with relative hashrate $p = 1 - q$

21 Model Parameters

Selfish Miner S with relative hashrate $q < \frac{1}{2}$

Honest miners H with relative hashrate $p = 1 - q$

New parameter γ: connectivity of the attacker

22 Model Parameters

Selfish Miner S with relative hashrate $q < \frac{1}{2}$

Honest miners H with relative hashrate $p = 1 - q$

New parameter γ: connectivity of the attacker

Fraction of honest miners following the selfish miner

23 Model Parameters

Selfish Miner S with relative hashrate $q < \frac{1}{2}$

Honest miners H with relative hashrate $p = 1 - q$

New parameter γ: connectivity of the attacker

Fraction of honest miners following the selfish miner

In case of a public competition between a honest block and a selfish block, there are **3 outcomes**.

24 Model Parameters

Selfish Miner S with relative hashrate $q < \frac{1}{2}$

Honest miners H with relative hashrate $p = 1 - q$

New parameter γ: connectivity of the attacker

Fraction of honest miners following the selfish miner

In case of a public competition between a honest block and a selfish block, there are **3 outcomes**.

The winner can be:

25 Model Parameters

Selfish Miner S with relative hashrate $q < \frac{1}{2}$

Honest miners H with relative hashrate $p = 1 - q$

New parameter γ: connectivity of the attacker

Fraction of honest miners following the selfish miner

In case of a public competition between a honest block and a selfish block, there are **3 outcomes**.

The winner can be:

- The attacker (with probability q)

26 Model Parameters

Selfish Miner S with relative hashrate $q < \frac{1}{2}$

Honest miners H with relative hashrate $p = 1 - q$

New parameter γ: connectivity of the attacker

Fraction of honest miners following the selfish miner

In case of a public competition between a honest block and a selfish block, there are **3 outcomes**.

The winner can be:

- The attacker (with probability q)
- A honest miner who has mined a block on top of the attacker's block (with probability γp)

27 Model Parameters

Selfish Miner S with relative hashrate $q < \frac{1}{2}$

Honest miners H with relative hashrate $p = 1 - q$

New parameter γ: connectivity of the attacker

Fraction of honest miners following the selfish miner

In case of a public competition between a honest block and a selfish block, there are **3 outcomes**.

The winner can be:

- The attacker (with probability q)
- A honest miner who has mined a block on top of the attacker's block (with probability γp)
- A honest miner who has mined a block on top of the honest block (with probability $(1 - \gamma)\, p$)

28 Description of the Strategy

1. S mines on top of the last block of the official blockchain

29 Description of the Strategy

1. S mines on top of the last block of the official blockchain
2. If H is first to validate a block, then S goes back to 1 (end of a **cycle**).

30 Description of the Strategy

1. S mines on top of the last block of the official blockchain
2. If H is first to validate a block, then S goes back to 1 (end of a **cycle**).
3. If S is first to validate a block, then S keeps on mining **secretly** on top of her secret block

31 Description of the Strategy

1. S mines on top of the last block of the official blockchain
2. If H is first to validate a block, then S goes back to 1 (end of a **cycle**).
3. If S is first to validate a block, then S keeps on mining **secretly** on top of her secret block
4. If S is first to validate a block but then H mines one block before S validates a second one, S broadcasts immediately her secret block. A competition follows. After resolution of this competition, S goes back to 1 (end of a **cycle**).

32 Description of the Strategy

1. S mines on top of the last block of the official blockchain
2. If H is first to validate a block, then S goes back to 1 (end of a **cycle**).
3. If S is first to validate a block, then S keeps on mining **secretly** on top of her secret block
4. If S is first to validate a block but then H mines one block before S validates a second one, S broadcasts immediately her secret block. A competition follows. After resolution of this competition, S goes back to 1 (end of a **cycle**).
5. If S mines two blocks in a row then, S keeps on mining secretly on top of her secret fork

33 Description of the Strategy

1. S mines on top of the last block of the official blockchain
2. If H is first to validate a block, then S goes back to 1 (end of a **cycle**).
3. If S is first to validate a block, then S keeps on mining **secretly** on top of her secret block
4. If S is first to validate a block but then H mines one block before S validates a second one, S broadcasts immediately her secret block. A competition follows. After resolution of this competition, S goes back to 1 (end of a **cycle**).
5. If S mines two blocks in a row then, S keeps on mining secretly on top of her secret fork
6. When the advance of S reduces to 1, S broadcasts her entire fork (end of a **cycle**).

34 Description of the Strategy

1. S mines on top of the last block of the official blockchain
2. If H is first to validate a block, then S goes back to 1 (end of a **cycle**).
3. If S is first to validate a block, then S keeps on mining **secretly** on top of her secret block
4. If S is first to validate a block but then H mines one block before S validates a second one, S broadcasts immediately her secret block. A competition follows. After resolution of this competition, S goes back to 1 (end of a **cycle**).
5. If S mines two blocks in a row then, S keeps on mining secretly on top of her secret fork
6. When the advance of S reduces to 1, S broadcasts her entire fork (end of a **cycle**).
7. (optional **for Bitcoin**) If the advance of S is greater than 2, then each time H mines a block, S broadcasts immediately the part of her fork sharing the same height as the official blockchain

35 A state machine approach

36 A state machine approach

Strategy with last optional point (7).

37 A state machine approach

Strategy with last optional point (7).

Modelization of the advance of the attack with the help of a Markov chain (almost a simple random walk on $\mathbb{N}$ with a partial reflexive bound at 0).

38 A state machine approach

Strategy with last optional point (7).

Modelization of the advance of the attack with the help of a Markov chain (almost a simple random walk on $\mathbb{N}$ with a partial reflexive bound at 0).

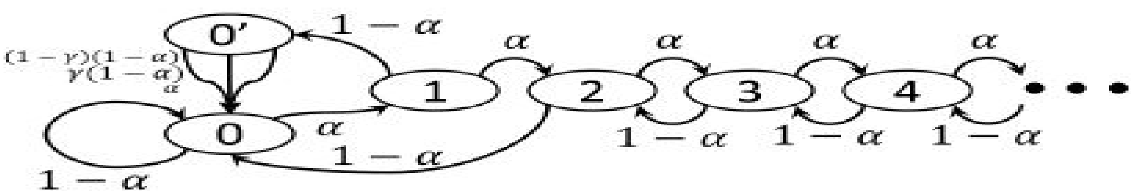


Fig. 1: State machine with transition frequencies.

39 A state machine approach

Strategy with last optional point (7).

Modelization of the advance of the attack with the help of a Markov chain (almost a simple random walk on $\mathbb{N}$ with a partial reflexive bound at 0).

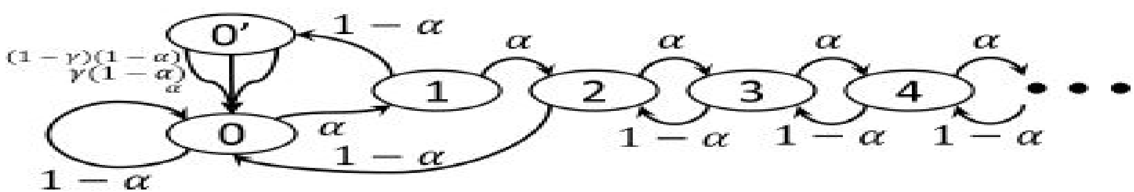


Fig. 1: State machine with transition frequencies.

Each transition gives a reward π for the honest miners and π' for the attacker. These are rewards that the honest miners or the selfish miner will **eventually** earn (possibly not immediatly).

40 Long Term Apparent hashrate

41 Long Term Apparent hashrate

Definition 4. *Let q' be the mean number of blocks mined by the attacker in the blockchain.*

42 Long Term Apparent hashrate

Definition 7. *Let q' be the mean number of blocks mined by the attacker in the blockchain.*

Lemma 8. *We have $q' = \frac{\mathbb{E}[\pi']}{\mathbb{E}[\pi] + \mathbb{E}[\pi']}$ where the probability here is the stationary probability.*

43 Long Term Apparent hashrate

Definition 10. *Let q' be the mean number of blocks mined by the attacker in the blockchain.*

Lemma 11. *We have $q' = \frac{\mathbb{E}[\pi']}{\mathbb{E}[\pi] + \mathbb{E}[\pi']}$ where the probability here is the stationary probability.*

Proof. Strong law of numbers ($\mathbb{E}[\pi] < +\infty, \mathbb{E}[\pi'] < +\infty$):

$$q' = \lim_{n\to\infty} \frac{\pi'_1 + \cdot \quad + \pi'_n}{\pi_1 + \cdot \quad + \pi_n + \pi'_1 + \cdot \quad + \pi'_n} = \lim_{n\to\infty} \frac{\frac{\pi'_1 + \cdot \quad + \pi'_n}{n}}{\frac{\pi_1 + \cdot \quad + \pi_n}{n} + \frac{\pi'_1 + \cdot \quad + \pi'_n}{n}}$$

□

44 Long Term Apparent hashrate

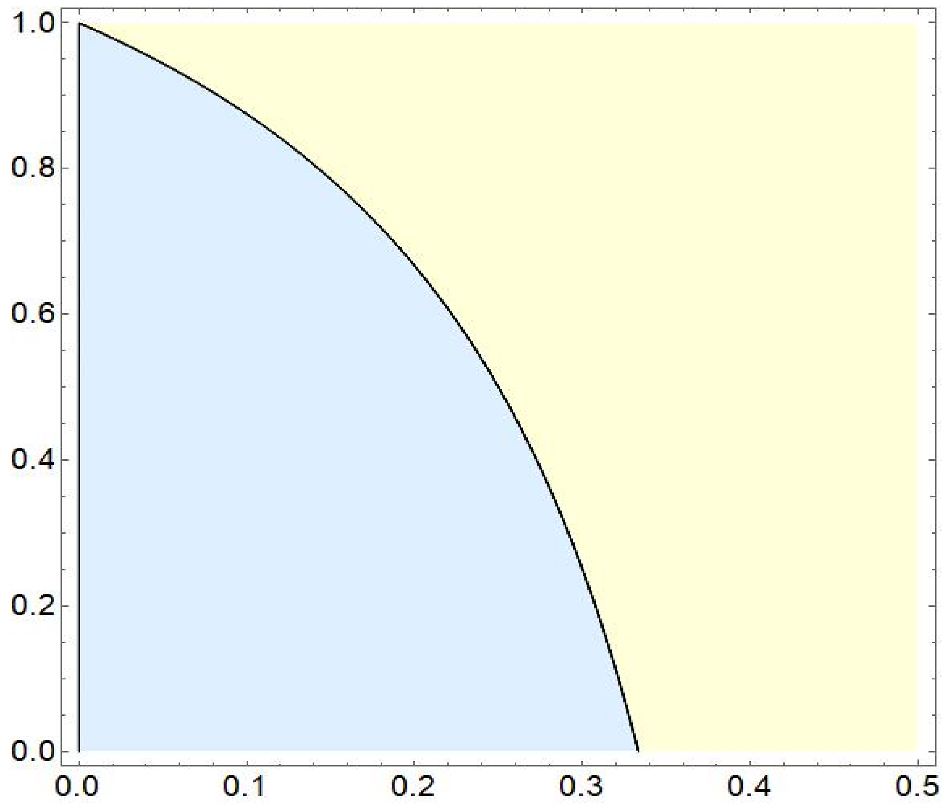
Definition 13. *Let q' be the mean number of blocks mined by the attacker in the blockchain.*

Lemma 14. *We have $q' = \frac{\mathbb{E}[\pi']}{\mathbb{E}[\pi] + \mathbb{E}[\pi']}$ where the probability here is the stationary probability.*

Proof. Strong law of numbers ($\mathbb{E}[\pi] < +\infty, \mathbb{E}[\pi'] < +\infty$):

$$q' = \lim_{n\to\infty} \frac{\pi'_1 + \cdot \quad + \pi'_n}{\pi_1 + \cdot \quad + \pi_n + \pi'_1 + \cdot \quad + \pi'_n} = \lim_{n\to\infty} \frac{\frac{\pi'_1 + \cdot \quad + \pi'_n}{n}}{\frac{\pi_1 + \cdot \quad + \pi_n}{n} + \frac{\pi'_1 + \cdot \quad + \pi'_n}{n}} \qquad \square$$

Theorem 15. *We have $q' = \frac{[(p-q)(1+p\,q) + p\,q]\,q - (p-q)\,p^2\,q\,(1-\gamma)}{p\,q^2 + p - q}$*



HM (blue) and SM (yellow) . X-axis: q, Y-axis: γ

Figure 1.

45 “Bitcoin is broken”?

46 "Bitcoin is broken"?

Eyal-Sirer (2013): in case of a competition, instead of the "first seen rule", nodes should broadcast randomly between two blocks sharing the same height: $\gamma = \frac{1}{2}$ always.

47 "Bitcoin is broken"?

Eyal-Sirer (2013): in case of a competition, instead of the "first seen rule", nodes should broadcast randomly between two blocks sharing the same height: $\gamma = \frac{1}{2}$ always.

Wrong solution as proved in *Optimal Selfish Mining Strategies in Bitcoin*, Ayelet Sapirshtein, Yonatan Sompolinsky, Aviv Zohar, Financial Cryptography 2016 : amplify the attack by a miner with low connectivity

48 "Bitcoin is broken"?

Eyal-Sirer (2013): in case of a competition, instead of the "first seen rule", nodes should broadcast randomly between two blocks sharing the same height: $\gamma = \frac{1}{2}$ always.

Wrong solution as proved in *Optimal Selfish Mining Strategies in Bitcoin*, Ayelet Sapirshtein, Yonatan Sompolinsky, Aviv Zohar, Financial Cryptography 2016 : amplify the attack by a miner with low connectivity

Ethan Heilman (2014) resorts to a non-decentralized timestamp server, *One Weird Trick to Stop Selfish Miners: Fresh Bitcoins, A Solution for the Honest Miner*, Financial Cryptography 2014.

49 "Bitcoin is broken"?

Eyal-Sirer (2013): in case of a competition, instead of the "first seen rule", nodes should broadcast randomly between two blocks sharing the same height: $\gamma = \frac{1}{2}$ always.

Wrong solution as proved in *Optimal Selfish Mining Strategies in Bitcoin*, Ayelet Sapirshtein, Yonatan Sompolinsky, Aviv Zohar, Financial Cryptography 2016 : amplify the attack by a miner with low connectivity

Ethan Heilman (2014) resorts to a non-decentralized timestamp server, *One Weird Trick to Stop Selfish Miners: Fresh Bitcoins, A Solution for the Honest Miner*, Financial Cryptography 2014.

Controversy: Reality of Selfish Mining?

50 "Bitcoin is broken"?

Eyal-Sirer (2013): in case of a competition, instead of the "first seen rule", nodes should broadcast randomly between two blocks sharing the same height: $\gamma = \frac{1}{2}$ always.

Wrong solution as proved in *Optimal Selfish Mining Strategies in Bitcoin*, Ayelet Sapirshtein, Yonatan Sompolinsky, Aviv Zohar, Financial Cryptography 2016 : amplify the attack by a miner with low connectivity

Ethan Heilman (2014) resorts to a non-decentralized timestamp server, *One Weird Trick to Stop Selfish Miners: Fresh Bitcoins, A Solution for the Honest Miner*, Financial Cryptography 2014.

Controversy: Reality of Selfish Mining?

None understood that the root of the problem lies in the difficulty adjustment

51 "Bitcoin is broken"?

Eyal-Sirer (2013): in case of a competition, instead of the "first seen rule", nodes should broadcast randomly between two blocks sharing the same height: $\gamma = \frac{1}{2}$ always.

Wrong solution as proved in *Optimal Selfish Mining Strategies in Bitcoin*, Ayelet Sapirshtein, Yonatan Sompolinsky, Aviv Zohar, Financial Cryptography 2016 : amplify the attack by a miner with low connectivity

Ethan Heilman (2014) resorts to a non-decentralized timestamp server, *One Weird Trick to Stop Selfish Miners: Fresh Bitcoins, A Solution for the Honest Miner*, Financial Cryptography 2014.

Controversy: Reality of Selfish Mining?

None understood that the root of the problem lies in the difficulty adjustment

Because none considered the **good objective function** to decide between two strategies

52 Profit and Loss per unit of time

53 Profit and Loss per unit of time

Time considerations

54 Profit and Loss per unit of time

Time considerations

Quantity of interest: **profit and loss per unit of time**

55 Profit and Loss per unit of time

Time considerations

Quantity of interest: **profit and loss per unit of time**

Definition 19. *For any activity with duration time T, we set:*

$$\begin{aligned} \mathrm{PnL} &= R - C \\ \mathrm{PnL}_t &= \frac{R-C}{T} \end{aligned}$$

We set also

$$\mathrm{PnL}_\infty = \lim_{T\to\infty} \frac{R-C}{T}$$

56 Repetitive Games

57 Repetitive Games

Definition 25. *A repetitive strategy is made of repetition of cycles*

58 Repetitive Games

Definition 30. *A repetitive strategy is made of repetition of cycles*

Example 31. A gambler plays repeatedly to a game such as "Head and Tail"

59 Repetitive Games

Definition 35. *A repetitive strategy is made of repetition of cycles*

Example 36. A gambler plays repeatedly to a game such as "Head and Tail"

Definition 37. *We denote by R (resp. C, T) the revenue (resp. cost, duration time) per cycle. The revenue ratio Γ and the cost ratio Υ of an integrable strategy are $\Gamma = \frac{\mathbb{E}[R]}{\mathbb{E}[T]}$ and $\Upsilon = \frac{\mathbb{E}[C]}{\mathbb{E}[T]}$.*

60 Repetitive Games

Definition 40. *A repetitive strategy is made of repetition of cycles*

Example 41. A gambler plays repeatedly to a game such as "Head and Tail"

Definition 42. *We denote by R (resp. C, T) the revenue (resp. cost, duration time) per cycle. The revenue ratio Γ and the cost ratio Υ of an integrable strategy are $\Gamma = \frac{\mathbb{E}[R]}{\mathbb{E}[T]}$ and $\Upsilon = \frac{\mathbb{E}[C]}{\mathbb{E}[T]}$.*

Theorem 43. *For an integrable repetitive strategy, we have $\mathrm{PnL}_\infty = \frac{\mathbb{E}[R] - \mathbb{E}[C]}{\mathbb{E}[T]}$.*

61 Repetitive Games

Definition 45. *A repetitive strategy is made of repetition of cycles*

Example 46. A gambler plays repeatedly to a game such as "Head and Tail"

Definition 47. *We denote by R (resp. C, T) the revenue (resp. cost, duration time) per cycle. The revenue ratio Γ and the cost ratio Υ of an integrable strategy are $\Gamma = \frac{\mathbb{E}[R]}{\mathbb{E}[T]}$ and $\Upsilon = \frac{\mathbb{E}[C]}{\mathbb{E}[T]}$.*

Theorem 48. *For an integrable repetitive strategy, we have $\mathrm{PnL}_\infty = \frac{\mathbb{E}[R] - \mathbb{E}[C]}{\mathbb{E}[T]}$.*

Theorem 49. *Let ξ and ξ' be two strategy ξ' sharing the **same cost per unit of time** i.e., $\Upsilon(\xi) = \Upsilon(\xi')$. Then, ξ is less profitable than ξ' if and only if $\Gamma(\xi) < \Gamma(\xi')$*

62 Key observations

63 Key observations

- A deviant strategy ξ and the honest strategy ξ_H shares the same cost per unit of time:

$$\Upsilon(\xi) = \Upsilon(\xi_H)$$

64 Key observations

- A deviant strategy ξ and the honest strategy ξ_H shares the same cost per unit of time:

$$\Upsilon(\xi) = \Upsilon(\xi_H)$$

- The relation $\xi \prec \xi'$ is independent with the exchange rate BTC/USD

65 Key observations

- A deviant strategy ξ and the honest strategy ξ_H shares the same cost per unit of time:

$$\Upsilon(\xi) = \Upsilon(\xi_H)$$

- The relation $\xi \prec \xi'$ is independent with the exchange rate BTC/USD
- We have $\mathbb{E}[R] = \mathbb{E}[L] \cdot (b + \mathbb{E}[f])$ where L is the number of official blocs added to the official blockchain after an attack cycle, b is the coinbase and f is the (random) fees per block.

66 Key observations

- A deviant strategy ξ and the honest strategy ξ_H shares the same cost per unit of time:

$$\Upsilon(\xi) = \Upsilon(\xi_H)$$

- The relation $\xi \prec \xi'$ is independent with the exchange rate BTC/USD
- We have $\mathbb{E}[R] = \mathbb{E}[L] \cdot (b + \mathbb{E}[f])$ where L is the number of official blocs added to the official blockchain after an attack cycle, b is the coinbase and f is the (random) fees per block.
- So, we can assume that the coinbase includes fees: $b \leftarrow b + \mathbb{E}[f]$

67 Key observations

- A deviant strategy ξ and the honest strategy ξ_H shares the same cost per unit of time:

$$\Upsilon(\xi) = \Upsilon(\xi_H)$$

- The relation $\xi \prec \xi'$ is independent with the exchange rate BTC/USD
- We have $\mathbb{E}[R] = \mathbb{E}[L] \cdot (b + \mathbb{E}[f])$ where L is the number of official blocs added to the official blockchain after an attack cycle, b is the coinbase and f is the (random) fees per block.
- So, we can assume that the coinbase includes fees: $b \leftarrow b + \mathbb{E}[f]$
- The relation $\xi \prec \xi'$ is independent with the amount of fees per block.

68 Key observations

- A deviant strategy ξ and the honest strategy ξ_H shares the same cost per unit of time:

$$\Upsilon(\xi) = \Upsilon(\xi_H)$$

- The relation $\xi \prec \xi'$ is independent with the exchange rate BTC/USD
- We have $\mathbb{E}[R] = \mathbb{E}[L] \cdot (b + \mathbb{E}[f])$ where L is the number of official blocs added to the official blockchain after an attack cycle, b is the coinbase and f is the (random) fees per block.
- So, we can assume that the coinbase includes fees: $b \leftarrow b + \mathbb{E}[f]$
- The relation $\xi \prec \xi'$ is independent with the amount of fees per block.
- **The revenue ratio is the good notion to decide between two mining strategies**

69 Bitcoin's stability theorem

70 Bitcoin's stability theorem

Theorem 51. *Without a difficulty adjustment, the best strategy is the honest one.*

71 Bitcoin's stability theorem

Theorem 52. *Without a difficulty adjustment, the best strategy is the honest one.*

Proof. For $t \in \mathbb{R}_+$, we denote by $\boldsymbol{N}(t)$ resp. $\boldsymbol{N}'(t)$) the number of blocks validated by the honest miners (resp. attacker) between 0 and t.

Without a difficulty adjustment, $\boldsymbol{N}(t)$, (resp. $\boldsymbol{N}'(t)$) is a true Poisson process with parameter $\alpha = \frac{p}{\tau_0}$ (resp. $\alpha' = \frac{q}{\tau_0}$) and $R(t) \leqslant \boldsymbol{N}'(t)$.

For any integrable stopping time τ, $\boldsymbol{N}(\tau) - \alpha\tau$ (resp. $\boldsymbol{N}'(\tau) - \alpha\tau$) is a martingale.

Then, we apply Doob's theorem. We get $\frac{\mathbb{E}[R(\tau)]}{\mathbb{E}[\tau]} \leqslant q\frac{b}{\tau_0} = \Gamma(\mathrm{HM})$. □

72 Bitcoin's stability theorem

Theorem 53. *Without a difficulty adjustment, the best strategy is the honest one.*

Proof. For $t \in \mathbb{R}_+$, we denote by $\boldsymbol{N}(t)$ resp. $\boldsymbol{N}'(t)$) the number of blocks validated by the honest miners (resp. attacker) between 0 and t.

Without a difficulty adjustment, $\boldsymbol{N}(t)$, (resp. $\boldsymbol{N}'(t)$) is a true Poisson process with parameter $\alpha = \frac{p}{\tau_0}$ (resp. $\alpha' = \frac{q}{\tau_0}$) and $R(t) \leqslant \boldsymbol{N}'(t)$.

For any integrable stopping time τ, $\boldsymbol{N}(\tau) - \alpha\tau$ (resp. $\boldsymbol{N}'(\tau) - \alpha\tau$) is a martingale.

Then, we apply Doob's theorem. We get $\frac{\mathbb{E}[R(\tau)]}{\mathbb{E}[\tau]} \leqslant q\frac{b}{\tau_0} = \Gamma(\mathrm{HM})$. □

So, the problem lies in the difficulty adjustment formula

73 Bitcoin difficulty adjustment

74 Bitcoin difficulty adjustment

The difficulty adjustment in Bitcoin today is $D_{\text{new}} = D_{\text{old}} \cdot \frac{n_0 \tau_0}{\boldsymbol{S}_{n_0}}$ where $\boldsymbol{S}_{\boldsymbol{n_0}}$ is the time used to mine $n_0 = 2016$ blocks.

75 Bitcoin difficulty adjustment

The difficulty adjustment in Bitcoin today is $D_{\text{new}} = D_{\text{old}} \cdot \frac{n_0 \tau_0}{\boldsymbol{S}_{n_0}}$ where $\boldsymbol{S}_{\boldsymbol{n_0}}$ is the time used to mine $n_0 = 2016$ blocks.

Note 56. In reality, due to a well known bug, it is $D_{\text{new}} = D_{\text{old}} \cdot \frac{n_0 \tau_0}{\boldsymbol{S}_{n_0 - 1}}$. So, if there is no attacker and the difficulty parameter remains constant, the exact mean interblock time τ in Bitcoin is given by ($\frac{1}{\boldsymbol{S}_{n_0 - 1}}$ follows an inverse Gamma distribution):

$$1 = \frac{n_0 \tau_0}{(n_0 - 2)\tau}$$

i.e., $\tau = \tau_0 + \frac{2}{n_0 - 2} \tau_0 > \tau_0$ (inverse Gamma distribution)

76 Analysis of the problem

77 Analysis of the problem

- An attacker first slows down the progression of the blockchain: $\boldsymbol{S}_{n_0} > n_0 \, \tau_0$. So, $D_{\text{new}} < D_{\text{old}}$ and the speeds of validation are modified: $\alpha_{\text{new}} = \alpha_{\text{old}} \cdot \frac{D_{\text{old}}}{D_{\text{new}}}$ (resp. $\alpha'_{\text{new}} = \alpha'_{\text{old}} \cdot \frac{D_{\text{old}}}{D_{\text{new}}}$).

78 Analysis of the problem

- An attacker first slows down the progression of the blockchain: $\boldsymbol{S}_{n_0} > n_0\,\tau_0$. So, $D_{\text{new}} < D_{\text{old}}$ and the speeds of validation are modified: $\alpha_{\text{new}} = \alpha_{\text{old}} \cdot \frac{D_{\text{old}}}{D_{\text{new}}}$ (resp. $\alpha'_{\text{new}} = \alpha'_{\text{old}} \cdot \frac{D_{\text{old}}}{D_{\text{new}}}$).
- After a difficulty adjustment, the expected revenue per cycle $\mathbb{E}[R]$ is not modified but the mean time it takes $\mathbb{E}\left[T\right]$ is reduced by a factor $\frac{D_{\text{old}}}{D_{\text{new}}}$.

79 Analysis of the problem

- An attacker first slows down the progression of the blockchain: $\boldsymbol{S}_{n_0} > n_0\,\tau_0$. So, $D_{\text{new}} < D_{\text{old}}$ and the speeds of validation are modified: $\alpha_{\text{new}} = \alpha_{\text{old}} \cdot \frac{D_{\text{old}}}{D_{\text{new}}}$ (resp. $\alpha'_{\text{new}} = \alpha'_{\text{old}} \cdot \frac{D_{\text{old}}}{D_{\text{new}}}$).
- After a difficulty adjustment, the expected revenue per cycle $\mathbb{E}[R]$ is not modified but the mean time it takes $\mathbb{E}\,[T]$ is reduced by a factor $\frac{D_{\text{old}}}{D_{\text{new}}}$.
- The revenue ratio is multiplied by a factor $\frac{D_{\text{old}}}{D_{\text{new}}}$.

80 Analysis of the problem

- An attacker first slows down the progression of the blockchain: $\boldsymbol{S}_{n_0} > n_0\,\tau_0$. So, $D_{\text{new}} < D_{\text{old}}$ and the speeds of validation are modified: $\alpha_{\text{new}} = \alpha_{\text{old}} \cdot \frac{D_{\text{old}}}{D_{\text{new}}}$ (resp. $\alpha'_{\text{new}} = \alpha'_{\text{old}} \cdot \frac{D_{\text{old}}}{D_{\text{new}}}$).
- After a difficulty adjustment, the expected revenue per cycle $\mathbb{E}[R]$ is not modified but the mean time it takes $\mathbb{E}\left[T\right]$ is reduced by a factor $\frac{D_{\text{old}}}{D_{\text{new}}}$.
- The revenue ratio is multiplied by a factor $\frac{D_{\text{old}}}{D_{\text{new}}}$.
- The difficulty parameter changes dramatically whereas the total hashrate remains constant.

81 Analysis of the problem

- An attacker first slows down the progression of the blockchain: $\boldsymbol{S}_{n_0} > n_0\,\tau_0$. So, $D_{\text{new}} < D_{\text{old}}$ and the speeds of validation are modified: $\alpha_{\text{new}} = \alpha_{\text{old}} \cdot \frac{D_{\text{old}}}{D_{\text{new}}}$ (resp. $\alpha'_{\text{new}} = \alpha'_{\text{old}} \cdot \frac{D_{\text{old}}}{D_{\text{new}}}$).
- After a difficulty adjustment, the expected revenue per cycle $\mathbb{E}[R]$ is not modified but the mean time it takes $\mathbb{E}\left[T\right]$ is reduced by a factor $\frac{D_{\text{old}}}{D_{\text{new}}}$.
- The revenue ratio is multiplied by a factor $\frac{D_{\text{old}}}{D_{\text{new}}}$.
- The difficulty parameter changes dramatically whereas the total hashrate remains constant.
- Why that? The difficulty parameter should reflect the exact hashrate of the network

82 Analysis of the problem

- An attacker first slows down the progression of the blockchain: $\boldsymbol{S}_{n_0} > n_0\,\tau_0$. So, $D_{\text{new}} < D_{\text{old}}$ and the speeds of validation are modified: $\alpha_{\text{new}} = \alpha_{\text{old}} \cdot \frac{D_{\text{old}}}{D_{\text{new}}}$ (resp. $\alpha'_{\text{new}} = \alpha'_{\text{old}} \cdot \frac{D_{\text{old}}}{D_{\text{new}}}$).
- After a difficulty adjustment, the expected revenue per cycle $\mathbb{E}[R]$ is not modified but the mean time it takes $\mathbb{E}\left[T\right]$ is reduced by a factor $\frac{D_{\text{old}}}{D_{\text{new}}}$.
- The revenue ratio is multiplied by a factor $\frac{D_{\text{old}}}{D_{\text{new}}}$.
- The difficulty parameter changes dramatically whereas the total hashrate remains constant.
- Why that? The difficulty parameter should reflect the exact hashrate of the network
- It is not the case because the difficulty adjustment formula ignores orphan blocks.

83 Solution for thwarting selfish mining

84 Solution for thwarting selfish mining

Incorporate the count of orphan blocks in the difficulty adjustment formula:

85 Solution for thwarting selfish mining

Incorporate the count of orphan blocks in the difficulty adjustment formula:

$$D_{\text{new}} = D_{\text{old}} \cdot \frac{(n_0 + \boldsymbol{n'})\,\tau_0}{\boldsymbol{S}_{n_0}}$$

where $\boldsymbol{n'}$ is the number of orphan blocks during the last period of n_0 official blocks

86 Solution for thwarting selfish mining

Incorporate the count of orphan blocks in the difficulty adjustment formula:

$$D_{\text{new}} = D_{\text{old}} \cdot \frac{(n_0 + \boldsymbol{n'})\,\tau_0}{\boldsymbol{S}_{n_0}}$$

where $\boldsymbol{n'}$ is the number of orphan blocks during the last period of n_0 official blocks

Miners signal orphan blocks in official blocks.

87 Solution for thwarting selfish mining

Incorporate the count of orphan blocks in the difficulty adjustment formula:

$$D_{\text{new}} = D_{\text{old}} \cdot \frac{(n_0 + \boldsymbol{n'})\,\tau_0}{\boldsymbol{S}_{n_0}}$$

where $\boldsymbol{n'}$ is the number of orphan blocks during the last period of n_0 official blocks

Miners signal orphan blocks in official blocks.

Nodes relay headers of orphan blocks. Do not need to relay full orphan blocks.

88 Solution for thwarting selfish mining

Incorporate the count of orphan blocks in the difficulty adjustment formula:

$$D_{\text{new}} = D_{\text{old}} \cdot \frac{(n_0 + \boldsymbol{n'})\,\tau_0}{\boldsymbol{S}_{n_0}}$$

where $\boldsymbol{n'}$ is the number of orphan blocks during the last period of n_0 official blocks

Miners signal orphan blocks in official blocks.

Nodes relay headers of orphan blocks. Do not need to relay full orphan blocks.

Incentives for miners: in case of a competition, nodes relay the block with the most proof-of-work, including proof-of-work of orphan blocks.

89 Solution for thwarting selfish mining

Incorporate the count of orphan blocks in the difficulty adjustment formula:

$$D_{\text{new}} = D_{\text{old}} \cdot \frac{(n_0 + \boldsymbol{n'})\,\tau_0}{\boldsymbol{S}_{n_0}}$$

where $\boldsymbol{n'}$ is the number of orphan blocks during the last period of n_0 official blocks

Miners signal orphan blocks in official blocks.

Nodes relay headers of orphan blocks. Do not need to relay full orphan blocks.

Incentives for miners: in case of a competition, nodes relay the block with the most proof-of-work, including proof-of-work of orphan blocks.

BIP proposal

90 Martingale approach to compute the long-term apparent hashrate

- Compute $\mathbb{E}[T], \mathbb{E}[R]$ and $\frac{D_{\text{new}}}{D_{\text{old}}}$

91 Martingale approach to compute the long-term apparent hashrate

- Compute $\mathbb{E}[T], \mathbb{E}[R]$ and $\frac{D_{\text{new}}}{D_{\text{old}}}$
- Exercise on Poisson process theory (Poisson races)

92 Martingale approach to compute the long-term apparent hashrate

- Compute $\mathbb{E}[T], \mathbb{E}[R]$ and $\frac{D_{\text{new}}}{D_{\text{old}}}$
- Exercise on Poisson process theory (Poisson races)
- Gives the time it takes before the attack becomes profitable: information unreachable with other methods

93 Martingale approach to compute the long-term apparent hashrate

- Compute $\mathbb{E}[T], \mathbb{E}[R]$ and $\frac{D_{\text{new}}}{D_{\text{old}}}$
- Exercise on Poisson process theory (Poisson races)
- Gives the time it takes before the attack becomes profitable: information unreachable with other methods
- Example. With $q=0.1$ and $\gamma=0.9$, the attack takes 10 weeks on average before profitability

94 Martingale approach to compute the long-term apparent hashrate

- Compute $\mathbb{E}[T], \mathbb{E}[R]$ and $\frac{D_{\text{new}}}{D_{\text{old}}}$
- Exercise on Poisson process theory (Poisson races)
- Gives the time it takes before the attack becomes profitable: information unreachable with other methods
- Example. With $q=0.1$ and $\gamma=0.9$, the attack takes 10 weeks on average before profitability
- **After a difficulty adjustment, equivalence between the two methods**: $\mathbb{E}[T]=\mathbb{E}[L]\,\tau_0$ where L is the number of blocks added to the official blockchain per cycle. So, $\Gamma = q'\,\frac{b}{\tau_0}$

95 Martingale approach to compute the long-term apparent hashrate

- Compute $\mathbb{E}[T], \mathbb{E}[R]$ and $\frac{D_{\text{new}}}{D_{\text{old}}}$
- Exercise on Poisson process theory (Poisson races)
- Gives the time it takes before the attack becomes profitable: information unreachable with other methods
- Example. With $q=0.1$ and $\gamma=0.9$, the attack takes 10 weeks on average before profitability
- **After a difficulty adjustment, equivalence between the two methods**: $\mathbb{E}[T]=\mathbb{E}[L]\,\tau_0$ where L is the number of blocks added to the official blockchain per cycle. So, $\Gamma=q'\,\frac{b}{\tau_0}$
- Can be applied to other block witholding strategies: cf *On Profitability of Selfish Mining*, *On Profitability of Stubborn Mining*, *On Profitability of Trailing Mining*.

96 Martingale approach to compute the long-term apparent hashrate

- Compute $\mathbb{E}[T], \mathbb{E}[R]$ and $\frac{D_{\text{new}}}{D_{\text{old}}}$
- Exercise on Poisson process theory (Poisson races)
- Gives the time it takes before the attack becomes profitable: information unreachable with other methods
- Example. With $q=0.1$ and $\gamma=0.9$, the attack takes 10 weeks on average before profitability
- **After a difficulty adjustment, equivalence between the two methods**: $\mathbb{E}[T]=\mathbb{E}[L]\,\tau_0$ where L is the number of blocks added to the official blockchain per cycle. So, $\Gamma=q'\,\frac{b}{\tau_0}$
- Can be applied to other block witholding strategies: cf *On Profitability of Selfish Mining*, *On Profitability of Stubborn Mining*, *On Profitability of Trailing Mining*.
- Graph of profitability with other strategies

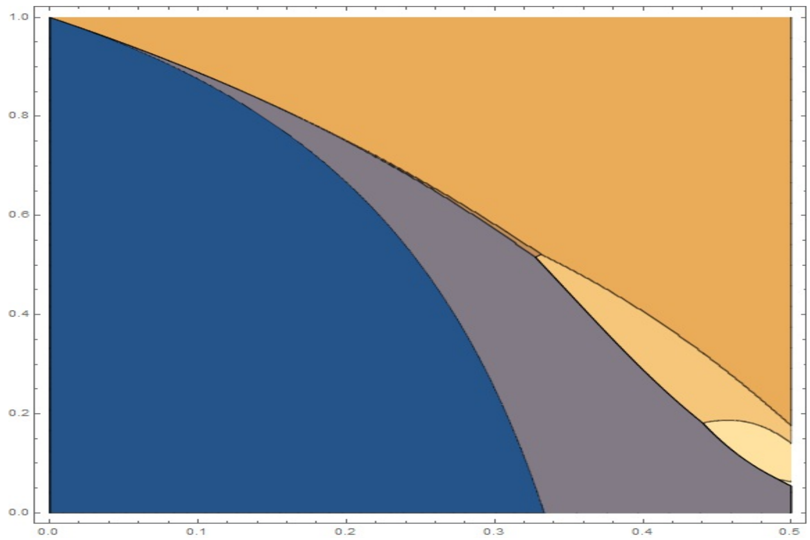


Figure 2. Comparing profitabilities of HM, SM, LSM, EFSM, A-TSM

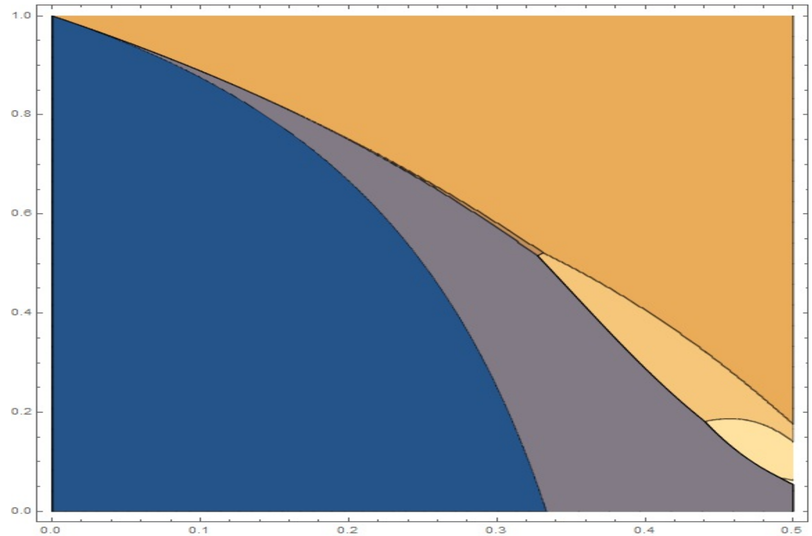


Figure 3. Comparing profitabilities of HM, SM, LSM, EFSM, A-TSM

Optimal strategy obtained by Zohar&al using a **black box solver** of Markov Decision Process

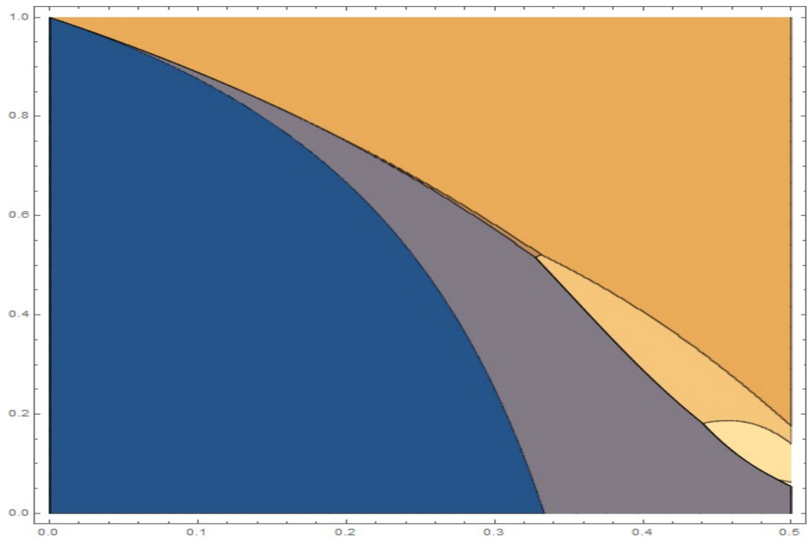


Figure 4. Comparing profitabilities of HM, SM, LSM, EFSM, A-TSM

Optimal strategy obtained by Zohar&al using a **black box solver** of Markov Decision Process

Analogous general study missing for Ethereum

97 A combinatorics approach

98 A combinatorics approach

Definition 60. *We denote by Z (resp. L) the number of blocks validated by the attacker (resp. the network) and added to the official blockchain per attack cycle.*

99 A combinatorics approach

Definition 63. *We denote by Z (resp. L) the number of blocks validated by the attacker (resp. the network) and added to the official blockchain per attack cycle.*

Proposition 64. *After a difficulty adjustment, we have $\Gamma = \frac{\mathbb{E}[Z]}{\mathbb{E}[L]} \cdot \frac{b}{\tau_0}$.*

100 A combinatorics approach

Definition 66. *We denote by* Z *(resp.* L*) the number of blocks validated by the attacker (resp. the network) and added to the official blockchain per attack cycle.*

Proposition 67. *After a difficulty adjustment, we have* $\Gamma = \frac{\mathbb{E}[Z]}{\mathbb{E}[L]} \cdot \frac{b}{\tau_0}$.

A cycle is described with the chronological sequence of discoveries S and H i.e. SS**SHSSHH**H

101 A combinatorics approach

Definition 69. *We denote by Z (resp. L) the number of blocks validated by the attacker (resp. the network) and added to the official blockchain per attack cycle.*

Proposition 70. *After a difficulty adjustment, we have $\Gamma = \frac{\mathbb{E}[Z]}{\mathbb{E}[L]} \cdot \frac{b}{\tau_0}$.*

A cycle is described with the chronological sequence of discoveries S and H i.e. SS**SHSSHH**H

Definition 71. *A Dyck word of length n built on $\{S, H\}$ is a string of S and H containing n S and n H and such that no initial segment of the string has more H's than S's. We denote by $\mathcal{D}_n$ the set of such words and by $\mathcal{D}$ the space of all Dyck words.*

Theorem 72. *The attack cycles of the selfish mining strategies are H, SHS, SHH and SSwH with* $w \in \mathcal{D}$.

Theorem 74. *The attack cycles of the selfish mining strategies are H, SHS, SHH and SSwH with* $w \in \mathcal{D}$.

Corollary 75. *We have* $\mathbb{P}[L=1] = p, \mathbb{P}[L=2] = p + p\,q^2$ *and* $\mathbb{P}[L=n] = p\,q^2\,(p\,q)^{n-2}\,C_{n-2}$ *for* $n > 2$ *with* $C_k = \frac{1}{k+1}\binom{2k}{k} = k$*-th Catalan number.*

Theorem 76. *The attack cycles of the selfish mining strategies are H, SHS, SHH and SSwH with* $w \in \mathcal{D}$.

Corollary 77. *We have* $\mathbb{P}[L=1]=p, \mathbb{P}[L=2]=p+p\,q^2$ *and* $\mathbb{P}[L=n]=p\,q^2\,(p\,q)^{n-2}\,C_{n-2}$ *for* $n>2$ *with* $C_k=\frac{1}{k+1}\binom{2k}{k}=k$*-th Catalan number.*

Similarly, we get the distribution of Z (note that for $n>2$, $[Z=n]=[L=n]$) and $\frac{\mathbb{E}[Z]}{\mathbb{E}[L]}$

Theorem 78. *The attack cycles of the selfish mining strategies are H, SHS, SHH and SSwH with* $w \in \mathcal{D}$.

Corollary 79. *We have* $\mathbb{P}[L=1]=p, \mathbb{P}[L=2]=p+p\,q^2$ *and* $\mathbb{P}[L=n]=p\,q^2\,(p\,q)^{n-2}\,C_{n-2}$ *for* $n>2$ *with* $C_k=\frac{1}{k+1}\binom{2k}{k}=k$*-th Catalan number.*

Similarly, we get the distribution of Z (note that for $n>2$, $[Z=n]=[L=n]$) and $\frac{\mathbb{E}[Z]}{\mathbb{E}[L]}$

We use:

$$\sum_{n\geqslant 0} p\,(p\,q)^n\,C_n = 1$$

$$\sum_{n\geqslant 0} p\,(p\,q)^n\,C_n = \frac{q}{p-q}$$

102 Ethereum network

103 Ethereum network

Interblock times τ_E reduced: between 13 and 14 sec today

104 Ethereum network

Interblock times τ_E reduced: between 13 and 14 sec today

More or less block propagation time

105 Ethereum network

Interblock times τ_E reduced: between 13 and 14 sec today

More or less block propagation time

A priori orphan blocks

106 Ethereum network

Interblock times τ_E reduced: between 13 and 14 sec today

More or less block propagation time

A priori orphan blocks

To decide between two blockchains, we count for uncles

107 Ethereum network

Interblock times τ_E reduced: between 13 and 14 sec today

More or less block propagation time

A priori orphan blocks

To decide between two blockchains, we count for uncles

Variation of GHOST protocol

108 Ethereum network

Interblock times τ_E reduced: between 13 and 14 sec today

More or less block propagation time

A priori orphan blocks

To decide between two blockchains, we count for uncles

Variation of GHOST protocol

Blocks signal uncles

109 Ethereum network

Interblock times τ_E reduced: between 13 and 14 sec today

More or less block propagation time

A priori orphan blocks

To decide between two blockchains, we count for uncles

Variation of GHOST protocol

Blocks signal uncles

Incentives: uncle rewards and inclusion rewards

110 Uncles and nephews

111 Uncles and nephews

Definition 82. *An "uncle" is a stale block whose parent belongs to the blockchain and signaled by an official block called "nephew".*

112 Uncles and nephews

Definition 84. *An "uncle" is a stale block whose parent belongs to the blockchain and signaled by an official block called "nephew".*

Definition 85. *The distance between a nephew and an uncle is the number of blocks between the nephew and the uncle's parent.*

113 Uncles and nephews

Definition 86. *An "uncle" is a stale block whose parent belongs to the blockchain and signaled by an official block called "nephew".*

Definition 87. *The distance between a nephew and an uncle is the number of blocks between the nephew and the uncle's parent.*

A nephew block can refer at most two uncles.

114 Uncles and nephews

Definition 88. *An “uncle” is a stale block whose parent belongs to the blockchain and signaled by an official block called “nephew”.*

Definition 89. *The distance between a nephew and an uncle is the number of blocks between the nephew and the uncle's parent.*

A nephew block can refer at most two uncles.

An uncle can be refered by a nephew only if its distance d satisfies $d \leqslant n_1$ with $n_1 = 6$ today.

115 Uncles and nephews

Definition 90. *An "uncle" is a stale block whose parent belongs to the blockchain and signaled by an official block called "nephew".*

Definition 91. *The distance between a nephew and an uncle is the number of blocks between the nephew and the uncle's parent.*

A nephew block can refer at most two uncles.

An uncle can be refered by a nephew only if its distance d satisfies $d \leqslant n_1$ with $n_1 = 6$ today.

Uncle reward $K_u(d) = \frac{8-d}{8} \mathbf{1}_{d \leqslant n_1} b$ with $b = 2$ ETH (coinbase)

116 Uncles and nephews

Definition 92. *An "uncle" is a stale block whose parent belongs to the blockchain and signaled by an official block called "nephew".*

Definition 93. *The distance between a nephew and an uncle is the number of blocks between the nephew and the uncle's parent.*

A nephew block can refer at most two uncles.

An uncle can be refered by a nephew only if its distance d satisfies $d \leqslant n_1$ with $n_1 = 6$ today.

Uncle reward $K_u(d) = \frac{8-d}{8} \mathbf{1}_{d \leqslant n_1} b$ with $b = 2$ ETH (coinbase)

Inclusion reward $K_n(d) = \pi b$ with $\pi = \frac{1}{32}$

117 Main differences with Bitcoin

118 Main differences with Bitcoin

A different reward system

119 Main differences with Bitcoin

A different reward system

Dangerous. A selfish miner earns money even if its attack fails.

120 Main differences with Bitcoin

A different reward system

Dangerous. A selfish miner earns money even if its attack fails.

Difficulty adjustment is made continuously

121 Main differences with Bitcoin

A different reward system

Dangerous. A selfish miner earns money even if its attack fails.

Difficulty adjustment is made continuously

No natural protection against SM as in Bitcoin with the quite important time before reaching difficulty adjustment and becoming profitable

122 Main differences with Bitcoin

A different reward system

Dangerous. A selfish miner earns money even if its attack fails.

Difficulty adjustment is made continuously

No natural protection against SM as in Bitcoin with the quite important time before reaching difficulty adjustment and becoming profitable

The attack is possibly immediatly profitable in Ethereum

123 Main differences with Bitcoin

A different reward system

Dangerous. A selfish miner earns money even if its attack fails.

Difficulty adjustment is made continuously

No natural protection against SM as in Bitcoin with the quite important time before reaching difficulty adjustment and becoming profitable

The attack is possibly immediatly profitable in Ethereum

Difficulty adjustment incorporates some orphan blocks

124 Main differences with Bitcoin

A different reward system

Dangerous. A selfish miner earns money even if its attack fails.

Difficulty adjustment is made continuously

No natural protection against SM as in Bitcoin with the quite important time before reaching difficulty adjustment and becoming profitable

The attack is possibly immediatly profitable in Ethereum

Difficulty adjustment incorporates some orphan blocks

The difficulty adjustment formula in Ethereum is more robust than the difficulty adjustment formula in Bitcoin.

125 Selfish Mining in Ethereum

126 Selfish Mining in Ethereum

There is only one selfish mining strategy in Bitcoin but there are plenty ones in Ethereum.

127 Selfish Mining in Ethereum

There is only one selfish mining strategy in Bitcoin but there are plenty ones in Ethereum.

In Bitcoin, only the number of blocks L and Z added to the official blockchain per cycle are important.

128 Selfish Mining in Ethereum

There is only one selfish mining strategy in Bitcoin but there are plenty ones in Ethereum.

In Bitcoin, only the number of blocks L and Z added to the official blockchain per cycle are important.

In Ethereum, if the attacker releases his block one by one, she creates a lot of competition with the honest miners. Hence, there are a lot of uncles.

129 Selfish Mining in Ethereum

There is only one selfish mining strategy in Bitcoin but there are plenty ones in Ethereum.

In Bitcoin, only the number of blocks L and Z added to the official blockchain per cycle are important.

In Ethereum, if the attacker releases his block one by one, she creates a lot of competition with the honest miners. Hence, there are a lot of uncles.

If the attacker witholds its fork and only release it at the end of an attack cycle, there are few competitions and few uncles.

130 Selfish Mining in Ethereum

There is only one selfish mining strategy in Bitcoin but there are plenty ones in Ethereum.

In Bitcoin, only the number of blocks L and Z added to the official blockchain per cycle are important.

In Ethereum, if the attacker releases his block one by one, she creates a lot of competition with the honest miners. Hence, there are a lot of uncles.

If the attacker witholds its fork and only release it at the end of an attack cycle, there are few competitions and few uncles.

Also the attacker can decide to ignore all uncles. She can also signal some uncles...

131 Short Bibliography

132 Short Bibliography

Quite recent topic

133 Short Bibliography

Quite recent topic

The Impact of Uncle Rewards on Selfish Mining in Ethereum, Fabian Ritz, Alf Zugenmaier

134 Short Bibliography

Quite recent topic

The Impact of Uncle Rewards on Selfish Mining in Ethereum, Fabian Ritz, Alf Zugenmaier

Selfish mining in Ethereum, Chen Feng, Jianyu Niu

135 Short Bibliography

Quite recent topic

The Impact of Uncle Rewards on Selfish Mining in Ethereum, Fabian Ritz, Alf Zugenmaier

Selfish mining in Ethereum, Chen Feng, Jianyu Niu

In both articles, only the classical case has been considered

136 Short Bibliography

Quite recent topic

The Impact of Uncle Rewards on Selfish Mining in Ethereum, Fabian Ritz, Alf Zugenmaier

Selfish mining in Ethereum, Chen Feng, Jianyu Niu

In both articles, only the classical case has been considered

Classical case = the attacker refers to all possible uncles and (if possible) always broadcasts the part of his fork sharing the same height that the official blockchain

137 Short Bibliography

Quite recent topic

The Impact of Uncle Rewards on Selfish Mining in Ethereum, Fabian Ritz, Alf Zugenmaier

Selfish mining in Ethereum, Chen Feng, Jianyu Niu

In both articles, only the classical case has been considered

Classical case = the attacker refers to all possible uncles and (if possible) always broadcasts the part of his fork sharing the same height that the official blockchain

First article: simulations

138 Short Bibliography

Quite recent topic

The Impact of Uncle Rewards on Selfish Mining in Ethereum, Fabian Ritz, Alf Zugenmaier

Selfish mining in Ethereum, Chen Feng, Jianyu Niu

In both articles, only the classical case has been considered

Classical case = the attacker refers to all possible uncles and (if possible) always broadcasts the part of his fork sharing the same height that the official blockchain

First article: simulations

Second article: state machine approach which leads to a quite complicated formula involving a double infinite sum for the long-term apparent hashrate

139 Some definitions

140 Some definitions

Definition 98. *Let* ω *be a cycle. We denote by* $U(\omega)$ *(resp.* $U_S(\omega), U_H(\omega)$*) the number of uncles created during the cycle* ω *which are refered 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 latter cycle.*

141 Some definitions

Definition 102. *Let ω be a cycle. We denote by $U(\omega)$ (resp. $U_S(\omega), U_H(\omega)$) the number of uncles created during the cycle ω which are refered 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 latter cycle.*

Definition 103. *We denote by R (resp. R_s, R_u, R_n) the revenue (resp. revenue coming from static blocks, uncle rewards, inclusion rewards) of a miner per cycle.*

142 Some definitions

Definition 106. *Let ω be a cycle. We denote by $U(\omega)$ (resp. $U_S(\omega), U_H(\omega)$) the number of uncles created during the cycle ω which are refered 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 latter cycle.*

Definition 107. *We denote by R (resp. R_s, R_u, R_n) the revenue (resp. revenue coming from static blocks, uncle rewards, inclusion rewards) of a miner per cycle.*

Note 108. We have: $R = R_s + R_u + R_n$ and R_s does not depend on the particular strategy.

Lemma 109. *Whatever the selfish mining strategy is, we get $\mathbb{E}[R_u] = p^2\, q\, (1-\gamma)\, K_u(1)$ with $K_u(1) = \frac{7}{8} b$ currently on Ethereum and $\mathbb{E}[R_s] = \mathbb{E}[L] b$ with $\mathbb{E}[L] = 1 + \frac{p^2\, q}{p-q}$.*

143 Selfish mining strategies

144 Selfish mining strategies

We consider three different selfish mining strategies:

- Strategy 1 = classical case = Maximum Belligerence & the attacker signals all uncles

145 Selfish mining strategies

We consider three different selfish mining strategies:

- Strategy 1 = classical case = Maximum Belligerence & the attacker signals all uncles
- Strategy 2A = Minimum Belligerence & the attacker signals all uncles

146 Selfish mining strategies

We consider three different selfish mining strategies:

- Strategy 1 = classical case = Maximum Belligerence & the attacker signals all uncles
- Strategy 2A = Minimum Belligerence & the attacker signals all uncles
- Strategy 2B = Minimum Belligerence & the attacker signals no uncles.

147 Selfish mining strategies

We consider three different selfish mining strategies:

- Strategy 1 = classical case = Maximum Belligerence & the attacker signals all uncles
- Strategy 2A = Minimum Belligerence & the attacker signals all uncles
- Strategy 2B = Minimum Belligerence & the attacker signals no uncles.

Strategy 1 maximizes $\mathbb{E}[U]$ and $\mathbb{E}[R]$.

148 Selfish mining strategies

We consider three different selfish mining strategies:

- Strategy 1 = classical case = Maximum Belligerence & the attacker signals all uncles
- Strategy 2A = Minimum Belligerence & the attacker signals all uncles
- Strategy 2B = Minimum Belligerence & the attacker signals no uncles.

Strategy 1 maximizes $\mathbb{E}[U]$ and $\mathbb{E}[R]$.

Strategy 2B minimizes $\mathbb{E}[U]$ and $\mathbb{E}[R]$.

149 Selfish mining strategies

We consider three different selfish mining strategies:

- Strategy 1 = classical case = Maximum Belligerence & the attacker signals all uncles
- Strategy 2A = Minimum Belligerence & the attacker signals all uncles
- Strategy 2B = Minimum Belligerence & the attacker signals no uncles.

Strategy 1 maximizes $\mathbb{E}[U]$ and $\mathbb{E}[R]$.

Strategy 2B minimizes $\mathbb{E}[U]$ and $\mathbb{E}[R]$.

Strategy 2A in the middle...

150 Revenue ratio with the new difficulty adjustment formula

151 Revenue ratio with the new difficulty adjustment formula

The revenue ratio of a strategy (recent DA on Ethreum) is proportional to

$$\begin{aligned}\tilde{\Gamma}_E &= \frac{\mathbb{E}[R]}{\mathbb{E}[L]+\mathbb{E}[U]} \\ &= \frac{\mathbb{E}[R_s]+\mathbb{E}[R_u]+\mathbb{E}[R_n]}{\mathbb{E}[L]+\mathbb{E}[U]}\end{aligned}$$

152 Revenue ratio with the new difficulty adjustment formula

The revenue ratio of a strategy (recent DA on Ethreum) is proportional to

$$\begin{aligned}\tilde{\Gamma}_E &= \frac{\mathbb{E}[R]}{\mathbb{E}[L] + \mathbb{E}[U]} \\ &= \frac{\mathbb{E}[R_s] + \mathbb{E}[R_u] + \mathbb{E}[R_n]}{\mathbb{E}[L] + \mathbb{E}[U]}\end{aligned}$$

Only the terms in red depend on the strategy.

153 Revenue ratio with the new difficulty adjustment formula

The revenue ratio of a strategy (recent DA on Ethreum) is proportional to

$$\begin{aligned}\tilde{\Gamma}_E &= \frac{\mathbb{E}[R]}{\mathbb{E}[L]+\mathbb{E}[U]} \\ &= \frac{\mathbb{E}[R_s]+\mathbb{E}[R_u]+\mathbb{E}[R_n]}{\mathbb{E}[L]+\mathbb{E}[U]}\end{aligned}$$

Only the terms in red depend on the strategy.

Strategy 1 maximizes the numerator (but also the denominator). Strategy 2B minimizes the denominator (but also the numerator).

154 Revenue ratio with the new difficulty adjustment formula

The revenue ratio of a strategy (recent DA on Ethreum) is proportional to

$$
\begin{aligned}
\tilde{\Gamma}_E &= \frac{\mathbb{E}[R]}{\mathbb{E}[L]+\mathbb{E}[U]} \\
&= \frac{\mathbb{E}[R_s]+\mathbb{E}[R_u]+\mathbb{E}[R_n]}{\mathbb{E}[L]+\mathbb{E}[U]}
\end{aligned}
$$

Only the terms in red depend on the strategy.

Strategy 1 maximizes the numerator (but also the denominator). Strategy 2B minimizes the denominator (but also the numerator).

Theorem 114. *We have:* $\tilde{\Gamma}_E = \tilde{\Gamma}_B \cdot \frac{\mathbb{E}[L]}{\mathbb{E}[L]+\mathbb{E}[U]} + \frac{p^2 q\, K_u(1)}{\mathbb{E}[L]+\mathbb{E}[U]} + \frac{\mathbb{E}[U_S]}{\mathbb{E}[L]+\mathbb{E}[U]}\,\pi$

155 Dyck words, Dyck paths and probability space

156 Dyck words, Dyck paths and probability space

A Dyck word w can be identified with a Dyck path $X : [0, 2n] \longrightarrow \mathbb{N}$ such that $X_0 = 0$ and $X_{n+1} = X_n + 1$ (resp. $X_{n+1} = X_n - 1$) if and only if $w_i = S$ (resp. $w_i = H$).

157 Dyck words, Dyck paths and probability space

A Dyck word w can be identified with a Dyck path $X : [0, 2n] \longrightarrow \mathbb{N}$ such that $X_0 = 0$ and $X_{n+1} = X_n + 1$ (resp. $X_{n+1} = X_n - 1$) if and only if $w_i = S$ (resp. $w_i = H$).

The space $\mathcal{D}$ is a probability space with a probability measure $\bar{\mathbb{P}}$ given by $\bar{\mathbb{P}}[w] = p\,(p\,q)^n$ for $w \in \mathcal{D}_n$. If $w \in \mathcal{D}$, then $\mathbb{P}[\omega = \mathrm{SS}\,w\,H] = q^2\,\bar{\mathbb{P}}[w]$.

158 Dyck words, Dyck paths and probability space

A Dyck word w can be identified with a Dyck path $X : [0, 2n] \longrightarrow \mathbb{N}$ such that $X_0 = 0$ and $X_{n+1} = X_n + 1$ (resp. $X_{n+1} = X_n - 1$) if and only if $w_i = S$ (resp. $w_i = H$).

The space $\mathcal{D}$ is a probability space with a probability measure $\bar{\mathbb{P}}$ given by $\bar{\mathbb{P}}[w] = p\,(p\,q)^n$ for $w \in \mathcal{D}_n$. If $w \in \mathcal{D}$, then $\mathbb{P}[\omega = \mathrm{SS}\,w\,H] = q^2\,\bar{\mathbb{P}}[w]$.

Dyck paths more appropriated than Dyck words for Ethereum for the following reason.

159 Dyck words, Dyck paths and probability space

A Dyck word w can be identified with a Dyck path $X : [0, 2n] \longrightarrow \mathbb{N}$ such that $X_0 = 0$ and $X_{n+1} = X_n + 1$ (resp. $X_{n+1} = X_n - 1$) if and only if $w_i = S$ (resp. $w_i = H$).

The space $\mathcal{D}$ is a probability space with a probability measure $\bar{\mathbb{P}}$ given by $\bar{\mathbb{P}}[w] = p\,(p\,q)^n$ for $w \in \mathcal{D}_n$. If $w \in \mathcal{D}$, then $\mathbb{P}[\omega = \mathrm{SS}\,w\,H] = q^2\,\bar{\mathbb{P}}[w]$.

Dyck paths more appropriated than Dyck words for Ethereum for the following reason.

Proposition 119. *Let ω be an attack cycle with $\omega = SSwH$ and $w \in \mathcal{D}$. Let $\mathfrak{b}_i$ be the i-th block validated in ω. If $\mathfrak{b}_i$ is an uncle, then $X_i = X_{i-1} - 1$ and $X_i < n_1 - 2$.*

Proof. We have that $X_i + 2 = h(\mathfrak{f}) - h(\mathfrak{b}_i)$ where $h(\mathfrak{f})$ (resp. $h(\mathfrak{b}_i)$) is the the height of the secret block at the time of the creation of $\mathfrak{b}_i$ (resp. the height of $\mathfrak{b}_i$). $\square$

160 Strategy 1: Maximum Belligerence & refers all (classical case)

161 Strategy 1: Maximum Belligerence & refers all (classical case)

We need to compute $\mathbb{E}[U_S]$ and $\mathbb{E}[U]$.

162 Strategy 1: Maximum Belligerence & refers all (classical case)

We need to compute $\mathbb{E}[U_S]$ and $\mathbb{E}[U]$.

We can precise Proposition 119.

Proposition 126. *Let ω be a cycle with $\omega = SSwH$ and $w \in \mathcal{D}$. Let $\mathfrak{b}_i$ be the i-th block validated in ω. If $X_i < X_{i-1}$ and $X_i < n_1 - 2$ then $\mathfrak{b}_i$ is an uncle with probability γ unless $X_i < n_1 - 2$ and $\mathfrak{b}_i$ is the first block validated by the honest miners.*

163 Strategy 1: Maximum Belligerence & refers all (classical case)

We need to compute $\mathbb{E}[U_S]$ and $\mathbb{E}[U]$.

We can precise Proposition 119.

Proposition 129. *Let ω be a cycle with $\omega = SSwH$ and $w \in \mathcal{D}$. Let $\mathfrak{b}_i$ be the i-th block validated in ω. If $X_i < X_{i-1}$ and $X_i < n_1 - 2$ then $\mathfrak{b}_i$ is an uncle with probability γ unless $X_i < n_1 - 2$ and $\mathfrak{b}_i$ is the first block validated by the honest miners.*

Definition 130. *If ω is a cycle starting with SS, we denote by $H(\omega)$ the number of blocks mined by the honest miners and corresponding to an index i such that $X_i < X_{i-1}$ and $X_i < n_1 - 2$.*

164 Strategy 1: Maximum Belligerence & refers all (classical case)

We need to compute $\mathbb{E}[U_S]$ and $\mathbb{E}[U]$.

We can precise Proposition 119.

Proposition 132. *Let ω be a cycle with $\omega = SSwH$ and $w \in \mathcal{D}$. Let $\mathfrak{b}_i$ be the i-th block validated in ω. If $X_i < X_{i-1}$ and $X_i < n_1 - 2$ then $\mathfrak{b}_i$ is an uncle with probability γ unless $X_i < n_1 - 2$ and $\mathfrak{b}_i$ is the first block validated by the honest miners.*

Definition 133. *If ω is a cycle starting with SS, we denote by $H(\omega)$ the number of blocks mined by the honest miners and corresponding to an index i such that $X_i < X_{i-1}$ and $X_i < n_1 - 2$.*

Proposition 134. *We have:* $\mathbb{E}[H(\omega)|\omega = \mathrm{SS}*] = \frac{p}{p-q}\left(1 - \left(\frac{q}{p}\right)^{n_1-1}\right)$

Proposition 135. *We have:* $\mathbb{E}[U] = q + \frac{q^3\gamma}{p-q} - \frac{p^3}{p-q}\left(\frac{q}{p}\right)^{n_1+1}\gamma - q^{n_1+1}(1-\gamma)$

Proposition 136. *We have:* $\mathbb{E}[U] = q + \frac{q^3\gamma}{p-q} - \frac{p^3}{p-q}\left(\frac{q}{p}\right)^{n_1+1}\gamma - q^{n_1+1}(1-\gamma)$

Proof. We have $U(\{H\}) = 0$ and $U(\omega) = 1$ if $\omega \in \{\mathrm{SHS}, \mathrm{SHH}\}$. Also,

$$\mathbb{E}[U|\omega = \mathrm{SS.}\ \ ] = \mathbb{E}[H(\omega)|\omega = \mathrm{SS}]\gamma + (1-\gamma)(p + p\,q + .\ \ + p\,q^{n_1-2})$$

Indeed, there is a probability γ that a block $\mathfrak{b}_i$ satisfying $X_i = X_{i-1} - 1$ and $X_i < n_1 - 2$ is an uncle except for the first block mined by the honest miners. In this case, the probability is 1. So,

$$\mathbb{E}[U] = p\,q + \left[\frac{p}{p-q}\left(1 - \left(\frac{q}{p}\right)^{n_1-1}\right)\gamma + (1-\gamma)\left(1 - q^{n_1-1}\right)\right]\cdot q^2$$

$\square$

Definition 137. *Let $V(\omega)$ be the number of uncles $\mathfrak{u} \in \omega$ refered by a nephew $\mathfrak{n} \notin \omega$.*

Definition 139. *Let* $V(\omega)$ *be the number of uncles* $\mathfrak{u} \in \omega$ *refered by a nephew* $\mathfrak{n} \notin \omega$.

Lemma 140. *We have:* $\mathbb{E}[V] = \frac{q^2}{p}\,(1 - q^{n_1 - 1})\,\gamma + (1-\gamma)\,p\,q^2\,\frac{1-(p\,q)^{n_1-1}}{1-p\,q}$

Definition 141. *Let $V(\omega)$ be the number of uncles $\mathfrak{u} \in \omega$ refered by a nephew $\mathfrak{n} \notin \omega$.*

Lemma 142. *We have:* $\mathbb{E}[V] = \frac{q^2}{p}(1 - q^{n_1 - 1})\,\gamma + (1 - \gamma)\,p\,q^2\,\frac{1 - (p\,q)^{n_1 - 1}}{1 - p\,q}$

Proof. We have $V(\omega) = 0$ if $\omega \in \{H, \mathrm{SHH}, \mathrm{SHS}\}$. If $\omega = *\mathrm{SHH}.\ \mathrm{H}$ with k H at the end, then only the last $n_1 - 1$ blocks can be uncles signaled by future blocks in the next cycle after ω unless $\omega = \mathrm{SS}.\ \mathrm{SH}.\ \mathrm{H}$ with at most n_1 letters S and $n_1 - 1$ letters H. In that case, the first block validated by the honest miners. So,

$$\mathbb{E}[V] = q^2 \sum_{k \geqslant 1} \inf(k, n_1 - 1)\,p\,q^{k-1}\gamma + (1 - \gamma)\,q \sum_{k=1}^{n_1 - 1} (p\,q)^k$$

Note that $p\,q^{k-1}$ is the probability that a Dyck word ends exactly with $(k - 1)$ H. □

Definition 143. *Let $V(\omega)$ be the number of uncles $\mathfrak{u} \in \omega$ refered by a nephew $\mathfrak{n} \notin \omega$.*

Lemma 144. *We have:* $\mathbb{E}[V] = \frac{q^2}{p} (1 - q^{n_1 - 1})\, \gamma + (1 - \gamma)\, p\, q^2 \frac{1 - (p\, q)^{n_1 - 1}}{1 - p\, q}$

Proof. We have $V(\omega) = 0$ if $\omega \in \{H, \mathrm{SHH}, \mathrm{SHS}\}$. If $\omega = *\mathrm{SHH}.\ \mathrm{H}$ with k H at the end, then only the last $n_1 - 1$ blocks can be uncles signaled by future blocks in the next cycle after ω unless $\omega = \mathrm{SS}.\ \mathrm{SH}.\ \mathrm{H}$ with at most n_1 letters S and $n_1 - 1$ letters H. In that case, the first block validated by the honest miners. So,

$$\mathbb{E}[V] = q^2 \sum_{k \geqslant 1} \inf(k, n_1 - 1)\, p\, q^{k-1} \gamma + (1 - \gamma)\, q \sum_{k=1}^{n_1 - 1} (p\, q)^k$$

Note that $p\, q^{k-1}$ is the probability that a Dyck word ends exactly with $(k-1)$ H. □

Proposition 145. *We have:* $\mathbb{E}[U_h] = p^2 q + (p + (1-\gamma)p^2 q)\,\mathbb{E}[V]$

Proposition 147. *We have:* $\mathbb{E}[U_h] = p^2 q + (p + (1-\gamma)p^2 q)\,\mathbb{E}[V]$

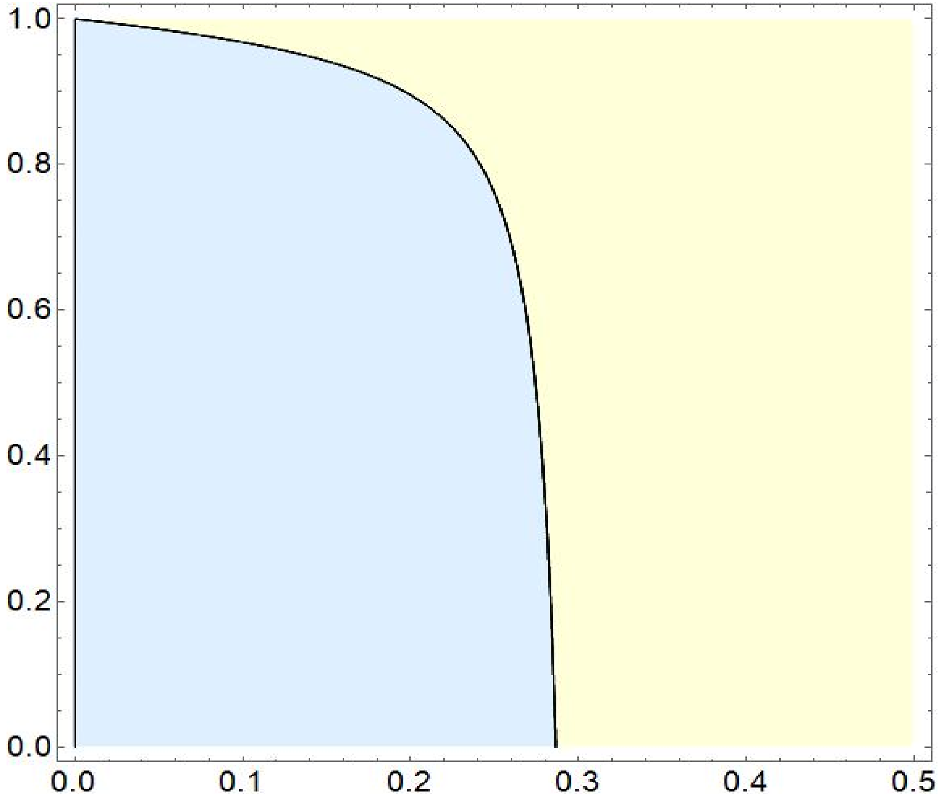
Proof. Let ω be a cycle and let $U_h^{(1)}(\omega)$ (resp. $U_h^{(2)}(\omega)$) be the number of uncles refered by honest nephews only present in ω (resp. not present in ω). Clearly, $\mathbb{E}\big[U_h^{(1)}\big] = p^2 q$. Moreover, the probability that H is the first official block of the next attack cycle is $p + (1-\gamma)p^2 q$. So, $\mathbb{E}\big[U_h^{(2)}\big] = (p + (1-\gamma)p^2 q)\,\mathbb{E}[V]$. □

Proposition 149. *We have:* $\mathbb{E}[U_h] = p^2 q + (p + (1-\gamma)p^2 q)\,\mathbb{E}[V]$

Proof. Let ω be a cycle and let $U_h^{(1)}(\omega)$ (resp. $U_h^{(2)}(\omega)$) be the number of uncles refered by honest nephews only present in ω (resp. not present in ω). Clearly, $\mathbb{E}\big[U_h^{(1)}\big] = p^2 q$. Moreover, the probability that H is the first official block of the next attack cycle is $p + (1-\gamma)p^2 q$. So, $\mathbb{E}\big[U_h^{(2)}\big] = (p + (1-\gamma)p^2 q)\,\mathbb{E}[V]$. □

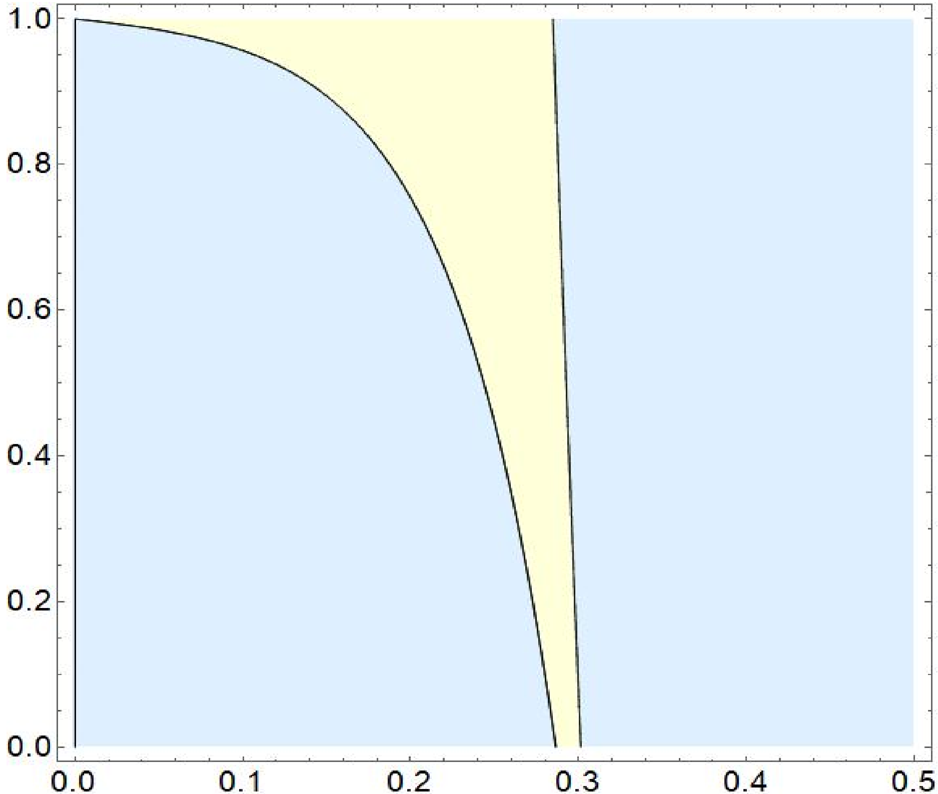
Corollary 150. *We have:*

$$\begin{aligned}\mathbb{E}[U_S] &= q + \frac{q^3\gamma}{p-q} - \frac{p\,q^2}{p-q}\left(\frac{q}{p}\right)^{n_1-1}\gamma - q^{n_1+1}(1-\gamma)\\ &- \left[p^2 q + (p + (1-\gamma)p^2 q)\left(\frac{q^2}{p}(1-q^{n_1-1})\gamma + (1-\gamma)p\,q^2\,\frac{1-(p\,q)^{n_1-1}}{1-p\,q}\right)\right]\end{aligned}$$



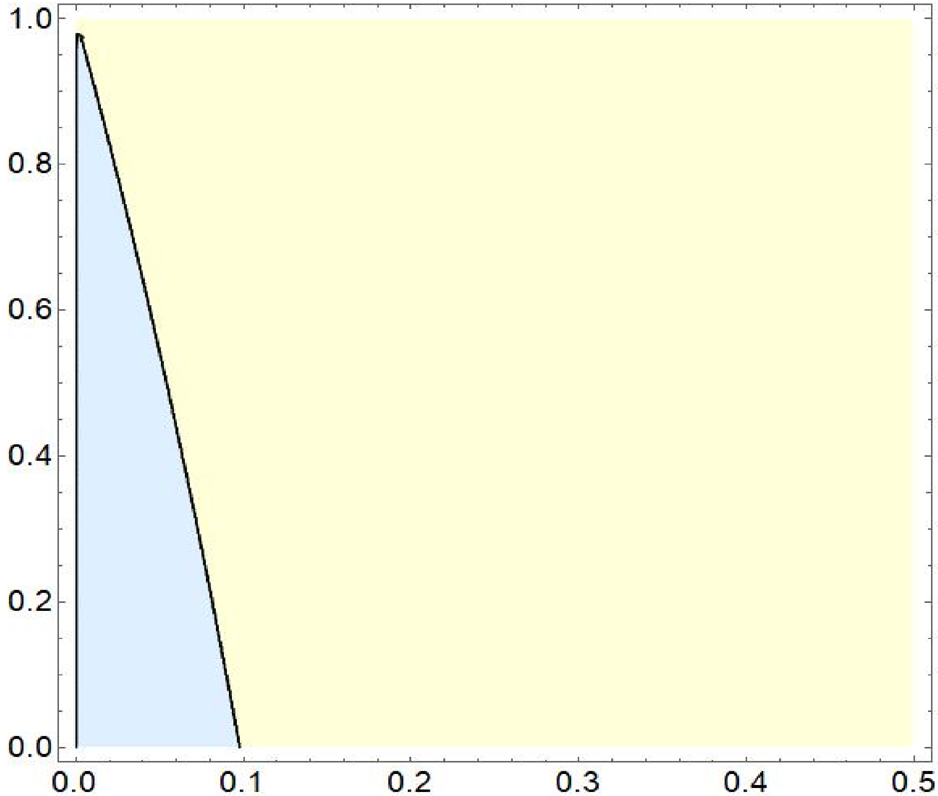
HM (blue) and SM (yellow). X-axis: q, Y-axis: γ

Figure 5.



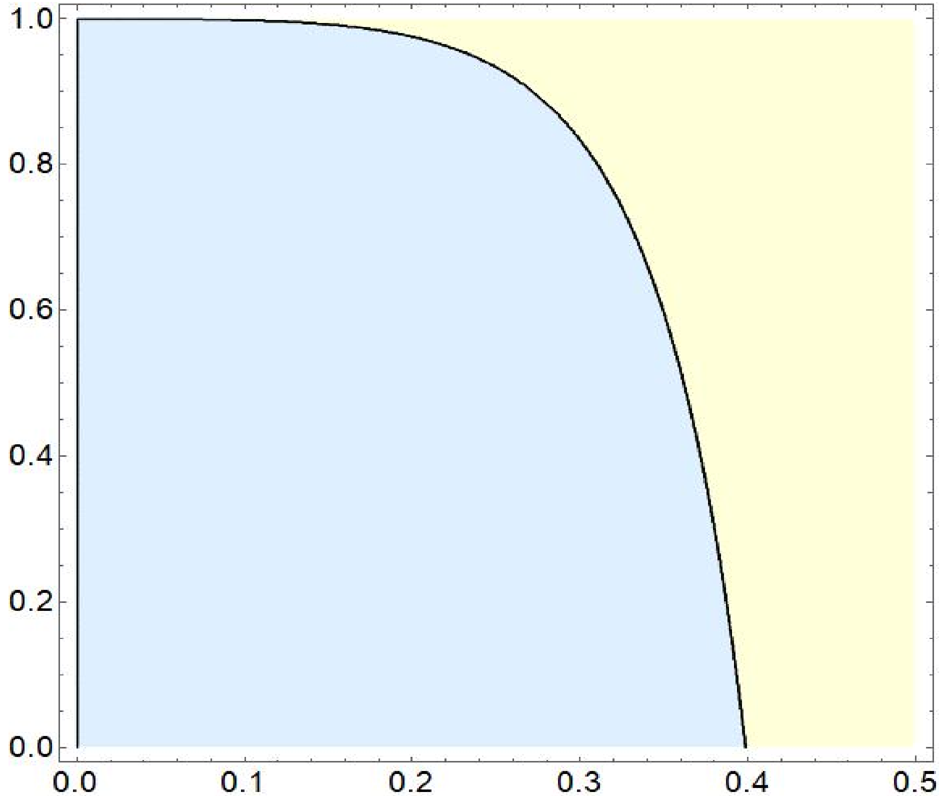
From left to right: HM, SM2A and SM2B

Figure 6.



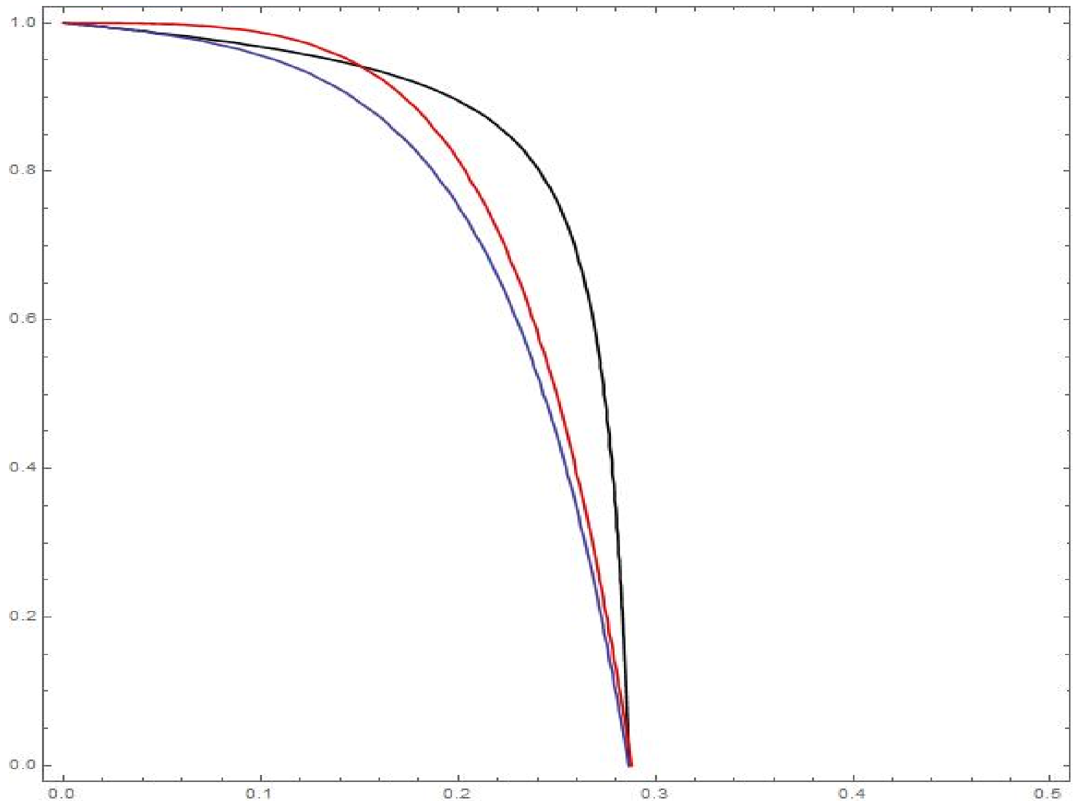
From left to right: HM, SM (old difficulty adjustment)

Figure 7.



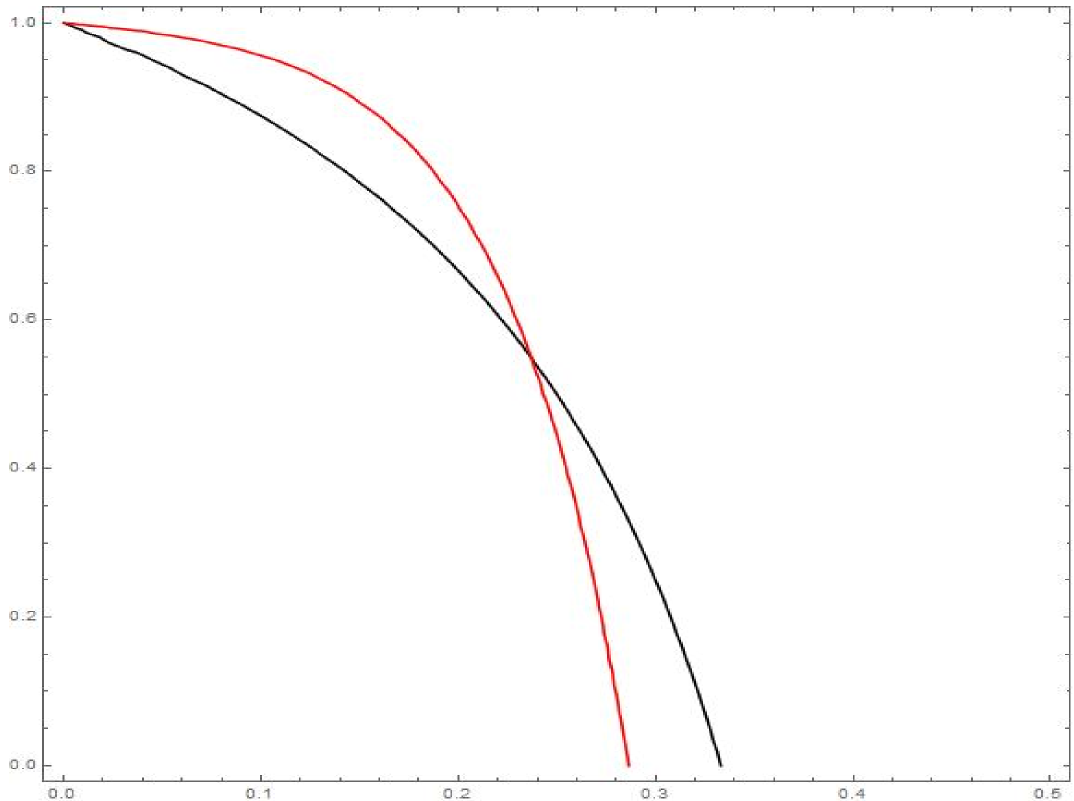
From left to right: HM, SM (possible difficulty adjustment with uncles)

Figure 8.



SM1 (black), SM2A (blue), SM2B (red)

Figure 9.



Thresholds SM Bitcoin (black) & SM2A Ethereum (red)

Figure 10.