The Impact of Uncle Rewards on Selfish Mining in Ethereum

Fabian Ritz
Alf Zugenmaier

12 Munich University of Applied Sciences
2 Concordia University
Ethereum relies on a PoW-Blockchain

Ethereum features stale block inclusion

Ethereum rewards uncles and nephews

Do uncles matter in Ethereum Selfish Mining?
Overview

1 | Selfish Mining in General
2 | Ethereum
3 | Selfish Mining in Ethereum
4 | Simulator Design
5 | Simulation Results
1 | Selfish Mining in General
1 | Selfish Mining | General Model

Selfish miner:

• deviates from the crypto protocol
• is perfectly connected
• fraction of mining power $\alpha$

Honest network:

• complies with the crypto protocol as intended
• is affected by network lag
• fraction of mining power $\beta$

$\alpha + \beta = 1$
• selfish miner builds a secret fork
1 | Selfish Mining | Strategy (1)

- selfish miner builds a secret fork
• selfish miner builds a secret fork
• honest network nearly catches up
• selfish miner builds a secret fork
• honest network nearly catches up

-> selfish miner publishes his fork
-> honest network adopts the longest chain and loses a block
The Impact of Uncle Rewards on Selfish Mining in Ethereum

-> applies to chains of any length
-> applies to chains of any length

-> stale blocks reduce the difficulty (long term effect)
• selfish miner tries to build a secret fork
• selfish miner tries to build a secret fork
• honest network catches up
1 | Selfish Mining | Strategy (3)

- selfish miner tries to build a secret fork
- honest network catches up

\[
\gamma: \text{fraction of honest network mining on selfish miner’s fork in case of ties}
\]

\[
\rightarrow \text{selfish miner publishes his fork}
\]

\[
\rightarrow \text{honest network’s mining power gets diverted}
\]
• selfish miner extends his fork

-> honest network loses a block
1 | Selfish Mining | Strategy (4)

- selfish miner extends his fork
- honest network extends selfish miner’s fork

-> honest network loses a block
• selfish miner extends his fork
• honest network extends selfish miner’s fork
-> honest network loses a block

• honest network extends its fork
-> selfish miner loses a block
2 | Ethereum
Ethereum blockchain:

- extended by miner nodes
- secured by Proof-of-Work performed in ETHash
- consensus rule: longest chain
- random tie breaking
Byzantinum (October 2017):

- latest of 7 hard forks
- 15 seconds average block time*
- 16% stale blocks (during March 2018)*
Referencing block (*nephew*):

- regular block (part of the main chain)
- at most 2 references
- 1/32 block reward per reference
Referenced block (*uncle*):

- direct descendant of a regular block
- at least 1, at most 6 blocks in the past
- at most 7/8, at least 2/8 block reward
- counted as regular block within difficulty calculation*

*Byzantinum*
b1a: first generation uncle (7/8 block reward)
b2b: second generation uncle (6/8 block reward)
b3b: stale block (no reward)
Uncles might change a lot of situations...
3 | Selfish Mining in Ethereum
3 | Selfish Mining | Stale Block Inclusion (1)
-> stale block inclusion reduces the risk of losing single blocks
3 | Selfish Mining | Stale Block Inclusion (2)
3 | Selfish Mining | Stale Block Inclusion (2)

-> stale block inclusion does not help (much) in case of long forks
• due to random tie breaking: \( y = \frac{1}{\text{number of competing chains}} \)

• uncles affect difficulty calculation*

• uncles slow the growth of the public chain

*Byzantinum
• selfish miner has a secret fork
• selfish miner has a secret fork
• honest network creates a new fork

-> selfish miner maintains lead
-> problematic if selfish miner starts a new fork
• selfish miner publishes a secret fork
3 | Selfish Mining | Stale Block Generation (2)

- selfish miner publishes a secret fork
- honest network forks the published fork

→ selfish miner might lose a single block
• selfish miner publishes a secret fork
• honest network extends the formerly stale branch

-> selfish miner might lose an entire fork
4 | Simulator Design
Stale blocks need to be generated:

- by the honest network
- at a certain ratio $\delta$

Stale blocks need to be referenced:

- with a probability of $1/3$ per slot (observed during November 2017)
- according to strategies:
  - selfish miner -> own blocks
  - honest network -> any blocks
Monte Carlo Simulation with discrete block generation events:

\( \alpha \) the selfish miner mines a block

\( \beta \) the honest network mines a block which is a

\( \beta \cdot \delta \) stale block

\( \beta - (\beta \cdot \delta) \) regular block
Relative revenue ratio (RRR):
• a party’s fraction of all mining rewards

Problem:
• selfish mining causes additional stale blocks
• stale blocks eventually become uncle blocks
• uncle blocks lower the overall mining rewards*

Absolute revenue ratio (ARR):
• takes lowered mining rewards into account

*Byzantinum
Relative block ratio (RBR):

- ratio of a party’s regular blocks to all regular blocks
- measures control over the blockchain

Relative network security (RNS):

- ratio of all regular blocks to all blocks
- measures the computational power that secures the blockchain
5 | Simulation Results
5 | Simulation Results | Absolute Revenue

The Impact of Uncle Rewards on Selfish Mining in Ethereum

- **selfish** \( \delta = 0.00 \)
  - include own uncles
- **selfish** \( \delta = 0.12 \)
  - include own uncles
- **selfish** \( \delta = 0.24 \)
  - include own uncles
- **honest** \( \delta = 0.12 \)
  - include all uncles

![Graph showing the relationship between selfish miner's fraction of mining power and absolute revenue ratio (ARR).](image-url)

- The x-axis represents the selfish miner's fraction of mining power, ranging from 0.10 to 0.45.
- The y-axis represents the selfish miner's absolute revenue ratio (ARR), ranging from 0.0 to 1.0.
- Lines indicate different scenarios:
  - Green line for selfish \( \delta = 0.00 \) with own uncles included.
  - Green dashed line for selfish \( \delta = 0.12 \) with own uncles included.
  - Green double-dashed line for selfish \( \delta = 0.24 \) with own uncles included.
  - Orange line for honest \( \delta = 0.12 \) with all uncles included.

**Legend:**
- Green line: selfish \( \delta = 0.00 \) include own uncles
- Green dashed line: selfish \( \delta = 0.12 \) include own uncles
- Green double-dashed line: selfish \( \delta = 0.24 \) include own uncles
- Orange line: honest \( \delta = 0.12 \) include all uncles
5 | Simulation Results | Break Even of Profitability

The Impact of Uncle Rewards on Selfish Mining in Ethereum

selfish miner's abs. revenue ratio ARR

selfish miner's fraction of mining power $\alpha$

- honest $\delta=0.24$
  - include all uncles
- selfish $\delta=0.24$
  - include own uncles
- honest $\delta=0.12$
  - include all uncles
- selfish $\delta=0.12$
  - include own uncles

0.14 0.17 0.20 0.23 0.26

0.185 0.225
5 | Simulation Results | Blockchain Control

The impact of uncle rewards on selfish mining in Ethereum.

The diagram shows the relationship between the selfish miner's fraction of mining power ($\alpha$) and the selfish miner's regular block ratio (RBR). The x-axis represents $\alpha$, ranging from 0.10 to 0.45, and the y-axis represents RBR, ranging from 0.0 to 1.0.

- The green line with a dotted style represents a selfish miner with $\delta=0.00$, including own uncles.
- The green line with a solid style represents a selfish miner with $\delta=0.12$, including own uncles.
- The green line with a dashed style represents a selfish miner with $\delta=0.24$, including own uncles.
- The orange line represents a threshold of $0.5 + \varepsilon$ attack.
- The gray line represents a honest miner with $\delta=0.12$, including all uncles.

At $\alpha=0.35$, the selfish miner's RBR reaches 0.34, indicating a significant impact on the stability of the blockchain.
The Impact of Uncle Rewards on Selfish Mining in Ethereum

5 | Simulation Results | Network Security

- **selfish δ=0.00**
  - include own uncles

- **selfish δ=0.12**
  - include own uncles

- **selfish δ=0.24**
  - include own uncles

- **honest δ=0.12**
  - include all uncles

![Graph showing relative network security (RNS) vs. selfish miner's fraction of mining power (α).]
5 | Simulation Results | Summary

1) selfish mining becomes profitable at $\alpha = 0.185$ if $\delta = 0.24$

2) selfish miner controls the blockchain with $\alpha = 0.34$ if $\delta = 0.24$

3) selfish mining diverts mining power from securing the blockchain
Uncles matter in Ethereum Selfish Mining!
Related Work

• Eyal and Sirer: Majority is not Enough - Bitcoin Mining is Vulnerable (2013)
• Sapirshtein et al.: Optimal Selfish Mining Strategies in Bitcoin (2015)
Questions?
6 | Backup
A mining party within the network tries to receive a share of the revenue larger than its share of mining power by temporary withholding blocks.
6 | Ethereum | Contracts and Network

• (smart) contracts / code
  • turing-complete capabilities
  • executed by every node via the Ethereum Virtual Machine (EVM)
  • execution requires gas paid in Ether (ETH)

• network
  • peer-to-peer-structure
  • kademlia-based
  • currently 16.000 nodes
The Impact of Uncle Rewards on Selfish Mining in Ethereum
ARR = \frac{nb(p,r) + nb(p,u)}{nb(a,r) + nb(a,u)} \cdot \frac{rev(p)}{nb(p,r) + nb(p,u)} = \frac{rev(p)}{nb(a,r) + nb(a,u)}

nb(p,r) = \text{number of a party’s regular blocks}

nb(p,u) = \text{number of a party’s uncle blocks}

nb(a,r) = \text{number of all regular blocks}

nb(a,u) = \text{number of all uncle blocks}

rev(p) = \text{party’s normalized total revenue, reward per regular block = 1}
• **smart data structure as a blockchain**
  - keeps track of all blocks
  - validates rules (e.g. stale block references)

• **helpers for (complex) problems**
  - find uncle candidates
  - find positions for stale blocks
  - evaluate the blockchain
onSelfFindsBlock() {
    if (secretFork.length() > 0) {
        secretFork.add(newBlock);
    } else {
        secretFork = new Fork(publicChain).add(newBlock);
    }
}
onOthersFindBlock() {
    if (secretFork.height() - publicChain.height() < 2) {
        secretFork.publish();
    }
}
For every parameter combination $\alpha$, $\delta$ and mining strategy:

- 100 blockchains (random walks) with $2^{17}$ blocks (steps)
- mean values and standard deviation $\sigma$ per quantity of interest
- acceptance of mean values
  - for $0.15 < \alpha < 0.30$: $\sigma < 0.001$
  - for any other $\alpha$: $\sigma < 0.01$

$\rightarrow$ accuracy +/- 0.015 near the break even of profitability
6 | If Ethereum Weighted Uncles

weight of sample chains without weighted uncles

weight of sample chains with weighted uncles

green = attacker’s secret fork
black = public chain

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