Selfish Mining in Blockchain Systems

Sheng-Wei Wang

Department of Electronic Engineering
National United University
Miaoli, TAIWAN

swwang@nuu.edu.tw

2024-09-30 @ NYCU

Outlines

1. **Introductions**
   - Blockchains
   - Selfish Mining Strategy
2. **Research Works on Multiple Selfish Miners**
   - Multiple Selfish Mining Strategy
   - An Accurate Analytical Model (ICC2024)
   - Rational Mining Strategy (ICNC2024)
3. **Ongoing & Future Works**
   - A Stochastic Lightweight Blockchain
   - Security Analysis of Sharded Blockchains

Bitcoins

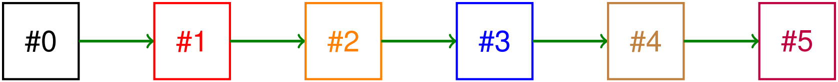
Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Blockchains

- Blockchain is a decentralized ledger stored in a distributed network
- Transactions are securely stored in blocks
- Consecutive blocks form a blockchain using cryptography
- Hash value of previous block is stored



Mining & Miners

- Node creating the block earns rewards (Miner)
- Mining is the process to create a valid block in order to get rewards

Let's talk about mining.....

Mining & Proof-of-Works

- Who will earn the reward?

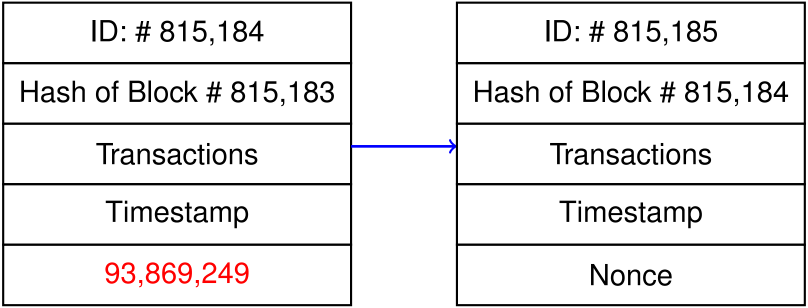
ID: # 815,184
Hash of Block # 815,183
Transactions
Timestamp
Nonce

- Hash of Block #815,183:

  00000000000000000003d09220e85bbdbb832b86e3dc711c5cda888b1daf5985

Mining & Proof-of-Works

- Who will earn the reward?



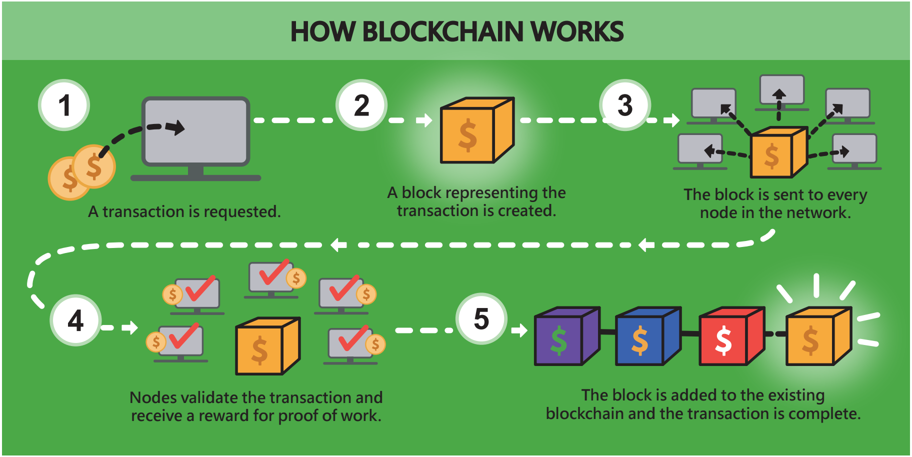
- Hash of Block #815,183:

  00000000000000000003d09220e85bbdbb832b86e3dc711c5cda888b1daf5985

- Hash of Block #815,184:

  00000000000000000000cc167c107a24883b34c16aad188aaa72412cc0ef437a

How A PoW Blockchain Works?



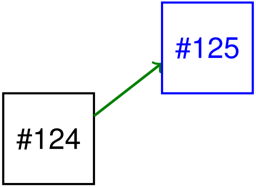
Selfish Mining & Rewards

- How many rewards can a miner earn?
- The number of nonces attempted by a miner per unit of time is defined as his mining rate
- Generally, the probability which the next block is mined by a specific miner shall be proportional to his mining rate
- A mining strategy called **selfish mining** enables a miner to be **profitable**; that is, to earn more rewards than he would be entitled to
- Main idea of selfish mining is not to broadcast the mined blocks when a selfish miner mined a new block

Selfish Mining & Rewards

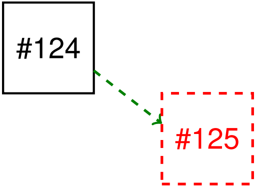
#124

Selfish Mining & Rewards



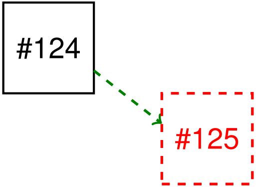
- If honest miner mined the next block first, he announces the block immediately
- All other miners validate the block and start to mine the next one

Selfish Mining & Rewards



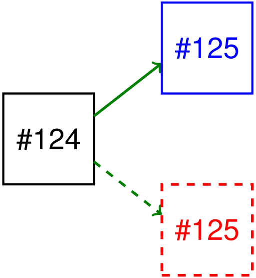
- If honest miner mined the next block first, he announces the block immediately
- All other miners validate the block and start to mine the next one
- If selfish miner mined the next block first, he hides the block in his private branch and starts to mine the next one

Selfish Mining & Rewards



If then honest miner mined the next block first under the above condition:

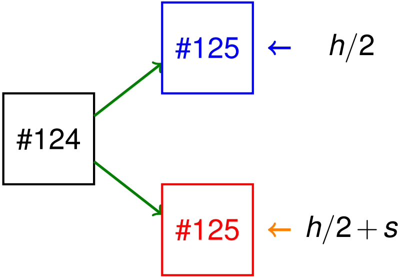
Selfish Mining & Rewards



If then honest miner mined the next block first under the above condition:

- The honest miner announces the block immediately

Selfish Mining & Rewards

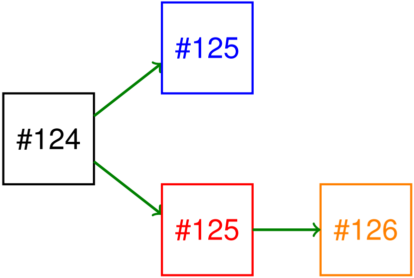


If then honest miner mined the next block first under the above condition:

- The honest miner announces the block immediately
- The selfish miner releases his hidden block immediately

Longest Chain Rule: The branch on which the next block is mined first (longest chain) becomes the valid chain

Selfish Mining & Rewards

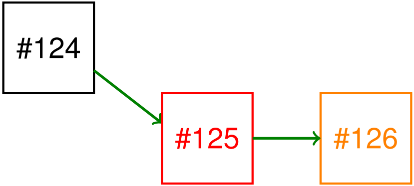


If then honest miner mined the next block first under the above condition:

- The honest miner announces the block immediately
- The selfish miner releases his hidden block immediately

Longest Chain Rule: The branch on which the next block is mined first (longest chain) becomes the valid chain

Selfish Mining & Rewards

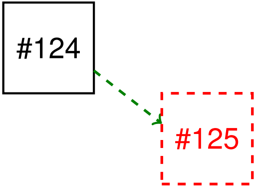


If then honest miner mined the next block first under the above condition:

- The honest miner announces the block immediately
- The selfish miner releases his hidden block immediately

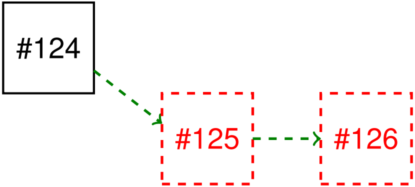
Longest Chain Rule: The branch on which the next block is mined first (longest chain) becomes the valid chain

Selfish Mining & Rewards



If selfish miner mined the next block first under the above condition:

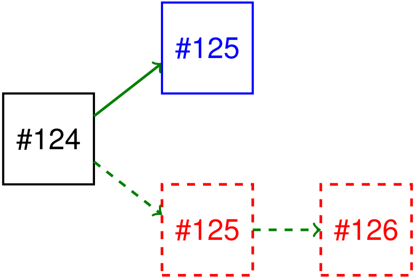
Selfish Mining & Rewards



If selfish miner mined the next block first under the above condition:

- The selfish miner hides the blocks in his branch

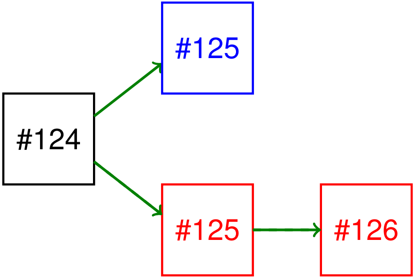
Selfish Mining & Rewards



If selfish miner mined the next block first under the above condition:

- The selfish miner hides the blocks in his branch
- If the honest miner mined the block now, the honest miner releases the block immediately

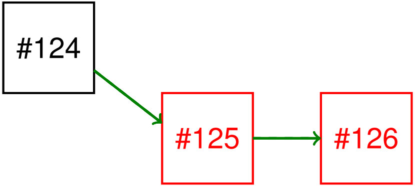
Selfish Mining & Rewards



If selfish miner mined the next block first under the above condition:

- The selfish miner hides the blocks in his branch
- If the honest miner mined the block now, the honest miner releases the block immediately
- The selfish miner releases his all blocks immediately

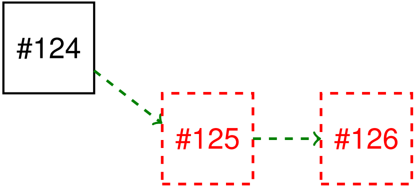
Selfish Mining & Rewards



If selfish miner mined the next block first under the above condition:

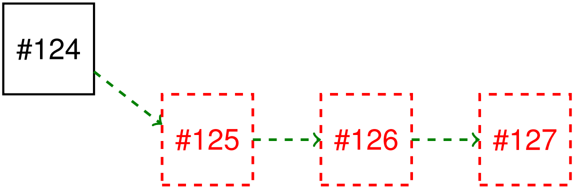
- The selfish miner hides the blocks in his branch
- If the honest miner mined the block now, the honest miner releases the block immediately
- The selfish miner releases his all blocks immediately
- Longest chain rule is applied

Selfish Mining & Rewards



If selfish miner mined the next block first under the above condition:

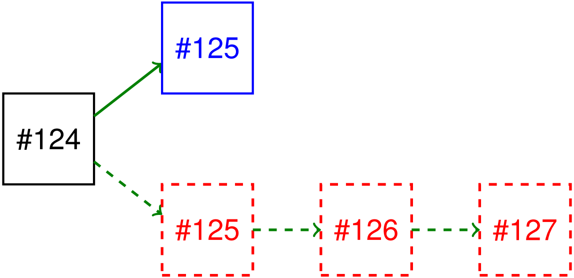
Selfish Mining & Rewards



If selfish miner mined the next block first under the above condition:

- The selfish miner hides the blocks in his branch

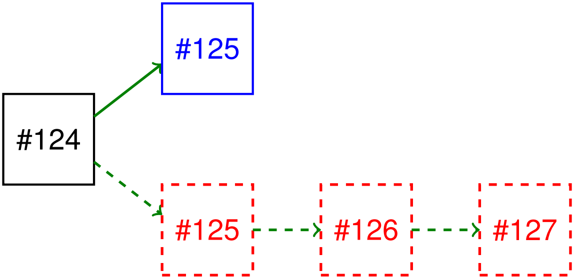
Selfish Mining & Rewards



If selfish miner mined the next block first under the above condition:

- The selfish miner hides the blocks in his branch
- If the honest miner mined the block now, the honest miner releases the block immediately

Selfish Mining & Rewards



If selfish miner mined the next block first under the above condition:

- The selfish miner hides the blocks in his branch
- If the honest miner mined the block now, the honest miner releases the block immediately
- The selfish miner does **NOTHING** now

Reward Earned by Selfish Miner

- Analytical Model Proposed by *I. Eyal* and *E.G. Sirer*

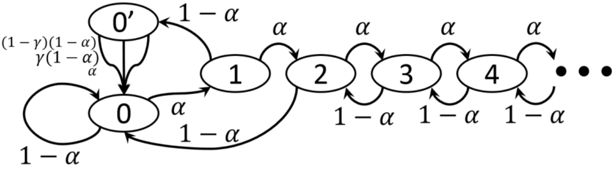
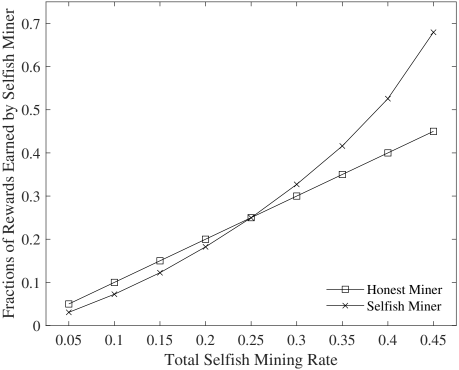


Fig. 1: State machine with transition frequencies.

- Fraction of reward earned by the selfish miner $RW(\alpha)$ can be calculated by a closed-form function of his mining rate α

$$RW(\alpha) = \begin{cases} 1 & \text{if } \alpha \geq 0.5, \\ \frac{\alpha(1-\alpha)^2[4\alpha+\frac{1}{2}(1-2\alpha)]-\alpha^3}{1-\alpha[1+(2-\alpha)\alpha]} & \text{otherwise .} \end{cases}$$

Rewards & Profitable Threshold (25%)



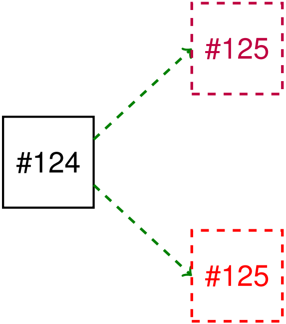
- Profitable: Earns more than those earned if he is honest
- Profitable threshold: the smallest mining rate making a miner profitable

Multiple Selfish Miners

- Selfish mining strategy enables a miner to be profitable
- Multiple miners with sufficient mining rates may choose to employ selfish mining strategy in order to earn more rewards
- There will be multiple independent selfish miners in the blockchain without knowing each other
- We consider a blockchain with TWO selfish miners in this paper

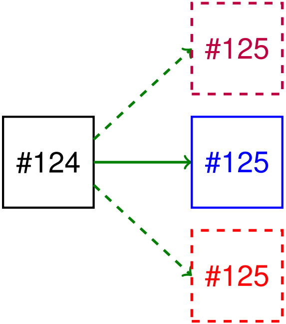
Two Selfish Miners (Case 1)

An honest miner *Henry (r_h)* and two selfish miners *Alice(r_a)* and *Bob(r_b)*



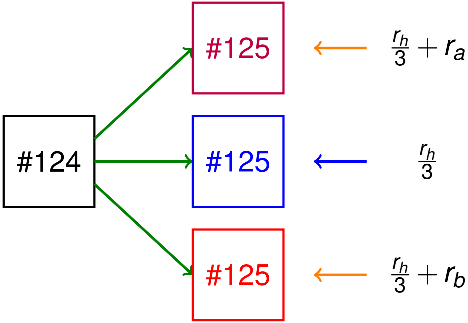
Two Selfish Miners (Case 1)

An honest miner *Henry (r_h)* and two selfish miners *Alice(r_a)* and *Bob(r_b)*



Two Selfish Miners (Case 1)

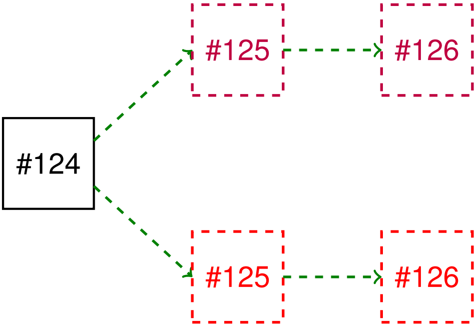
An honest miner *Henry (r_h)* and two selfish miners *Alice(r_a)* and *Bob(r_b)*



Longest chain rule shall be applied.

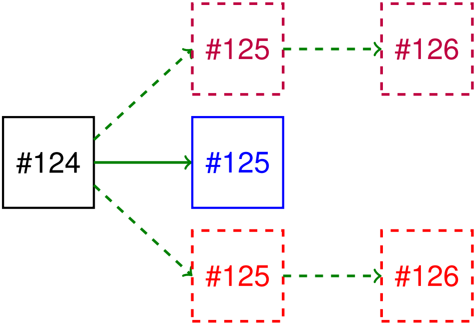
Two Selfish Miners (Case 2)

An honest miner *Henry (r_h)* and two selfish miners *Alice(r_a)* and *Bob(r_b)*



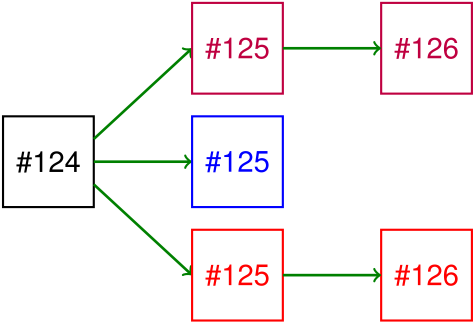
Two Selfish Miners (Case 2)

An honest miner *Henry* (r_h) and two selfish miners *Alice*(r_a) and *Bob*(r_b)



Two Selfish Miners (Case 2)

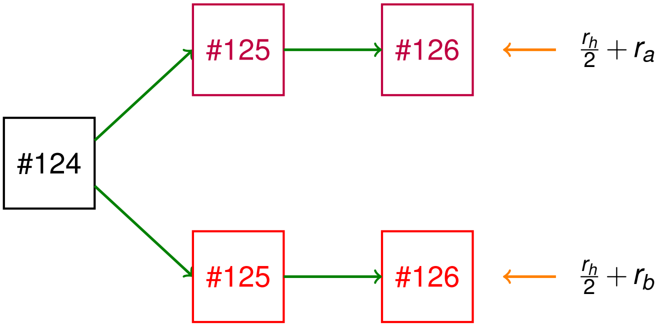
An honest miner *Henry* (r_h) and two selfish miners *Alice*(r_a) and *Bob*(r_b)



Longest chain rule shall be applied.

Two Selfish Miners (Case 2)

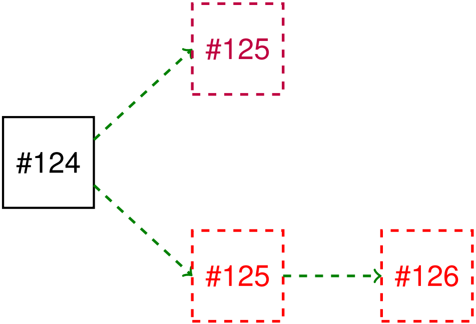
An honest miner *Henry (r_h)* and two selfish miners *Alice(r_a)* and *Bob(r_b)*



Longest chain rule shall be applied.

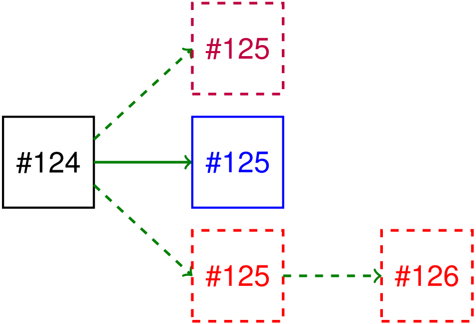
Two Selfish Miners (Case 3)

An honest miner *Henry (r_h)* and two selfish miners *Alice(r_a)* and *Bob(r_b)*



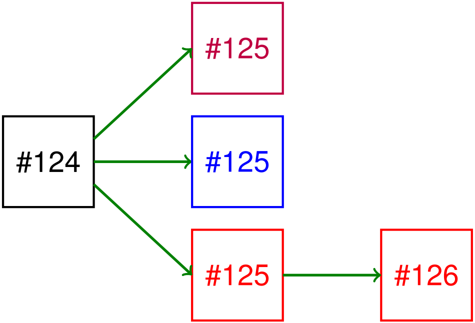
Two Selfish Miners (Case 3)

An honest miner *Henry (r_h)* and two selfish miners *Alice(r_a)* and *Bob(r_b)*



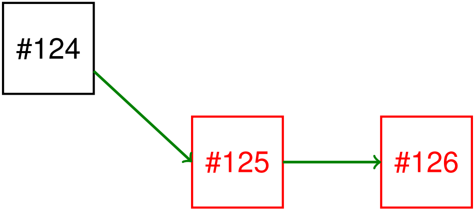
Two Selfish Miners (Case 3)

An honest miner *Henry (r_h)* and two selfish miners *Alice(r_a)* and *Bob(r_b)*



Two Selfish Miners (Case 3)

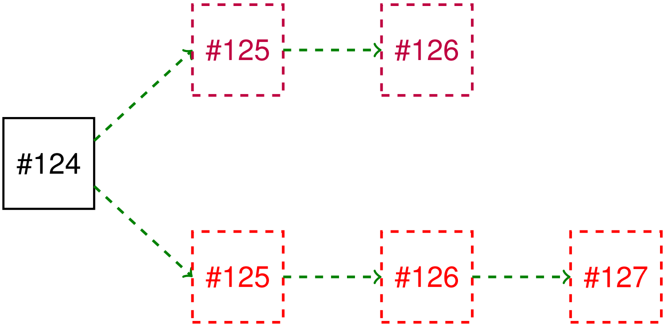
An honest miner *Henry* (r_h) and two selfish miners *Alice*(r_a) and *Bob*(r_b)



Longest chain rule shall be applied.

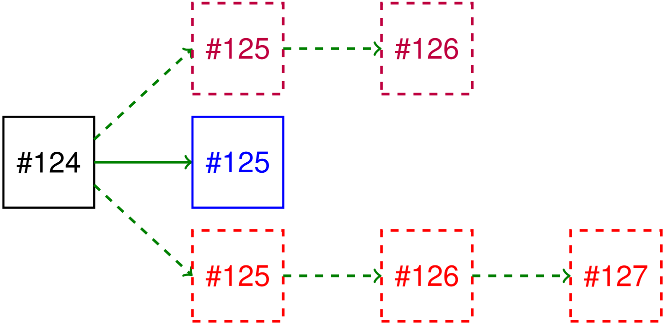
Two Selfish Miners (Case 4)

An honest miner *Henry* (r_h) and two selfish miners *Alice*(r_a) and *Bob*(r_b)



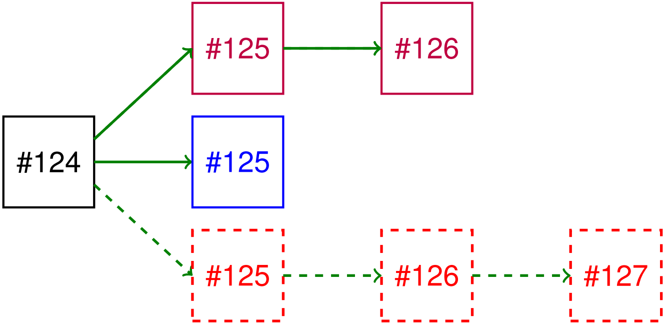
Two Selfish Miners (Case 4)

An honest miner *Henry* (r_h) and two selfish miners *Alice*(r_a) and *Bob*(r_b)



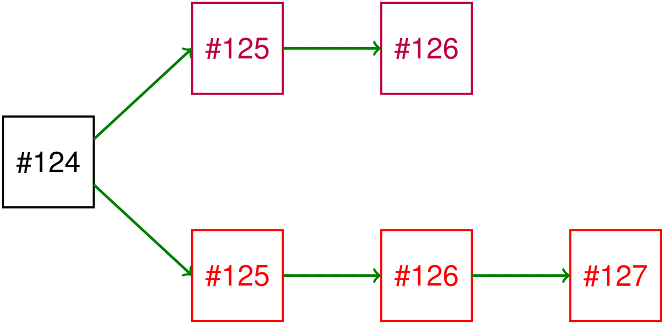
Two Selfish Miners (Case 4)

An honest miner *Henry* (r_h) and two selfish miners *Alice*(r_a) and *Bob*(r_b)



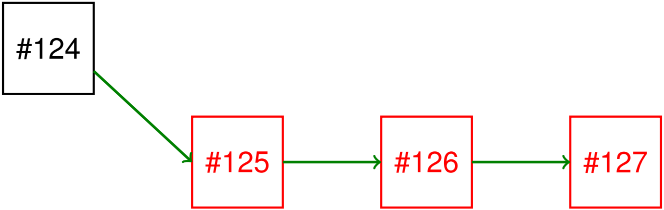
Two Selfish Miners (Case 4)

An honest miner *Henry (r_h)* and two selfish miners *Alice(r_a)* and *Bob(r_b)*



Two Selfish Miners (Case 4)

An honest miner *Henry (r_h)* and two selfish miners *Alice(r_a)* and *Bob(r_b)*



Longest chain rule shall be applied.

An Accurate Analytical Model

- S.W. Wang and S.S. Tzeng, ”An Accurate Analytical Model for A Proof-of-Work Blockchain with Multiple Selfish Miners,” in *2024 IEEE International Conference on Communications (ICC)* Denver, Colorado, USA, June 9-13, 2024

Motivations & Contributions

- Previous works use simulations to study the interesting properties of earned rewards
  - Time consuming
  - Lack of theoretical contributions
- An analytical model to calculate the rewards earned by different miners is much more desirable

Motivations & Contributions

- Previous works use simulations to study the interesting properties of earned rewards
  - Time consuming
  - Lack of theoretical contributions
- An analytical model to calculate the rewards earned by different miners is much more desirable

The Question

Can we efficiently and accurately calculate the reward earned by each miner in a blockchain with two selfish miners?

Motivations & Contributions

- Previous works use simulations to study the interesting properties of earned rewards
  - Time consuming
  - Lack of theoretical contributions
- An analytical model to calculate the rewards earned by different miners is much more desirable

The Question

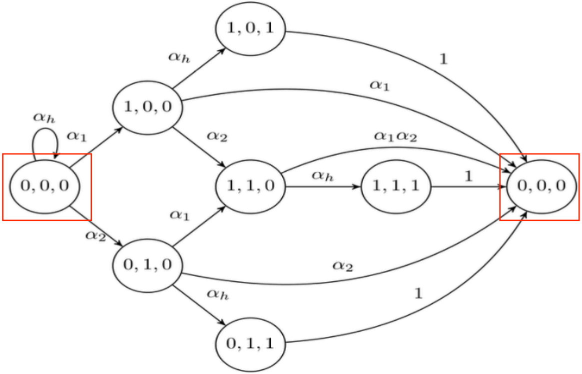
Can we efficiently and accurately calculate the reward earned by each miner in a blockchain with two selfish miners?

The Answer & Our Contribution

Yes. A closed-form expression with high accuracy is derived.

Previous Work: Two Selfish Miners

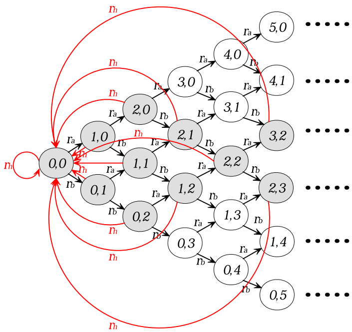
- Analytical Model Proposed by *Q. Bai*, and *et al.*



  - Two states with the same definition
  - Not very accurate because some states are ignored

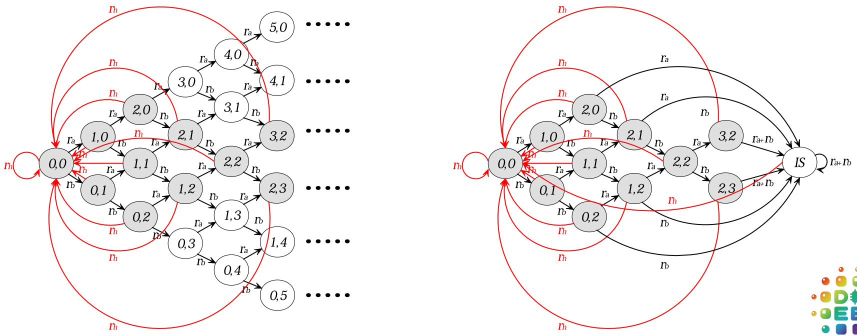
The Proposed Analytical Model

- State (n_a, n_b): Alice and Bob have their private branches with n_a and n_b blocks respectively
- End-of-Selfish (ES) states and In-Selfish (IS) states



The Proposed Analytical Model

- State (n_a, n_b): Alice and Bob have their private branches with n_a and n_b blocks respectively
- End-of-Selfish (ES) states and In-Selfish (IS) states



Steady-State Probabilities

- Steady state probability of an ES state (n_a, n_b):

$$\pi_{n_a,n_b} = \binom{n_a+n_b}{n_a} {r_a}^{n_a} \binom{n_b}{n_b} {r_b}^{n_b} \pi_{0,0}$$

- Steady state probability of an IS state:

$$\pi_{IS} = r_a(\pi_{2,0} + \pi_{2,1} + \pi_{3,2} + \pi_{2,3}) + r_b(\pi_{0,2} + \pi_{1,2} + \pi_{3,2} + \pi_{2,3})$$

- Sum of the probabilities equals to 1 where $\pi_{0,0}$ can be easily obtained.

$$\pi_{IS} + \sum_{s \in ES} \pi_s = 1$$

- Closed-form expressions are obtained

Our Model: End-of-Selfish (ES) and In-Selfish (IS) States

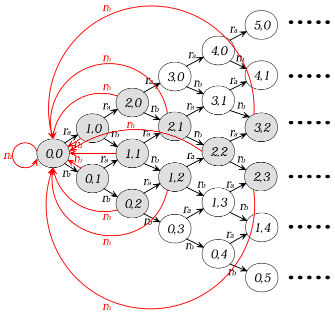


Figure 1: Exact Model

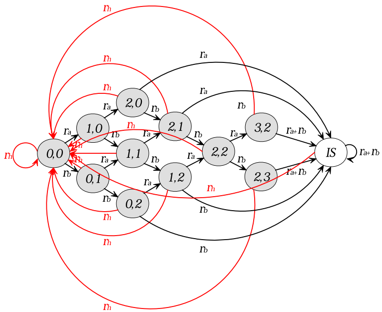
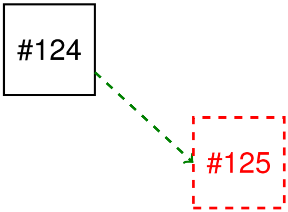
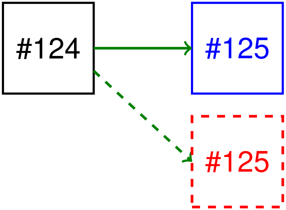


Figure 2: Approximate Model

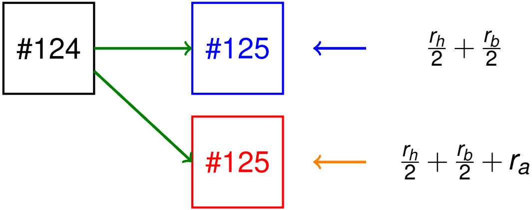
State $(0,1)$ and $(1,0)$



State $(0,1)$ and $(1,0)$

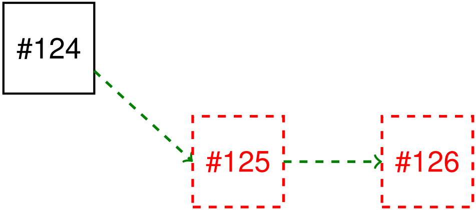


State $(0,1)$ and $(1,0)$

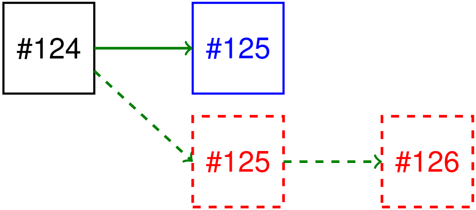


State s	Alice $R_a(s)$	Bob $R_b(s)$	Henry $R_h(s)$
(1,0)	$2r_a + r_b/2 + r_h/2$	r_b	$3r_h/2 + r_b/2$
(0,1)	r_a	$r_a/2 + 2r_b + r_h/2$	$3r_h/2 + r_a/2$

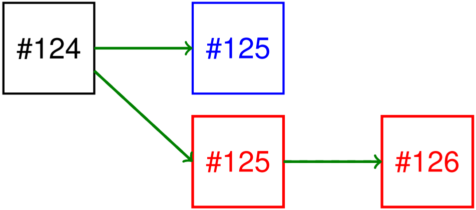
State $(0, 2)$ and $(2, 0)$



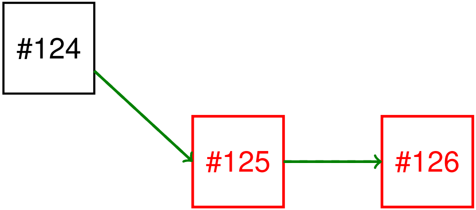
State $(0, 2)$ and $(2, 0)$



State $(0, 2)$ and $(2, 0)$



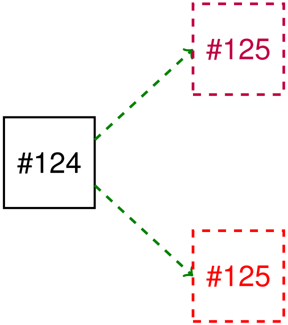
State $(0, 2)$ and $(2, 0)$



State s	Alice $R_a(s)$	Bob $R_b(s)$	Henry $R_h(s)$
(2,0)	2	0	0
(0,2)	0	2	0

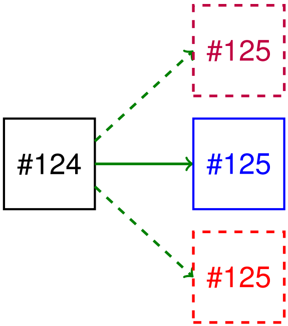
State $(1, 1)$

Honest miner *Henry (r_h)* and Selfish Miners *Alice(r_a)* and *Bob(r_b)*



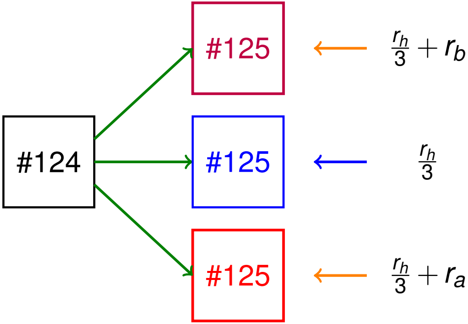
State $(1,1)$

Honest miner *Henry (r_h)* and Selfish Miners *Alice(r_a)* and *Bob(r_b)*



State $(1,1)$

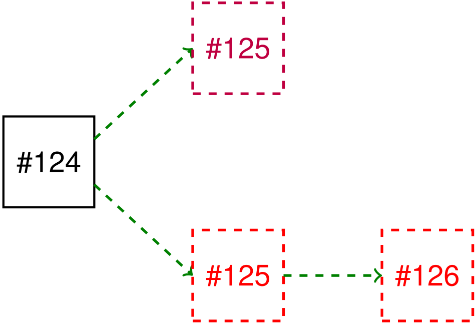
Honest miner *Henry* (r_h) and Selfish Miners *Alice*(r_a) and *Bob*(r_b)



State s	Alice $R_a(s)$	Bob $R_b(s)$	Henry $R_h(s)$
(1,1)	$2r_a + r_h/3$	$2r_b + r_h/3$	$4r_h/3$

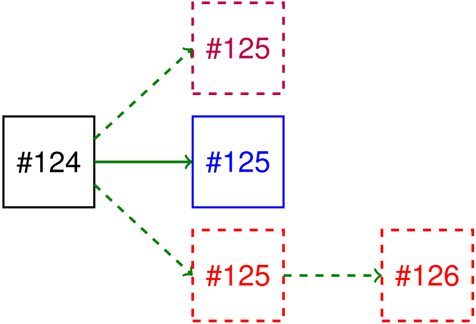
State $(1, 2)$ and $(2, 1)$

Honest miner *Henry* (r_h) and Selfish Miners *Alice*(r_a) and *Bob*(r_b)



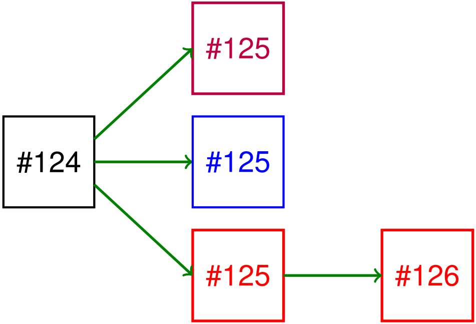
State $(1, 2)$ and $(2, 1)$

Honest miner *Henry (r_h)* and Selfish Miners *Alice(r_a)* and *Bob(r_b)*



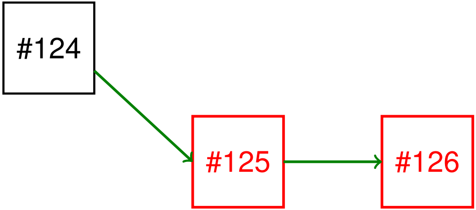
State $(1,2)$ and $(2,1)$

Honest miner *Henry (r_h)* and Selfish Miners *Alice(r_a)* and *Bob(r_b)*



State $(1,2)$ and $(2,1)$

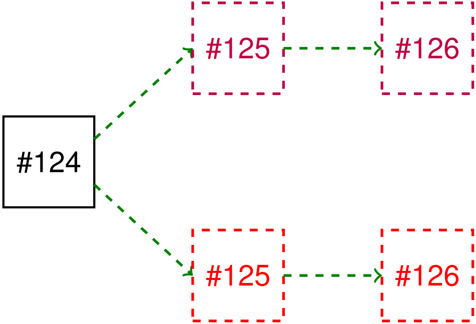
Honest miner *Henry (r_h)* and Selfish Miners *Alice(r_a)* and *Bob(r_b)*



State s	Alice $R_a(s)$	Bob $R_b(s)$	Henry $R_h(s)$
(2,1)	2	0	0
(1,2)	0	2	0

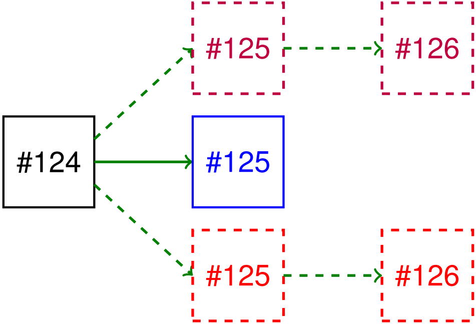
State $(2, 2)$

Honest miner *Henry (r_h)* and Selfish Miners *Alice(r_a)* and *Bob(r_b)*



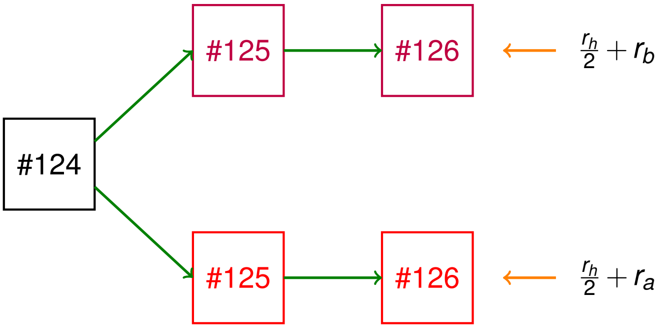
State $(2, 2)$

Honest miner *Henry* (r_h) and Selfish Miners *Alice*(r_a) and *Bob*(r_b)



State $(2,2)$

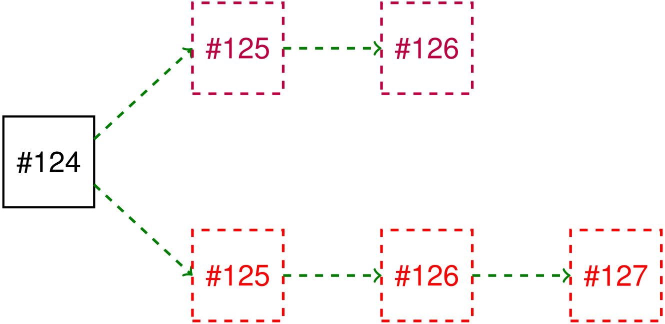
Honest miner *Henry* (r_h) and Selfish Miners *Alice*(r_a) and *Bob*(r_b)



State s	Alice $R_a(s)$	Bob $R_b(s)$	Henry $R_h(s)$
(2,2)	$3r_a + r_h$	$3r_b + r_h$	r_h

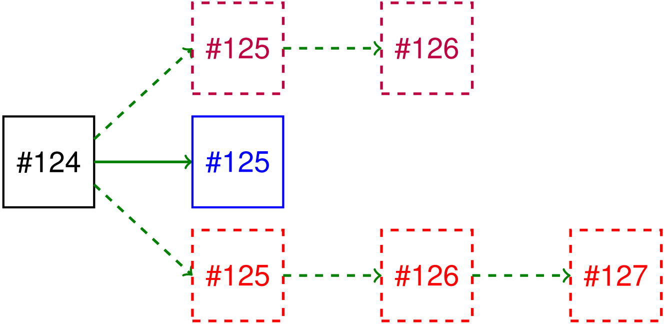
State $(2,3)$ and $(3,2)$

Honest miner *Henry (r_h)* and Selfish Miners *Alice(r_a)* and *Bob(r_b)*



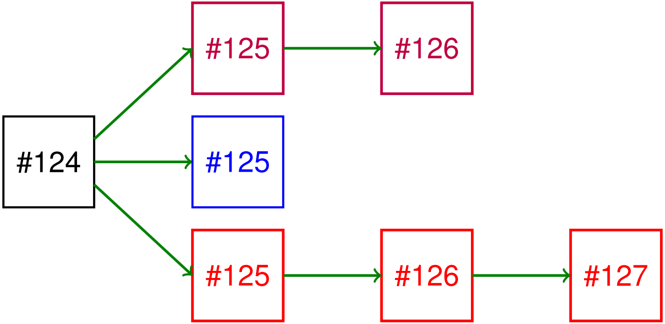
State $(2,3)$ and $(3,2)$

Honest miner *Henry* (r_h) and Selfish Miners *Alice*(r_a) and *Bob*(r_b)



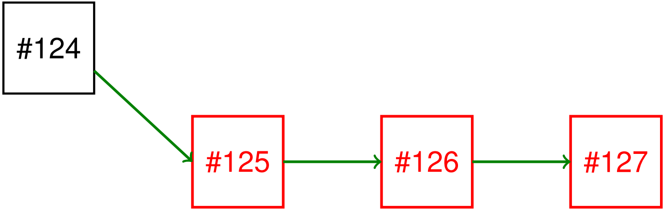
State $(2,3)$ and $(3,2)$

Honest miner *Henry* (r_h) and Selfish Miners *Alice*(r_a) and *Bob*(r_b)



State $(2,3)$ and $(3,2)$

Honest miner *Henry* (r_h) and Selfish Miners *Alice*(r_a) and *Bob*(r_b)



State s	Alice $R_a(s)$	Bob $R_b(s)$	Henry $R_h(s)$
(3,2)	3	0	0
(2,3)	0	3	0

IS State

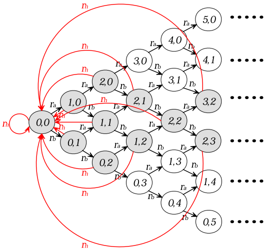


Figure 3: Exact Model

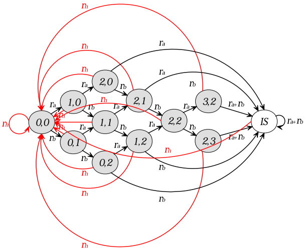


Figure 4: Approximate Model

State s	Alice $R_a(s)$	Bob $R_b(s)$	Henry $R_h(s)$
IS	$3r_a^3/(r_a^3+r_b^3)$	$3r_b^3/(r_a^3+r_b^3)$	0

Our Model: Expected Earned Rewards

State s	Alice $R_a(s)$	Bob $R_b(s)$	Henry $R_h(s)$
(0,0)	0	0	1
(1,0)	$2r_a + r_b/2 + r_h/2$	r_b	$3r_h/2 + r_b/2$
(0,1)	r_a	$r_a/2 + 2r_b + r_h/2$	$3r_h/2 + r_a/2$
(2,0)	2	0	0
(0,2)	0	2	0
(1,1)	$2r_a + r_h/3$	$2r_b + r_h/3$	$4r_h/3$
(2,1)	2	0	0
(1,2)	0	2	0
(2,2)	$3r_a + r_h$	$3r_b + r_h$	r_h
(3,2)	3	0	0
(2,3)	0	3	0
IS	$3{r_a}^3/({r_a}^3 + {r_b}^3)$	$3{r_b}^3/({r_a}^3 + {r_b}^3)$	0

Our Model: Steady-State Probability

- Let π_{n_a,n_b} be the steady-state probability of state (n_a, n_b).

$$\pi_{n_a,n_b} = \binom{n_a + n_b}{n_b} r_a{}^{n_a} \; r_b{}^{n_b} \; \pi_{0,0}$$

- π_{IS} can be calculated as follows.

$$\pi_{IS} = r_a(\pi_{2,0} + \pi_{2,1} + \pi_{3,2} + \pi_{2,3}) + r_b(\pi_{0,2} + \pi_{1,2} + \pi_{3,2} + \pi_{2,3})$$

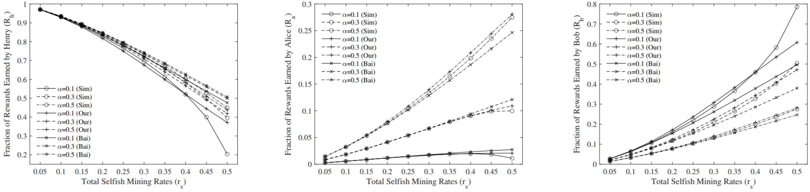
- Sum of the steady-state probabilities equals to 1.

$$\pi_{IS} + \sum_{s \in ES} \pi_s = 1 \qquad (1)$$

  where $\pi_{0,0}$ can be easily obtained.

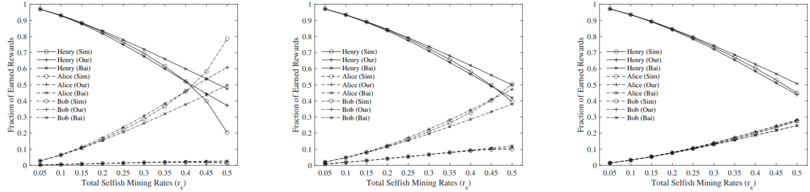
- The steady-state probability and expected earned rewards can be expressed in a closed-form of r_a, r_b, and r_h.

Numerical Results



(a) Henry's Rewards (b) Alice's Rewards (c) Bob's Rewards

Fig. 3: Fractions of rewards earned by Henry, Alice, and Bob



(a) $\alpha = 0.1$ (b) $\alpha = 0.3$ (c) $\alpha = 0.5$

Fig. 4: Fractions of rewards earned with different values of α

Our Model: Extension to Multiple Selfish Miners

Step. 1 Use n-tuple states to describe the blockchain with n selfish miners

Step. 2 Identify the ES states and IS states

Step. 3 For each ES states, calculate the expected rewards

Step. 4 For IS states, merge them into one single state and approximate the expected rewards

Step. 5 Calculate steady-state probability

Step. 6 Calculate the fractions of earned rewards

Conclusions

- An accurate analytical model for Proof-of-Work blockchain with two selfish miners is proposed
- Except the situation when there is a selfish miner with dominant mining rate, the maximum percentage of differences is 4.98%
- Our proposed analytical model performs closer to the simulation results than previous approach

Rational Mining Strategy

- S.W. Wang, ”A Game Theory Based Rational Mining Strategy in Blockchains With Multiple Rational Miners,” in *2024 International Conference on Computing Networking and Communications (ICNC) (ICNC 2024)*, Big Island, Hawaii, USA, 2024.

Rational Miners

- If a miner is rational, he may choose honest rather than selfish mining strategy in order to earn more rewards if his mining rate is not large enough
- In a blockchain with a single rational miner and all others are honest miners, it has been shown that the miner can be profitable if the fraction of his mining rate is larger than 25%
- Rational Mining in a blockchain with a single rational miner:
  - fraction of mining rate > 0.25: selfish mining
  - fraction of mining rate < 0.25: honest mining

Rational Miners

- Blockchains with two (2) rational miners are investigated
- Analytical models are employed
- Two selfish miners *Alice* and *Bob* are independent without knowing each other
- Payoff matrices with mining rates between 0.1 and 0.5

Rewards (Alice, Bob)		Bob's Strategy	
		Honest	Selfish
Alice's Strategy	Honest	R_a^{HH}, R_b^{HH}	R_a^{HS}, R_b^{HS}
	Selfish	R_a^{SH}, R_b^{SH}	R_a^{SS}, R_b^{SS}

Calculations of Earned Rewards

Rewards (Alice, Bob)		Bob's Strategy	
		Honest	Selfish
Alice's Strategy	Honest	R_a^{HH}, R_b^{HH}	R_a^{HS}, R_b^{HS}
	Selfish	R_a^{SH}, R_b^{SH}	R_a^{SS}, R_b^{SS}

[1] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang and Q. Kong, "A Deep Dive Into Blockchain Selfish Mining," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1-6.

Calculations of Earned Rewards

Rewards (Alice, Bob)		Bob's Strategy: Honest	Bob's Strategy: Selfish
Alice's Strategy	Honest	R_a^{HH}, R_b^{HH}	R_a^{HS}, R_b^{HS}
	Selfish	R_a^{SH}, R_b^{SH}	R_a^{SS}, R_b^{SS}

- R_a^{HH} and R_b^{HH}: Proportional to their mining rates

[1] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang and Q. Kong, "A Deep Dive Into Blockchain Selfish Mining," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1-6.

Calculations of Earned Rewards

Rewards (Alice, Bob)		Bob's Strategy	
		Honest	Selfish
Alice's Strategy	Honest	R_a^{HH}, R_b^{HH}	R_a^{HS}, R_b^{HS}
	Selfish	R_a^{SH}, R_b^{SH}	R_a^{SS}, R_b^{SS}

- R_a^{HH} and R_b^{HH}: Proportional to their mining rates
- R_a^{HS},R_b^{HS}, R_a^{SH},R_b^{SH}: Only one single selfish miner

[1] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang and Q. Kong, "A Deep Dive Into Blockchain Selfish Mining," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1-6.

Calculations of Earned Rewards

Rewards (Alice, Bob)		Bob's Strategy: Honest	Bob's Strategy: Selfish
Alice's Strategy	Honest	R_a^{HH}, R_b^{HH}	R_a^{HS}, R_b^{HS}
	Selfish	R_a^{SH}, R_b^{SH}	R_a^{SS}, R_b^{SS}

- R_a^{HH} and R_b^{HH}: Proportional to their mining rates
- R_a^{HS}, R_b^{HS}, R_a^{SH}, R_b^{SH}: Only one single selfish miner
  - Selfish miner: Earns rewards by *RW* function

[1] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang and Q. Kong, "A Deep Dive Into Blockchain Selfish Mining," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1-6.

Calculations of Earned Rewards

Rewards (Alice, Bob)		Bob's Strategy: Honest	Bob's Strategy: Selfish
Alice's Strategy	Honest	R_a^{HH}, R_b^{HH}	R_a^{HS}, R_b^{HS}
	Selfish	R_a^{SH}, R_b^{SH}	R_a^{SS}, R_b^{SS}

- R_a^{HH} and R_b^{HH}: Proportional to their mining rates
- R_a^{HS},R_b^{HS}, R_a^{SH},R_b^{SH}: Only one single selfish miner
  - Selfish miner: Earns rewards by *RW* function
  - Honest miner: Shares the remaining rewards with Henry

[1] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang and Q. Kong, "A Deep Dive Into Blockchain Selfish Mining," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1-6.

Calculations of Earned Rewards

Rewards (Alice, Bob)		Bob's Strategy	
		Honest	Selfish
Alice's Strategy	Honest	R_a^{HH}, R_b^{HH}	R_a^{HS}, R_b^{HS}
	Selfish	R_a^{SH}, R_b^{SH}	R_a^{SS}, R_b^{SS}

- R_a^{HH} and R_b^{HH}: Proportional to their mining rates
- R_a^{HS},R_b^{HS}, R_a^{SH},R_b^{SH}: Only one single selfish miner
  - Selfish miner: Earns rewards by *RW* function
  - Honest miner: Shares the remaining rewards with Henry
- R_a^{SS},R_b^{SS}: By an analytical model proposed by *Bai, et. al.*[1]

[1] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang and Q. Kong, "A Deep Dive Into Blockchain Selfish Mining," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1-6.

Payoff Matrices

Payoff Matrices			Bob's Strategy							
		r_b	0.1		0.2		0.3		0.4	
	r_a		Honest	Selfish	Honest	Selfish	Honest	Selfish	Honest	Selfish
Alice's Strategy	0.1	Honest	**0.100,0.100**	**0.103**,0.072	**0.100,0.200**	**0.102**,0.182	**0.100**,0.300	**0.096,0.327**	**0.100**,0.400	**0.079,0.526**
		Selfish	0.072,**0.103**	0.078,0.078	0.072,**0.206**	0.079,0.199	0.072,0.309	0.068,**0.359**	0.072,0.412	0.054,**0.550**
	0.2	Honest	**0.200,0.100**	**0.206**,0.072	**0.200,0.200**	**0.204**,0.182	**0.200**,0.300	**0.192,0.327**	**0.200**,0.400	**0.158,0.526**
		Selfish	0.182,**0.102**	0.199,0.076	0.182,**0.204**	0.199,0.199	0.182,0.307	0.177,**0.369**	0.182,0.409	0.136,**0.574**
	0.3	Honest	0.300,**0.100**	0.309,0.072	0.300,**0.200**	0.307,0.182	0.300,0.300	0.289,**0.327**	0.300,0.400	0.237,**0.526**
		Selfish	**0.327,0.096**	**0.359**,0.068	**0.327,0.192**	**0.369**,0.177	**0.327**,0.289	**0.341,0.341**	**0.327**,0.385	**0.262,0.555**
	0.4	Honest	0.400,**0.100**	0.412,0.072	0.400,**0.200**	0.409,0.182	0.400,0.300	0.385,**0.327**	0.400,0.400	0.316,**0.526**
		Selfish	**0.526,0.079**	**0.550**,0.054	**0.526,0.158**	**0.574**,0.136	**0.526**,0.237	**0.555,0.262**	**0.526**,0.316	**0.455,0.455**

- All above payoff matrices have only one Nash equilibrium
- Find more details when mining rate is between 0.2 and 0.3

Payoff Matrices

Payoff Matrices			Bob's Strategy							
		r_b	0.20		0.21		0.22		0.23	
	r_a		Honest	Selfish	Honest	Selfish	Honest	Selfish	Honest	Selfish
Alice's Strategy	0.20	Honest	**0.200,0.200**	**0.204**,0.182	**0.200,0.210**	**0.204**,0.195	**0.200,0.220**	**0.203**,0.209	**0.200,0.230**	**0.202**,0.222
		Selfish	0.182,**0.204**	0.199,0.199	0.182,**0.215**	0.198,0.214	0.182,0.225	0.198,**0.230**	0.182,0.235	0.196,**0.246**
	0.21	Honest	**0.210,0.200**	**0.215**,0.182	**0.210,0.210**	**0.214**,0.195	**0.210,0.220**	**0.213**,0.209	**0.210,0.230**	**0.212**,0.222
		Selfish	0.195,**0.204**	0.214,0.198	0.195,**0.214**	0.213,0.213	0.195,0.224	0.212,**0.229**	0.195,0.234	0.211,**0.245**
	0.22	Honest	**0.220,0.200**	0.225,0.182	**0.220,0.210**	0.224,0.195	**0.220,0.220**	0.223,0.209	**0.220,0.230**	0.222,0.222
		Selfish	0.209,**0.203**	**0.229**,0.197	0.209,**0.213**	**0.229**,0.212	0.208,0.223	**0.228,0.228**	0.208,0.233	**0.227,0.245**
	0.23	Honest	**0.230,0.200**	0.235,0.182	**0.230,0.210**	0.234,0.195	**0.230,0.220**	0.233,0.209	**0.230,0.230**	0.232,0.222
		Selfish	0.222,**0.202**	**0.246**,0.196	0.222,**0.212**	**0.245**,0.211	0.222,0.222	**0.244,0.227**	0.222,0.232	**0.243,0.243**
	0.24	Honest	**0.240,0.200**	0.245,0.182	**0.240,0.210**	0.244,0.195	**0.240,0.220**	0.244,0.209	**0.240,0.230**	0.243,0.222
		Selfish	0.236,**0.201**	**0.263**,0.195	0.236,**0.211**	**0.262**,0.210	0.236,0.221	**0.262,0.225**	0.236,0.231	**0.261,0.242**
	0.25	Honest	**0.250,0.200**	0.256,0.182	**0.250,0.210**	0.255,0.195	**0.250,0.220**	0.254,0.209	**0.250,0.230**	0.253,0.222
		Selfish	**0.250,0.200**	**0.281**,0.193	**0.250,0.210**	**0.281**,0.208	**0.250**,0.220	**0.280,0.223**	**0.250**,0.230	**0.279,0.240**
	0.26	Honest	0.260,**0.200**	0.266,0.182	0.260,**0.210**	0.265,0.195	0.260,**0.220**	0.264,0.209	0.260,**0.230**	0.263,0.222
		Selfish	**0.265,0.299**	**0.300**,0.191	**0.265,0.209**	**0.300**,0.205	**0.265**,0.219	**0.299,0.221**	**0.265**,0.229	**0.299,0.237**
	0.27	Honest	0.270,**0.200**	0.276,0.182	0.270,**0.210**	0.275,0.195	0.270,**0.220**	0.274,0.209	0.270,**0.230**	0.273,0.222
		Selfish	**0.280,0.197**	**0.319**,0.189	**0.280,0.207**	**0.320**,0.203	**0.280**,0.217	**0.320,0.218**	**0.280**,0.227	**0.320,0.234**

Payoff Matrices

Payoff Matrices		r_b	Bob's Strategy							
			0.24		0.25		0.26		0.27	
	r_a		Honest	Selfish	Honest	Selfish	Honest	Selfish	Honest	Selfish
Alice's Strategy	0.20	Honest	**0.200,0.240**	**0.201**,0.236	**0.200,0.250**	**0.200,0.250**	**0.200**,0.260	**0.199,0.265**	**0.200**,0.270	**0.197,0.280**
		Selfish	0.182,0.245	0.195,**0.263**	0.182,0.256	0.193,**0.281**	0.182,0.266	0.191,**0.300**	0.182,0.279	0.188,**0.319**
	0.21	Honest	**0.210,0.240**	**0.211**,0.236	**0.210,0.250**	**0.210,0.250**	**0.210**,0.260	**0.209,0.265**	**0.210**,0.270	**0.207,0.280**
		Selfish	0.195,0.244	0.210,**0.263**	0.195,0.255	0.208,**0.281**	0.195,0.265	0.206,**0.300**	0.195,0.275	0.203,**0.320**
	0.22	Honest	**0.220,0.240**	0.221,0.236	**0.220,0.250**	0.220,**0.250**	**0.220**,0.260	0.219,**0.265**	**0.220**,0.270	0.217,**0.280**
		Selfish	0.209,0.244	**0.225,0.262**	0.209,0.254	**0.224,0.280**	0.209,0.264	**0.221,0.299**	0.209,0.274	**0.218,0.320**
	0.23	Honest	**0.230,0.240**	0.231,0.236	**0.230,0.250**	0.230,**0.250**	**0.230**,0.260	0.229,**0.265**	**0.230**,0.270	0.227,**0.280**
		Selfish	0.222,0.243	**0.242,0.261**	0.222,0.253	**0.240,0.279**	0.222,0.263	**0.238,0.299**	0.222,0.273	**0.234,0.320**
	0.24	Honest	**0.240,0.240**	0.241,0.236	**0.240,0.250**	0.240,**0.250**	**0.240**,0.260	0.238,**0.265**	**0.240**,0.270	0.237,**0.280**
		Selfish	0.236,0.241	**0.259,0.250**	0.236,0.251	**0.257,0.278**	0.236,0.261	**0.255,0.297**	0.236,0.272	**0.252,0.319**
	0.25	Honest	**0.250,0.240**	0.251,0.236	**0.250,0.250**	0.250,**0.250**	**0.250**,0.260	0.248,**0.265**	**0.250**,0.270	0.247,**0.280**
		Selfish	**0.250**,0.240	**0.278,0.257**	**0.250**,0,250	**0.276,0,276**	**0.250**,0.260	**0.274,0.295**	**0.250**,0.270	**0.270,0.318**
	0.26	Honest	0.260,**0.240**	0.261,0.236	0.260,**0.250**	0.260,**0.250**	0.260,0.260	0.258,**0.265**	0.260,0.270	0.257,**0.280**
		Selfish	**0.265**,0.238	**0.298,0.255**	**0.265**,0.248	**0.296,0.274**	**0.265**,0.258	**0.293,0.293**	**0.265**,0.268	**0.290,0.315**
	0.27	Honest	0.270,**0.240**	0.272,0.236	0.270,**0.250**	0.270,**0.250**	0.270,0.260	0.268,**0.265**	0.270,0.270	0.266,**0.280**
		Selfish	**0.280**,0.237	**0.319,0.252**	**0.280**,0.247	**0.317,0.270**	**0.280**,0.257	**0.315,0.290**	**0.280**,0.266	**0.312,0.312**

Payoff Matrices: Two miners both have dominant strategies

Rewards (Alice, Bob)		Bob $r_b = 0.1$	
		Honest	Selfish
Alice $r_a = 0.2$	Honest	**0.200**,**0.100**	**0.206**,0.072
	Selfish	0.182,**0.102**	0.199,0.076

Rewards (Alice, Bob)		Bob $r_b = 0.4$	
		Honest	Selfish
Alice $r_a = 0.3$	Honest	0.300,0.400	0.237,**0.526**
	Selfish	**0.327**,0.385	**0.262**,**0.555**

- Mining rate > 0.25: Selfish mining strategy
- Mining rate < 0.22: Honest mining strategy

Payoff Matrices: Only one miner has dominant strategy

Rewards (Alice, Bob)		Bob $r_b = 0.21$ Honest	Bob $r_b = 0.21$ Selfish
Alice $r_a = 0.24$	Honest	**0.240**,**0.210**	0.244,0.195
	Selfish	0.236,**0.211**	**0.262**,0.210

Rewards (Alice, Bob)		Bob $r_b = 0.27$ Honest	Bob $r_b = 0.27$ Selfish
Alice $r_a = 0.23$	Honest	**0.230**,0.270	0.227,**0.280**
	Selfish	0.222,0.273	**0.234**,**0.320**

- Mining rate between $[0.22, 0.25]$: Follow the other rational miner's strategy if he has dominant strategy

Payoff Matrices: No miner has dominant strategy

Rewards (Alice, Bob)		Bob $r_b = 0.24$ Honest	Bob $r_b = 0.24$ Selfish
Alice $r_a = 0.23$	Honest	**0.230**,**0.240**	0.231,0.236
	Selfish	0.222,0.243	**0.242**,**0.261**

- Two Nash Equilibria exist
- Mixed strategy can be applied
- Select a strategy according to a probability distribution

Mixed Strategy

Rewards (Alice, Bob)		Bob's Strategy: Honest(q)	Bob's Strategy: Selfish $(1-q)$
Alice's Strategy	Honest (p)	R_a^{HH}, R_b^{HH}	R_a^{HS}, R_b^{HS}
	Selfish $(1-p)$	R_a^{SH}, R_b^{SH}	R_a^{SS}, R_b^{SS}

Main idea: to make the other miner earn *indifferent* rewards no matter which strategy the other miner uses.

- Using honest mining, Bob earns $p \times R_b^{HH} + (1-p) \times R_b^{SH}$.
- Using selfish mining, Bob earns $p \times R_b^{HS} + (1-p) \times R_b^{SS}$.
- Solve equation $p \times R_b^{HH} + (1-p) \times R_b^{SH} = p \times R_b^{HS} + (1-p) \times R_b^{SS}$

Mixed Strategy

Rewards (Alice, Bob)		Bob's Strategy	
		Honest(q)	Selfish $(1-q)$
Alice's Strategy	Honest (p)	R_a^{HH}, R_b^{HH}	R_a^{HS}, R_b^{HS}
	Selfish $(1-p)$	R_a^{SH}, R_b^{SH}	R_a^{SS}, R_b^{SS}

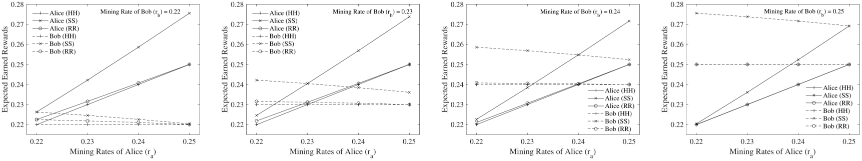
Main idea: to make the other miner earn *indifferent* rewards no matter which strategy the other miner uses.

- $p = \frac{R_b^{SS} - R_b^{SH}}{R_b^{HH} + R_b^{SS} - R_b^{SH} - R_b^{HS}}$
- $q = \frac{R_a^{SS} - R_a^{SH}}{R_a^{HH} + R_a^{SS} - R_a^{SH} - R_a^{HS}}$

Rational Mining Strategy with Two Rational Miners

- If mining rate is < 0.22, use Honest Mining
- If mining rate is > 0.25, use Selfish Mining
- If mining rate ranges from 0.22 to 0.25,
  - If the other miner has dominant strategy, follow his dominant mining strategy
  - If the other miner has no dominant strategy, solve the payoff matrices according to the probability distribution

Numerical Results



(a) $r_b = 0.22$ (b) $r_b = 0.23$ (c) $r_b = 0.24$ (d) $r_b = 0.25$

Fig. 1: Expected rewards earned by Alice and Bob under both honest, selfish, or rational mining strategies

- Both honest (HH) $\leq$ both rational (RR) $\leq$ both selfish (SS)
- Mixed strategy performs close to honest strategy

Ongoing Research Works

- A Stochastic Lightweight Blockchain
- Security Analysis of Sharded Blockchain

Ongoing Research Works

- A Stochastic Lightweight Blockchain
- Security Analysis of Sharded Blockchain

Motivations

- High computational power consumption is a crucial problem in a Proof-of-Work blockchain
- Reducing the number of miners raises security problem
- We propose a stochastic lightweight blockchain called *SLChain* which is able:
  - to reduce power consumption
  - to maintain fairness among all miners
  - to maintain robustness of majority attack
  - to maintain block time consistency
  - to mitigate the selfish mining attacks

  simultaneously.

SLChain: A Stochastic Lightweight Blockchain

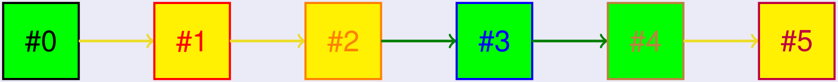
- Lightweight: Only a subset of miners is entitled to mine the next block
- Stochastic: The subset of entitled miners is randomly selected according to the hash value of previous block

SLChain: A Stochastic Lightweight Blockchain

1. Each miner is randomly assigned a unique miner id m_{id} which is an integer.
2. The hash value of previous block is an integer h.
3. A miner is entitled to mine the next block if the miner id m_{id} and the hash value of previous block h are congruent modulo G. That is,

$$m_{id} \equiv h \pmod{G}. \tag{2}$$

Example of SLChain when $G = 2$



Properties

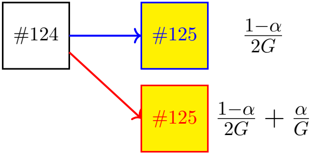
- N: the total number of miners
- G: the number of groups
- Computational power: $\frac{1}{G}$ of Nakamoto blockchain
- Fairness: The probability a miner mines the next block
  - Nakamoto blockchain: $\frac{1}{N}$
  - SLChain: $\frac{1}{G} \times \frac{1}{\frac{N}{G}} = \frac{1}{N}$
- Robustness to majority attack: The mined block shall be validated by **ALL** miners
- Block time consistency: the number of leading zeros shall be adjusted to $\lfloor z - \log_2 G \rfloor$ or $\lceil z - \log_2 G \rceil$ where z is the number of leading zeros in Nakamoto blockchain

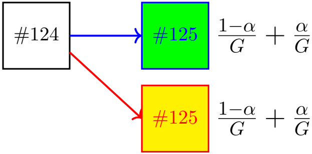
Selfish Mining in Nakamoto Blockchain

We have the following rewards in the competitive situation:

- $0 < \alpha < \frac{1}{2}$
- $R_h(\alpha) = 2 \times \frac{1-\alpha}{2} + 1 \times \frac{1-\alpha}{2} = \frac{3(1-\alpha)}{2}$
- $R_s(\alpha) = 2 \times \alpha + 1 \times \frac{1-\alpha}{2} = \frac{1+3\alpha}{2}$
- $R_h(\alpha) + R_s(\alpha) = 2$

Selfish Mining in SLChain





- $R_h^s(\alpha, G) = \frac{3(1-\alpha)}{2}$
- $R_s^s(\alpha, G) = \frac{1+3\alpha}{2}$
- Probability: $\frac{1}{G}$

- $R_h^d(\alpha, G) = \frac{3-2\alpha}{2}$
- $R_s^d(\alpha, G) = \frac{1+2\alpha}{2}$
- Probability: $\frac{G-1}{G}$

Expected Earned Rewards

- $R_h(\alpha, G) = \frac{1}{G} \times R_h^s + \frac{G-1}{G} \times R_h^d = \frac{3G-\alpha-2\alpha G}{2G}$
- $R_s(\alpha, G) = \frac{1}{G} \times R_s^s + \frac{G-1}{G} \times R_s^d = \frac{G+\alpha+2\alpha G}{2G}$

Mitigation of Selfish Mining Attacks

Theorem

The rewards earned by selfish miner in the proposed SLChain with $G \geq 2$ is less than that in Nakamoto blockchain. That is, $R_s(\alpha, G) < R_s(\alpha, 1)$ where $G \geq 2$ and $0 < \alpha < 1$.

Proof.

$$
\begin{aligned}
R_s(\alpha, G) - R_s(\alpha, 1) &= \frac{2\alpha G + G + \alpha}{2G} - \frac{1 + 3\alpha}{2} \\
&= \frac{2\alpha G + G + \alpha - 3\alpha G - G}{2G} \\
&= \frac{\alpha(1 - G)}{2G} < 0
\end{aligned}
$$

The last inequality holds since $G \geq 2$ and $\alpha > 0$. □

Earned Rewards in SLChain

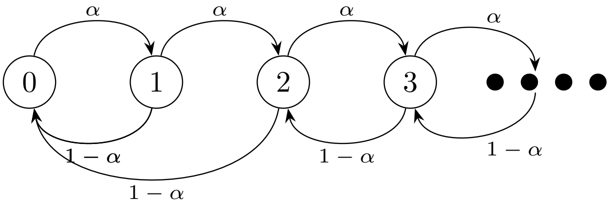


Figure 5: Markov chain of a blockchain with one selfish miner

$$
\begin{cases}
\pi_0 = \frac{1-2\alpha}{1-\alpha-\alpha^2} \\
\pi_1 = \frac{1-2\alpha}{1-\alpha-\alpha^2} \times \alpha \\
\pi_2 = \frac{1-2\alpha}{1-\alpha-\alpha^2} \times \frac{\alpha^2}{1-\alpha} \\
\pi_i = \frac{1-2\alpha}{1-\alpha-\alpha^2} \times \frac{\alpha^i}{(1-\alpha)^{i-1}} \quad \text{for all } i \geq 3
\end{cases}
$$

Rewards earned by the miners

Table 1: Expected rewards when the honest miner mined the block in each state of the SLChain

State i	Honest Miner	Selfish Miner	Total Rewards
$i = 0$	1	0	1
$i = 1$	$(3G - \alpha - 2\alpha G)/2G$	$(G + \alpha + 2\alpha G)/2G$	2
$i = 2$	0	2	2
$i \geq 3$	0	1	1

Expected rewards

- Honest miner: $RW_h(\alpha, G) = 1 \times \pi_0 + R_h(\alpha, G) \times \pi_1$
- Selfish Miner: $RW_s(\alpha, G) = R_s(\alpha, G) \times \pi_1 + 2 \times \pi_2 + 1 \times \sum_{i=3}^{\infty} \pi_i;$

Fraction of rewards earned by selfish miner

$$
\begin{aligned}
F_s(\alpha, G) &= \frac{RW_s(\alpha, G)}{RW_h(\alpha, G) + RW_s(\alpha, G)} \\
&= \frac{R_s(\alpha, G)\pi_1 + 2 \times \pi_2 + 1 \times \sum\limits_{i=3}^{\infty} \pi_i}{1 + \pi_1 + \pi_2} \\
&= \frac{(4G+2)\alpha^4 - (10G+3)\alpha^3 + (3G+1)\alpha^2 + G\alpha}{2G(\alpha^3 - 2\alpha^2 - \alpha + 1)}
\end{aligned}
$$

Profitable Threshold

Theorem

The value of profitable threshold $\bar{\alpha}(G)$ *is* $\frac{(4G+1)-\sqrt{8G^2+1}}{4(G+1)}$ *in the proposed SLChain with the number of groups of miners G and* $0 < \alpha < 0.5$.

Proof.

We try to solve the inequality:

$$\frac{(4G+2)\alpha^4 - (10G+3)\alpha^3 + (3G+1)\alpha^2 + G\alpha}{2G(\alpha^3 - 2\alpha^2 - \alpha + 1)} \geq \alpha. \tag{3}$$

We have

$$\frac{\alpha(\alpha-1)[(2G+2)\alpha^2 - (4G+1)\alpha + G]}{2G(\alpha^3 - 2\alpha^2 - \alpha + 1)} \geq 0. \tag{4}$$

□

Profitable Threshold

Proof.

Since $0 < \alpha < 0.5$, we have $\alpha(\alpha - 1) < 0$ and $\alpha^3 - 2\alpha^2 - \alpha + 1 > 0$.

Then, we solve the inequality:

$$(2G+2)\alpha^2 - (4G+1)\alpha + G \leq 0. \tag{5}$$

We have

$$\frac{(4G+1) - \sqrt{8G^2+1}}{4(G+1)} \leq \alpha < \frac{1}{2}. \tag{6}$$

The profitable threshold is $\frac{(4G+1)-\sqrt{8G^2+1}}{4(G+1)}$.

Note: When $G = 1$, the profitable threshold is $\frac{4+1-\sqrt{8+1}}{4\times 2} = \frac{2}{8} = 25\%$. □

Upper Bounds of Profitable Threshold

Theorem

The upper bound of profitable threshold in the proposed SLChain is 29.29%.

Proof.

The profitable threshold $\bar{\alpha}(G)$ has been proved in previous theorem. When the number of G approaches to infinity, we have

$$\lim_{G \to \infty} \frac{(4G+1) - \sqrt{8G^2+1}}{4(G+1)} = \frac{4G - \sqrt{8G^2}}{4G} = \frac{4 - 2\sqrt{2}}{4} = 29.29\% \quad (7)$$

□

Simulation Results

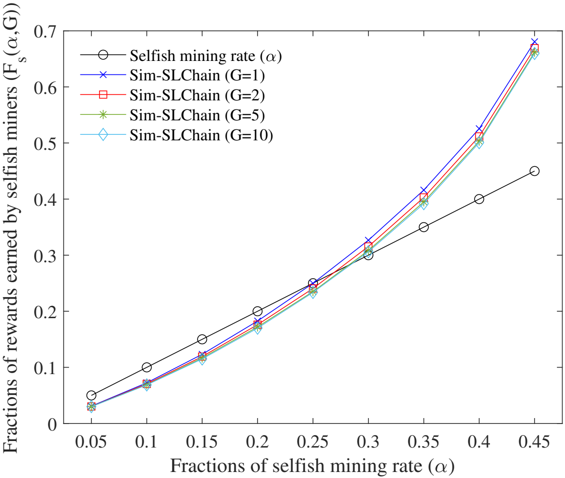


Figure 6: Fractions of rewards earned by selfish miner in SLChain

Simulation Results

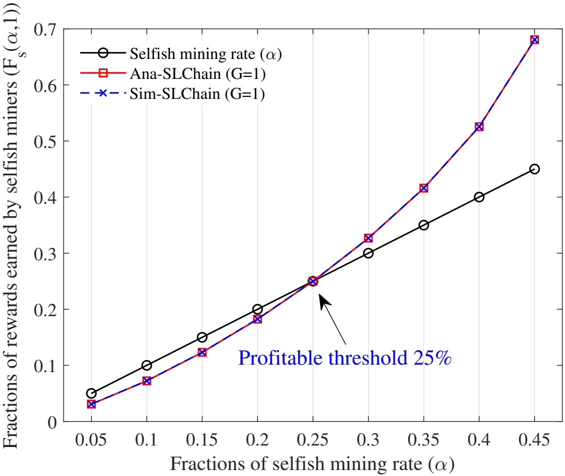


Figure 7: SLChain (G=1)

Simulation Results

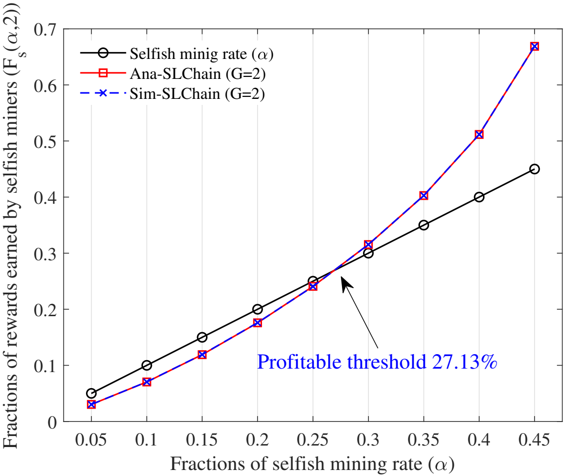


Figure 8: SLChain (G=2)

Simulation Results

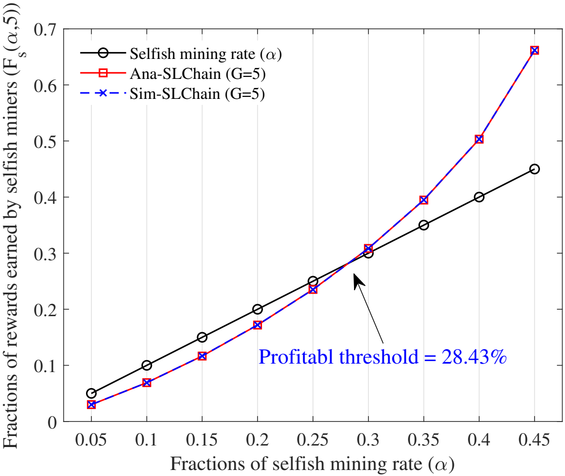


Figure 9: SLChain (G=5)

Simulation Results

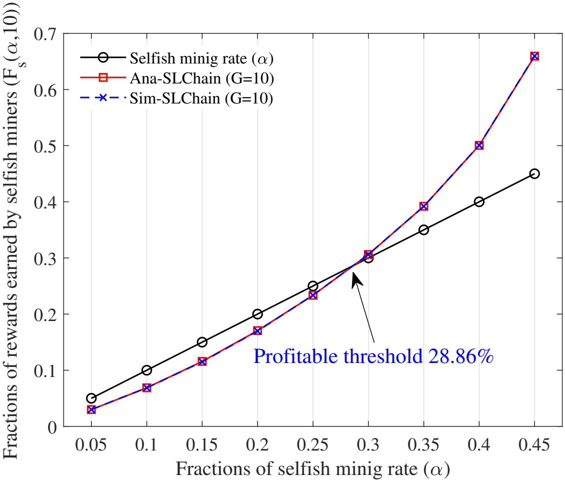


Figure 10: SLChain (G=10)

Simulation Results

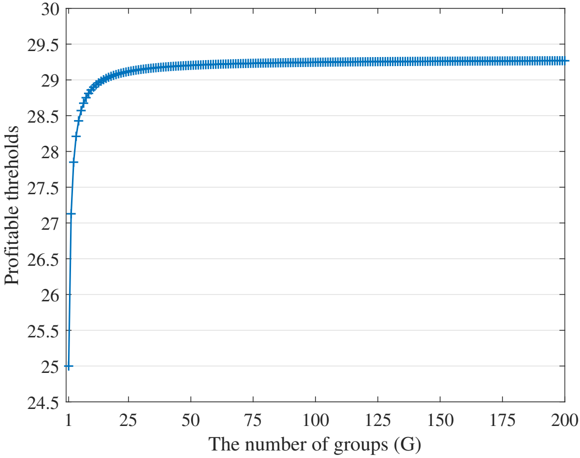


Figure 11: Profitable thresholds of SLChain

Conclusions

- A stochastic lightweight blockchain (SLChain) is proposed:
  - to reduce power consumption
  - to maintain fairness among miners
  - to maintain robustness of majority attack
  - to maintain block time consistency
  - to mitigate selfish mining attack

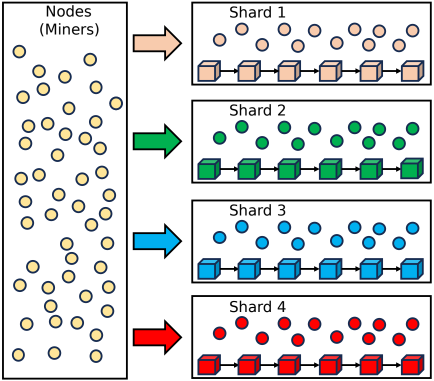
  simultaneously.
- An accurate analytical model is proposed to calculate the fraction of rewards earned by selfish miner
- Upper bounds of the profitable threshold is derived (29.29%)

Ongoing Research Works

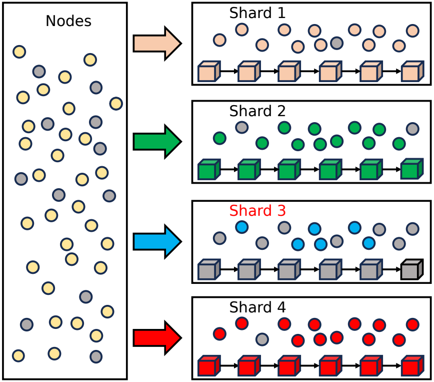
- A Stochastic Lightweight Blockchain
- Security Analysis of Sharded Blockchain

Motivations

- Scalability of Bitcoin blockchain: 7 transactions per second
- Sharding: dividing the nodes and transactions into different groups(shards)

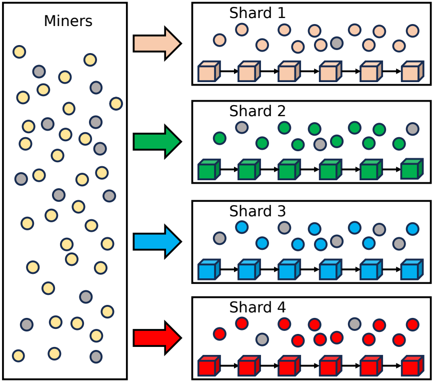


Majority Attacks



- Single Shard Takeover (SST): Blockchain fails if one shard is attacked
- Probability of SST (P_{SST}) is calculated

Selfish Mining Attacks



- Nakamoto blockchain: $RW(\frac{11}{48}) = 22.08\% < \frac{11}{48} = 22.92\%$ (Not profitable)
- Sharded blockchain:
  $\frac{1}{4}[RW(\frac{1}{12}) + RW(\frac{3}{12}) + RW(\frac{5}{12}) + RW(\frac{2}{12})] = 25.48\%$ (Profitable)

Questions

The Question

In a shard, the number of nodes/miners is much smaller than Nakamoto blockchain. Therefore, sharding mechanism reduces the security level of a blockchain, is it?

Our method

- We propose some analytical models to calculate:
  - Probability of SST (P_{SST}) for majority attack
  - Rewards earned by selfish miners (ER) for selfish mining attack
  - Profitable threshold (PT) for selfish mining attack
- We conduct simulations to verify the proposed analytical model
- We study the relationships between the three metrics and the number of shards

Simulation Results

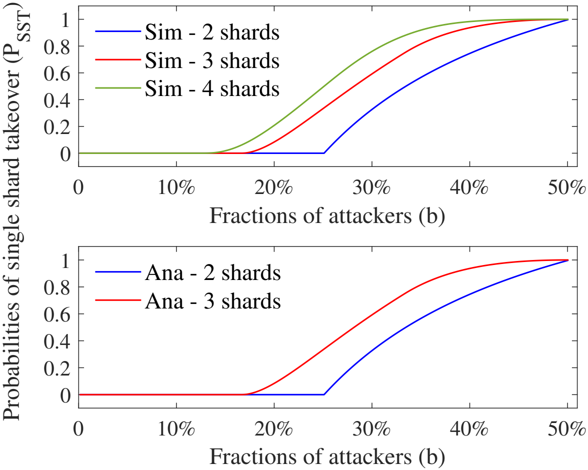


Figure 12: Majority attack in sharded blockchain

Simulation Results

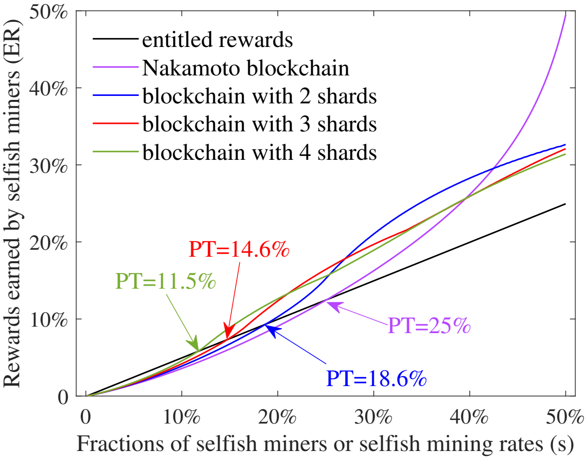


Figure 13: Selfish mining attack in sharded blockchain

Conclusions

The Question

Sharding mechanism reduces the security level of a blockchain, is it?

Answer

Not necessarily.

- Majority attack:
  - Probability of SST (P_{SST}): YES. P_{SST} increases with the number of shards.
- Selfish mining attack:
  - Rewards earned by selfish miners (*ER*): Not neccessarily.
  - Profitable threshold (*PT*): YES. *PT* decreases with the number of shards.

Thank you!

swwang@nuu.edu.tw