TRUTH JUSTICE ALGOS

Cryptocurrencies II: Selfish Mining

Teachers: Ariel Procaccia and Alex Psomas (this time)

- Last time:
  - Basic concepts
  - Double spend attack
- Today: Block withholding attacks (Selfish mining)
  - Get a taste of some AGT works on cryptocurrencies

SETUP

- Each miner i has mining power p_i
- $\sum_{i=1}^{n} p_i = 1$
- Each miner chooses a chain to mine on top of, and find a block after a random time t distributed (according to an exponential random variable with mean p_i^{-1})
- Pools behave as a single agent with mining power equal to the sum of participants
- The expected **reward** of i is the (expected) fraction of blocks that i mined out of the total number of blocks *in the longest chain*

LONGEST CHAIN IN THIS WORLD

- Whenever selected to build a block, point to the node "furthest from the root"
  - Break ties in favor of the one you hear first
- Broadcast to the whole network

Intuition [Nakamoto 08, the entire Bitcoin community]

- If all other miners follow the longest chain protocol
- And you have <50% of the mining power
- Your best response is to also follow the longest chain protocol

WHY?

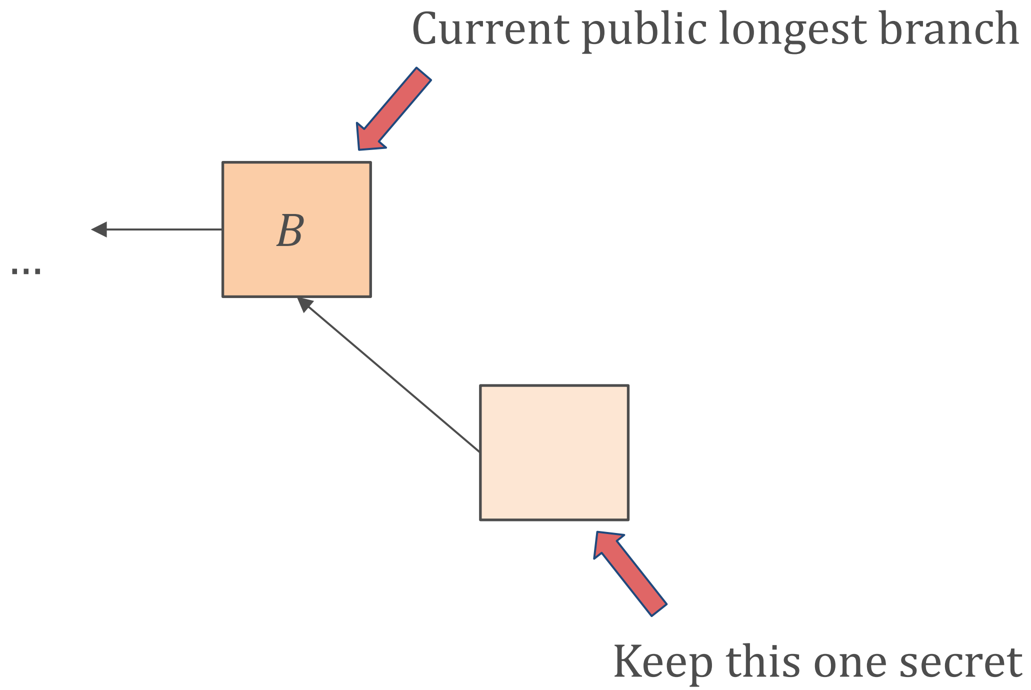
- Intuition:
- You only get rewards if your blocks are included in the longest chain
- The rest of the network has more power than you, so if you try to mine you own private chain you'll never catch up
- Nakamoto even has a correct random walk analysis
  - Doesn't consider more clever deviations

SELFISH MINE: IDEA

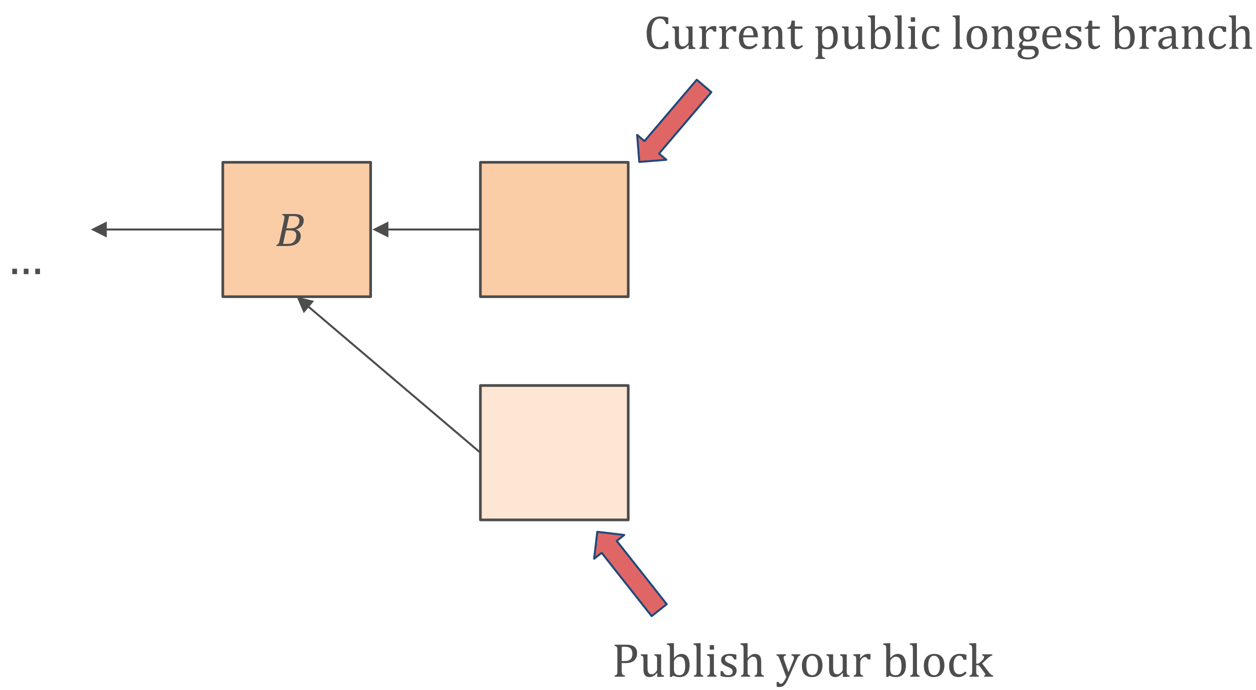
- Everyone mines on top of block B
- Hide a valid block B_s
- Everyone else is wasting resources trying to extend B, while you extend B_s without any competition

Theorem [Eyal-Sirer 14]

If you have >33% of the mining power, following the longest chain protocol is **not** a best response to all others following the longest chain protocol

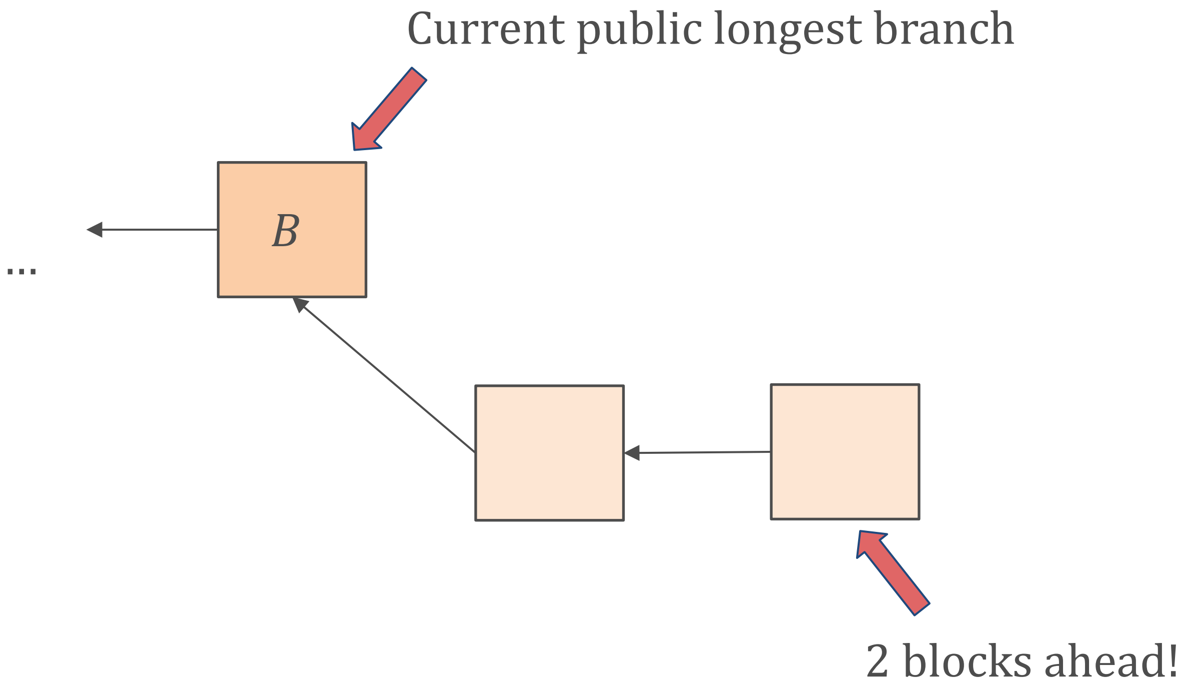

Current public longest branch
...
B
Keep this one secret

SCENARIO 1: THE OTHERS CATCH UP



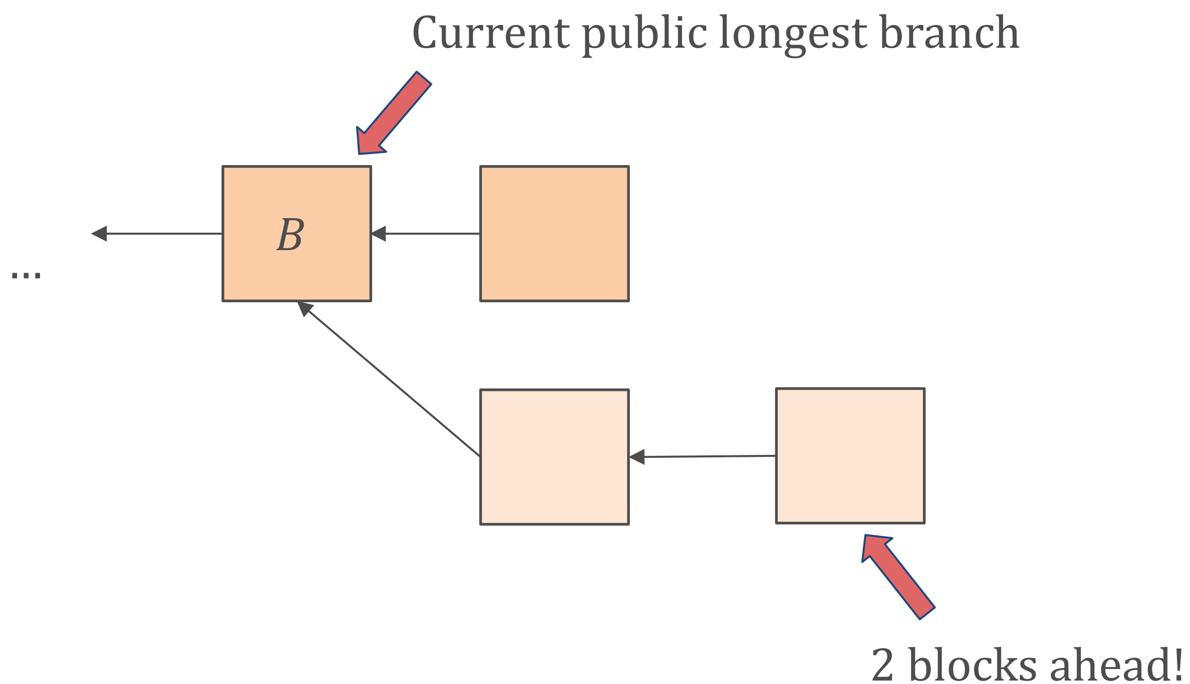
- Some honest miners will try extend your block because they heard about it first (natural network delays)
- Basically a toss-up

SCENARIO 2: YOU MINE A NEW ONE

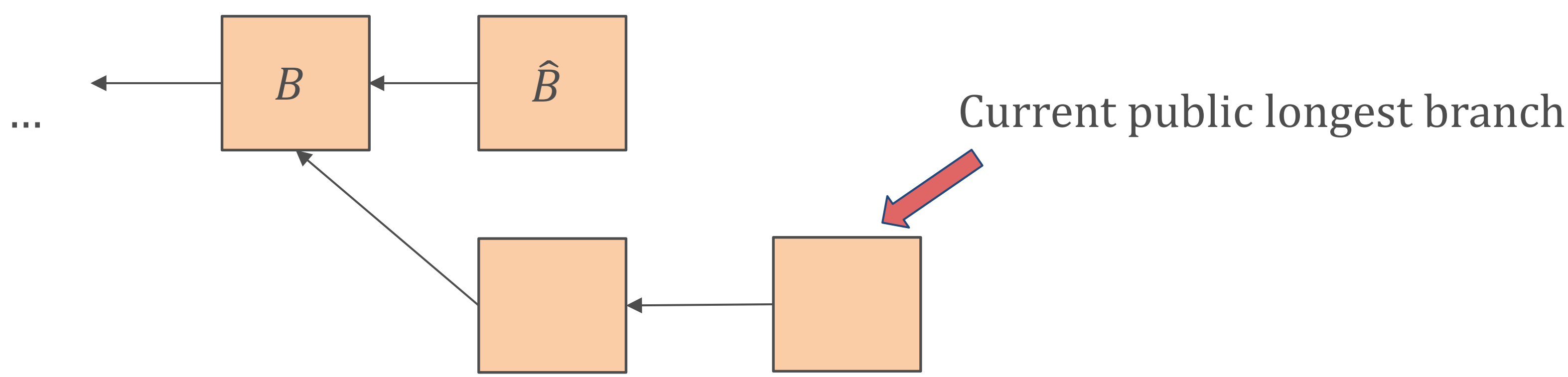


Try to make your private chain even longer!

SCENARIO 2: YOU MINE A NEW ONE



SCENARIO 2: YOU MINE A NEW ONE



- *Intuition:* The effort of honest miners for creating $\hat{B}$ is wasted!

TOY ANALYSIS

- LuckyLongestChain:
  - Whenever selected to build a block, point to the longest chain node, and break ties in favor of SelfishMiner.
  - Always broadcast your block.
- LuckySelfishMine
  - Whenever selected to build a block, point to the longest chain node, and break ties in favor of SelfishMiner.
  - Broadcast your block iff there is another node of the same distance from the root

TOY ANALYSIS

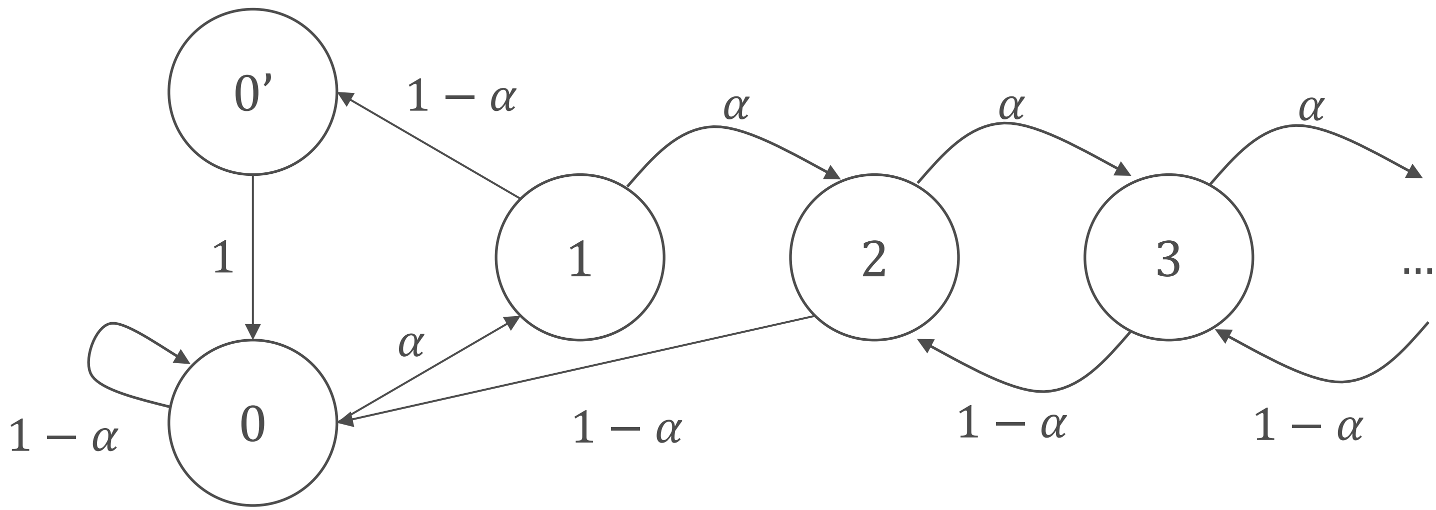
- LuckySelfishMine is strictly better than LuckyLongestChain, if everyone else is playing LuckyLongestChain.
  - With x fraction of the mining power it gives $x/(1-x)$ fraction of the blocks (instead of x)
- Intuition:
  - Every block is on the longest chain
  - Every block "negates" one other block by the honest people, effectively reducing the overall computational power that goes in actual block making
- We'll show morally the same result for real LongestChain

SELFISH MINE RECAP

- Maintain a private chain
- If $private\ chain = 0$, and others find block try to extend that
- If $private\ chain = 1$ and others find block, publish $private\ chain$ and try to extend it
- If $private\ chain = 2$ and others find block, publish $private\ chain$ and restart
- If $private\ chain > 2$ and others find block, publish first unpublished block of $private\ chain$

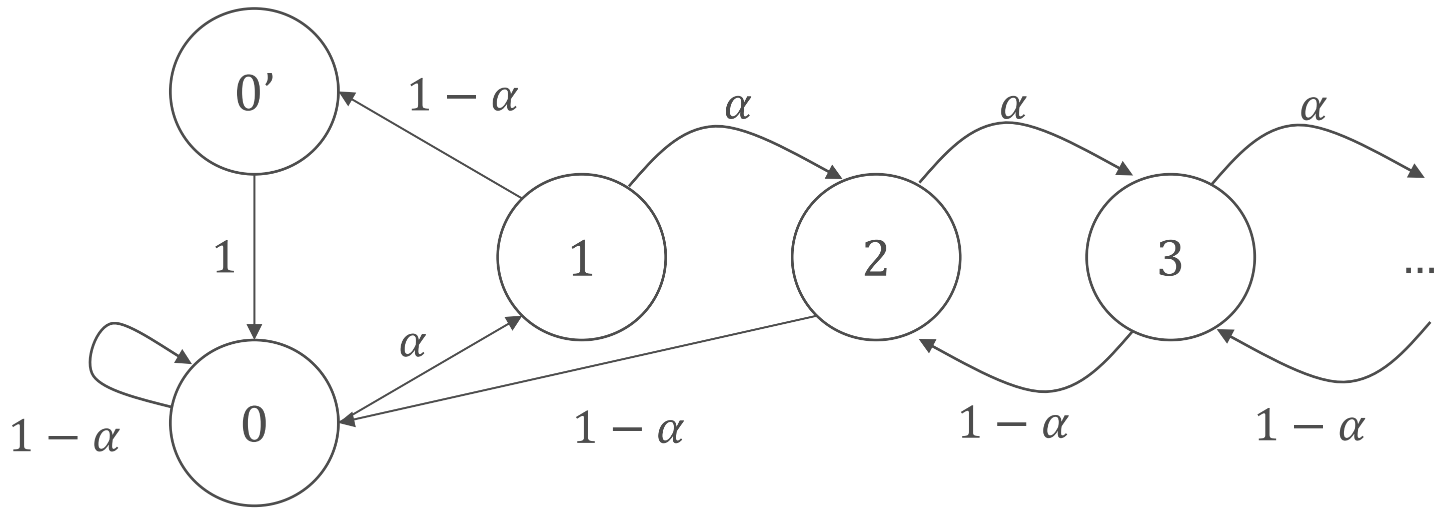
MODEL AS A 2 PLAYER GAME

- Attacker has α fraction of the computational power
- Honest miners have a $1 - \alpha$ fraction
- $\gamma =$ fraction of honest miners who break tie in favor of the attacker when there are two branches of equal length
- Goal: show that the selfish mining attack leads to the attacker having more than an α fraction of the blocks in the final chain



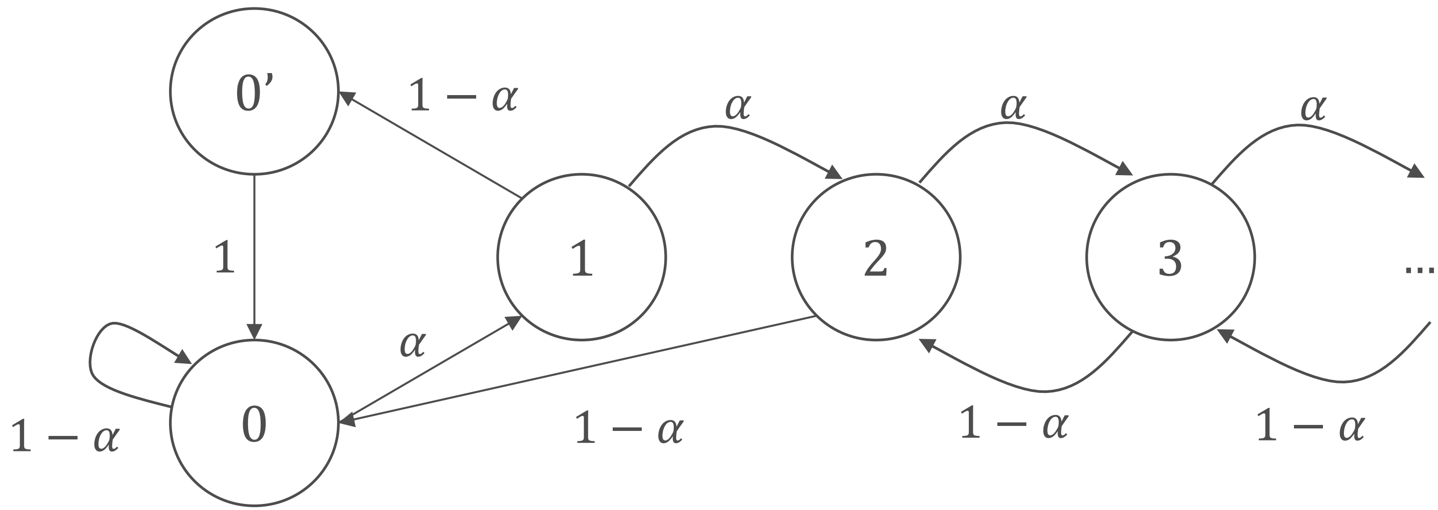
- State 0: no branches
- State 0’: two public branches of length 1
- State i: private chain is i blocks long
- From 0’ to 0:
  - Attacker makes a public block with frequency α
  - Honest miners that follow attacker make a public block with frequency $(1-\alpha)\gamma$
  - Honest miners not following attacker make a public block with frequency $(1-\alpha)(1-\gamma)$

ANALYSIS



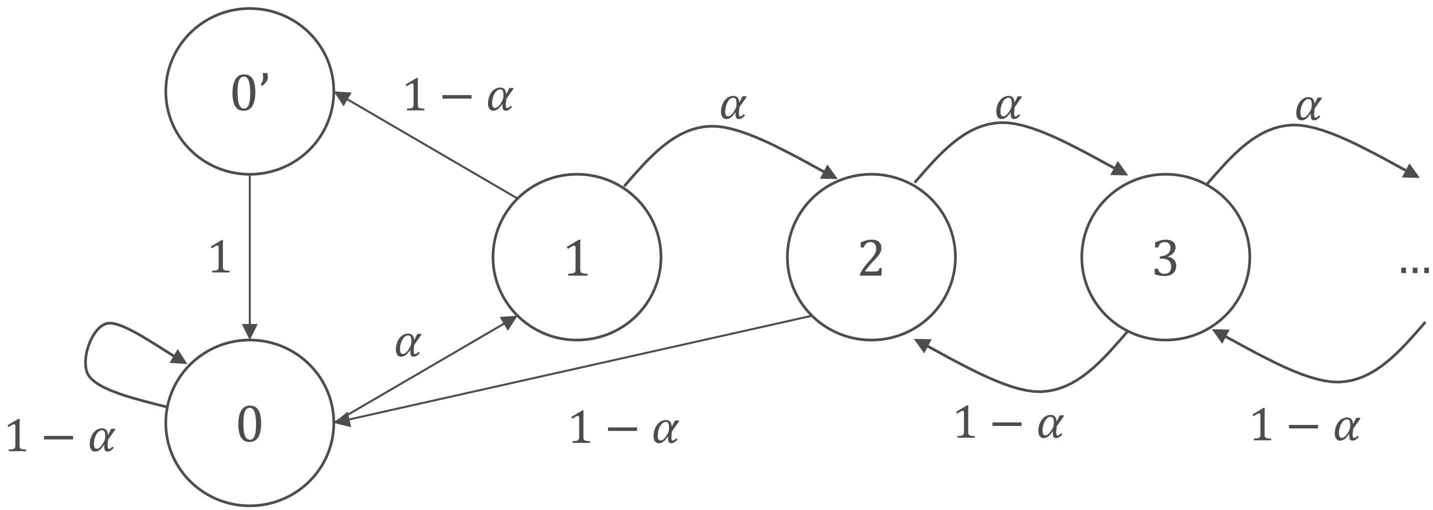
- $p_0 = (1-\alpha)p_1 + (1-\alpha)p_2 + (1-\alpha)p_0$
- $p_{0'} = (1-\alpha)p_1$
- $\alpha p_1 = (1-\alpha)p_2$
- $\forall k \geq 2\colon \alpha p_k = (1-\alpha)p_{k+1}$
- $\sum_{k=0}^{\infty} p_k + p_{0'} = 1$

ANALYSIS



- $p_0 = \frac{\alpha - 2\alpha^2}{\alpha(2\alpha^3 - 4\alpha^2 + 1)}$
- $p_{0'} = \frac{(1-\alpha)(\alpha - 2\alpha^2)}{1 - 4\alpha^2 + 2\alpha^3}$
- $p_1 = \frac{\alpha - 2\alpha^2}{2\alpha^3 - 4\alpha^2 + 1}$
- $\forall k \geq 2, p_k = \left(\frac{\alpha}{1-\alpha}\right)^{k-1} \frac{\alpha - 2\alpha^2}{2\alpha^3 - 4\alpha^2 + 1}$

REVENUE

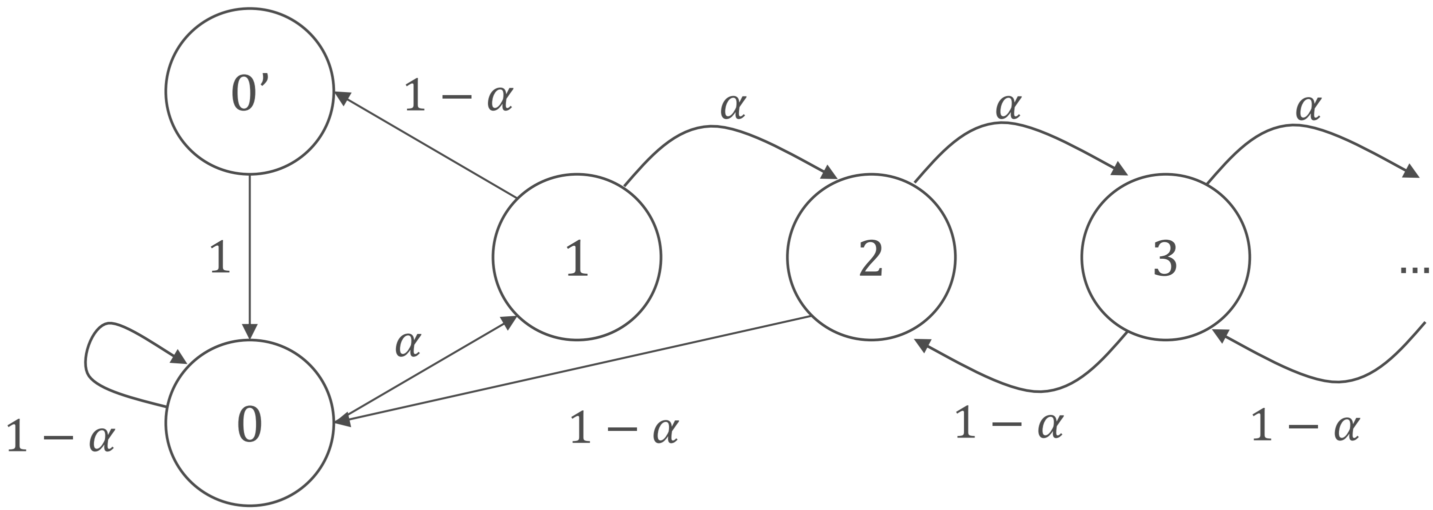


a) Two branches of length 1, attacker finds a block
   - Attacker makes revenue of 2
   - $r_{att} \mathrel{+}= 2 \cdot p_{0'} \cdot \alpha$

b) Two branches of length 1, honest miners find a block on top of attacker’s block
   - Attacker and honest make 1 each
   - $r_{att} \mathrel{+}= p_{0'} \cdot \gamma \cdot (1-\alpha), r_{hon} \mathrel{+}= p_{0'} \cdot \gamma \cdot (1-\alpha)$

c) Two branches of length 1, honest miners find a block on top of honest block
   - Honest make revenue of 2
   - $r_{hon} \mathrel{+}= p_{0'} \cdot (1-\gamma) \cdot (1-\alpha)$

REVENUE



d) No private branch, honest find block
- Honest make revenue of 1
- $r_{hon} \mathrel{+}= p_0 \cdot (1-\alpha)$

e) Lead is 2. Honest find block; attacker publishes private chain
- Attacker makes revenue of 2
- $r_{att} \mathrel{+}= p_2 \cdot (1-\alpha) \cdot 2$

f) Lead more than 2. Honest find block; attacker publishes one block
- Attacker makes revenue of 1
- $r_{att} \mathrel{+}= \Pr[lead > 2] \cdot (1-\alpha)$

REVENUE

- Protocol adjusts difficulty so that there is a block every ~10 mins
- So, total revenue for attacker is

$$\frac{r_{att}}{r_{att}+r_{hon}}=\frac{\alpha(1-\alpha)^2\big(4\alpha+\gamma(1-2\alpha)\big)-\alpha^3}{1-\alpha(1+(2-\alpha)\alpha)}$$

<u>Observation:</u> Selfish mining is profitable when

$$\frac{1-\gamma}{3-2\gamma}<\alpha<\frac{1}{2}$$

REVENUE

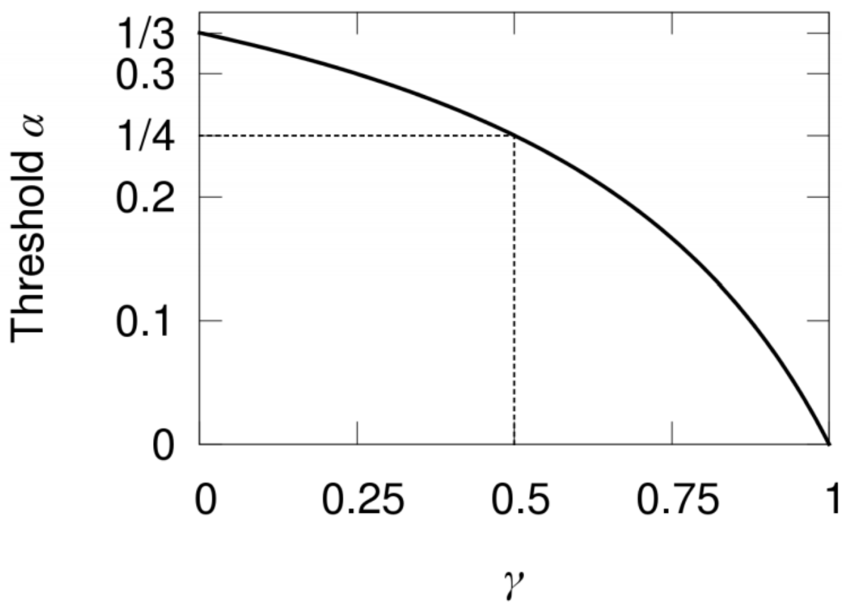


Fig. 3: For a given γ, the threshold α shows the minimum power selfish mining pool that will trump the honest protocol. The current Bitcoin protocol allows $\gamma = 1$, where Selfish-Mine is always superior. Even under unrealistically favorable assumptions, the threshold is never below 1/3.

KIAYIAS, KOUTSOUPIAS, KYROPOULOU,TSELEKOUNIS 16

- Study strategic considerations regarding block withholding
- When is honest/longest chain behavior a Nash equilibrium?

SETUP [KKKT 16]

- n players/miners
- p_i= Probability that miner solves puzzle
  - $\sum_i p_i = 1$
- d = Depth of the game
  - Payoffs count only after d blocks
  - Mostly $d = \infty$
- r^*= reward of mining a block
  - Normalized to 1

SETUP

- Public state:
  - A rooted tree of blocks
  - Every node is labeled by one of the players (the miner)
  - Every level has at most one block labeled by player i (no reason for i to mine two)
- Private state of player i:
  - Same as public state, but might have some extra blocks labeled by i
  - Public state is a subtree

TWO MODELS

1. **Immediate release model (today)**
   - Whenever a miner succeeds in mining a block, he releases it immediately, and all miners can continue from the newly mined block.
2. **Strategic release model**
   - Whenever a miner succeeds in mining a block, it becomes common knowledge. The miner can decide to postpone its release; others cannot extend it until its public, but know it exists
   - Of course, not meant to be realistic, but a stepping stone to the incomplete information game

STRATEGIES

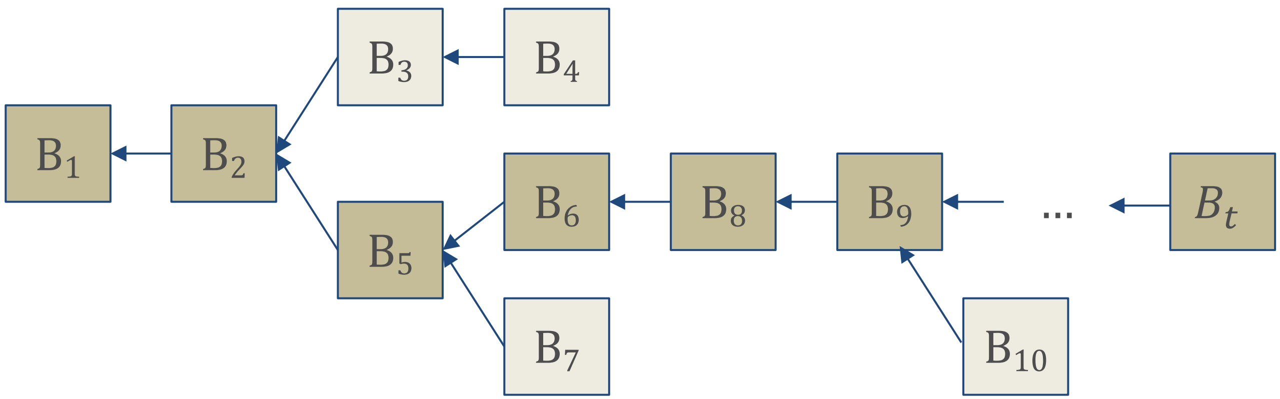
- Strategy: Two functions (μ_i, ρ_i)
  - Mining function μ_i selects a block from the public state to mine
  - Release function ρ_i which is a (perhaps empty) private part of the player's state which is added to the public state.
- FRONTIER/honest strategy: release any mined block immediately and select to mine one of the deepest blocks

PHASES

- Game is played in phases
- In phase t player i is selected with probability p_i to extend the block indicated by μ_i
- Then everyone adds information to the public tree according to their release functions
- Repeat

PAYMENTS

- A miner makes revenue of 1 for *every* node in the *first* path to make it to depth d

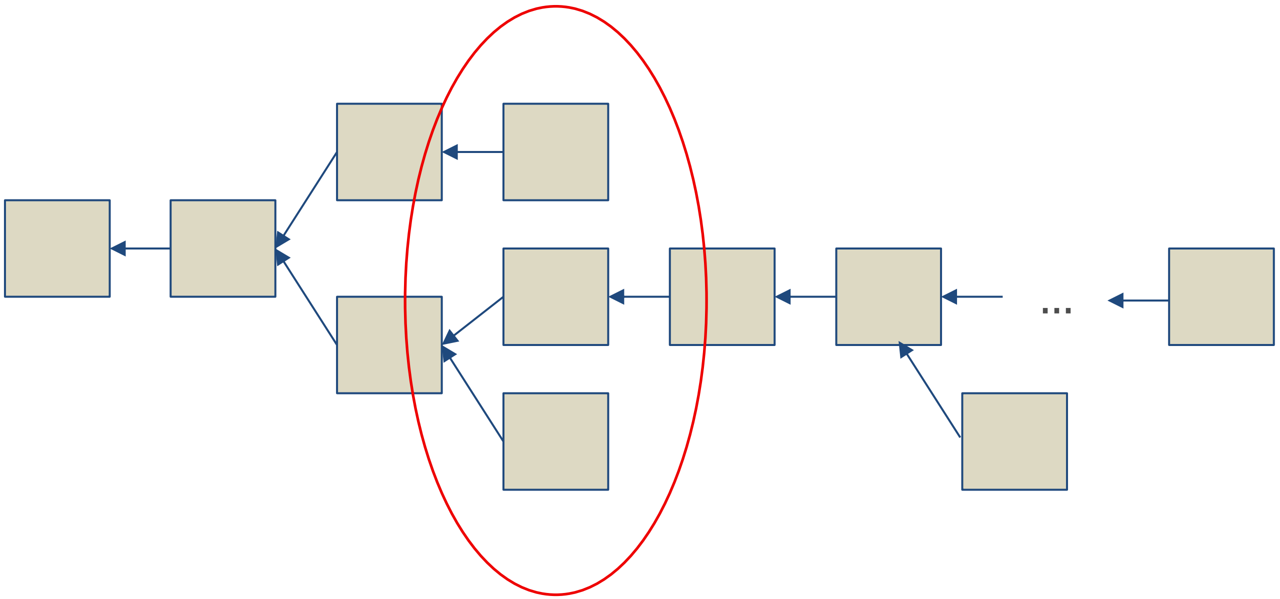


- Once B_5 is paid, no one tries to extend B_3 or B_4

IMMEDIATE RELEASE GAME

- Want to see when FRONTIER is a best response to everyone else playing FRONTIER
- Problem reduces to a two player game
- Miner 2 with computational power $1 - p$ plays honestly/FRONTIER
- Miner 1 with computational power p best responds to miner 1
- Public state is a tree of width at most 2: two long branches with lengths (a, b)
  - $a =$ length of branch where miner 1 mines
  - $b =$ length of branch where miner 2 mines

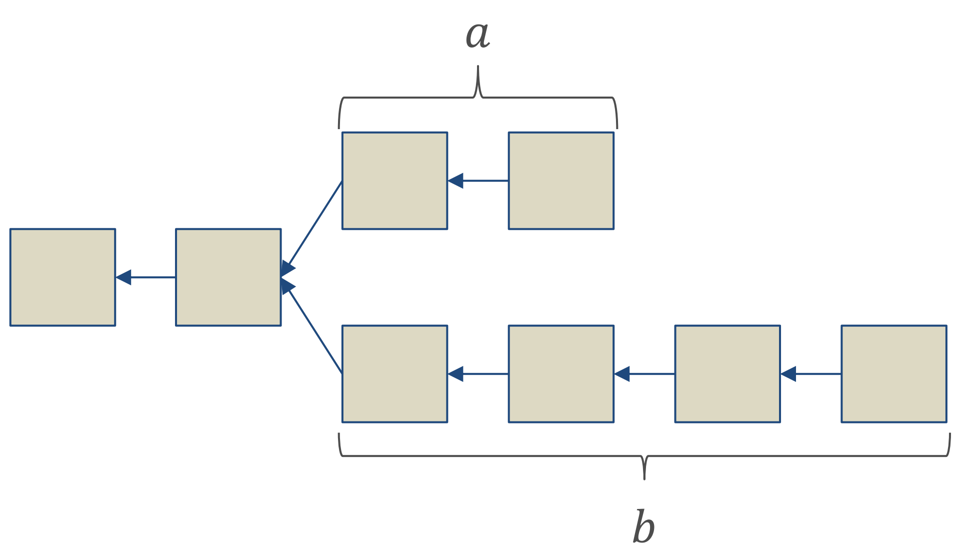
IMMEDIATE RELEASE GAME



This never happens

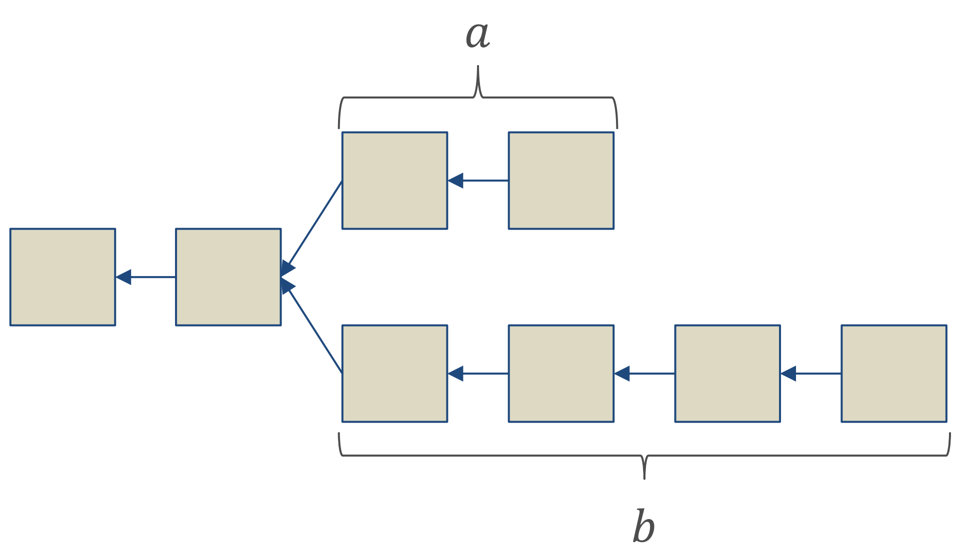
IMMEDIATE RELEASE GAME

- State could be (0,0)
- If $b > 0$, then since Miner 2 is extending the longest chain, $b > a$
  - Eg (3,1) never happens



IMMEDIATE RELEASE GAME

- **Mining states (M):** both mine their own chain
- **Capitulation states (C):** miner 1 gives up
- **Winning states (W):** miner 2 switches ($a > b$)



IMMEDIATE RELEASE GAME

- $g_k(a, b)$: expected gain of miner 1 when the branch of the honest miner in the execution tree is extended by k levels, when starting from an (a, b) tree
  - Intuitively should not depend on (a, b)
- $g^* =$ expected gain per level
- $g^* = \frac{g_k(a,b)-g_{k'}(a,b)}{k-k'}$, for large k, k' and all a, b
- $g_k(a, b) = k \cdot g^* + \phi(a, b)$
  - $\phi(a, b) = \lim_{k\to\infty} g_k(a, b) - k \cdot g^* =$ advantage of miner 1 for being in state (a, b)
  - Alternatively, $\phi(a, b)$ is the expected value of $g_k(a, b) - k \cdot g^*$ until $(0,0)$ is reached
- Objective of miner 1: **maximize $\boldsymbol{g^*}$**

IMMEDIATE RELEASE GAME

- For $(a, b) \in M$: with probability p we go to $(a + 1, b)$, otherwise to $(a, b + 1)$
- For $(a, b) \in$ C: miner 1 abandons branch. New state $(0, s)$
  - Not necessarily (0,0)
- For $(a, b) \in W$: miner 2 abandons branch. New state (0,0)

- Strategy $=$ pair (M, s) where $(0, s)$ is the state miner 1 jumps to when giving up

IMMEDIATE RELEASE GAME

- Define $g_k(a,b)$ recursively

$$g_k(a,b) = \begin{cases} g_{k-1}(0,0) + a, if\ a = b+1 \\ \max\{\underbrace{\max_{s=0,\dots,b-1}\{g_k(0,s)\}}_{\text{Give up}}, \underbrace{p g_k(a+1,b) + (1-p) g_{k-1}(a,b+1)}_{\text{Don't give up}}\} \end{cases}$$

- Similar for ϕ

$$\phi(a,b) = \begin{cases} \phi(0,0) + a - g^*, if\ a = b+1 \\ \max\{\max_s \phi(0,s), p\phi(a+1,b) + (1-p)\phi(a,b+1) - (1-p)g^*\} \end{cases}$$

- $\phi(0,0) = 0$

IMMEDIATE RELEASE GAME

Theorem: FRONTIER is not a best response for $p \geq 0.455$

Proof:

- Say $d = 3$
- $M = \{(0,0), (0,1), (1,1), (1,2), (2,2)\}, s = 1$
  - Capitulate in $(a, b), b \geq 3$, and jump to $(0,1)$
- Need to confirm that $g^* \geq p$
  1. Compute $\phi(a, b)$
     - $\phi(0,0) = 0, \phi(0,1) = (g^* - p)/(1 - p), \phi(2,2) = \cdots$
  2. It must be that $\phi(a, b) \geq \phi(0,1)$ for $(a, b) \in \mathrm{M}$
  3. Picking $g^* = \frac{p^2(2+2p-5p^2+2p^3)}{1-p^2+2p^3-p^4}$ makes everything hold for all $p \geq 0.455$

IMMEDIATE RELEASE GAME

Theorem: FRONTIER is a NE if and only if $p \leq h_0$, where $h_0 \in [0.361, 0.455]$

Corollary: Frontier is a NE if $p \leq 0.361$

IMMEDIATE RELEASE GAME

Proof sketch:

Starting at any state (a, b), one of the two miners will give up.

1. Bound the probability that miner 1 wins this race starting from state (a, b)
   - $r(a, b) \leq \left(\frac{p}{1-p}\right)^{1+b-a}$
2. Bound the difference of ϕ between different states as a function of the probability of winning
- $\phi(a, b)$ is non-decreasing in a
3. Using all of the above, get an upper bound on $\phi(0,1)$ as a function of p
   - $\phi(0,1)$ can't be positive, so solve for p

STRATEGIC RELEASE GAME

Theorem: FRONTIER is a NE when a miner i has relative computational power $p_i \leq 0.308$

Major open direction:

- Do these results extend to incomplete information games?

NEXT TIME

- Transaction fees
- Incentives in mining pools
- Beyond Proof of Work