Blockchains & Cryptocurrencies

Mining and Alternative Puzzles

Instructor: Abhishek Jain
Johns Hopkins University - Spring 2021

*Some slides based on NBFMG

Today

- Mining Strategies
- Alternative puzzles

Along the way, keep identifying directions for improvements (or, motivation for altcoins)

Mining strategies

Game-theoretic analysis of mining

Several strategic decisions

- Which transactions to include in a block
  - Default: any above minimum transaction fee
- Which block to mine on top of
  - Default: longest valid chain
- How to choose between colliding blocks
  - Default: first block heard
- When to announce new blocks
  - Default: immediately after finding them

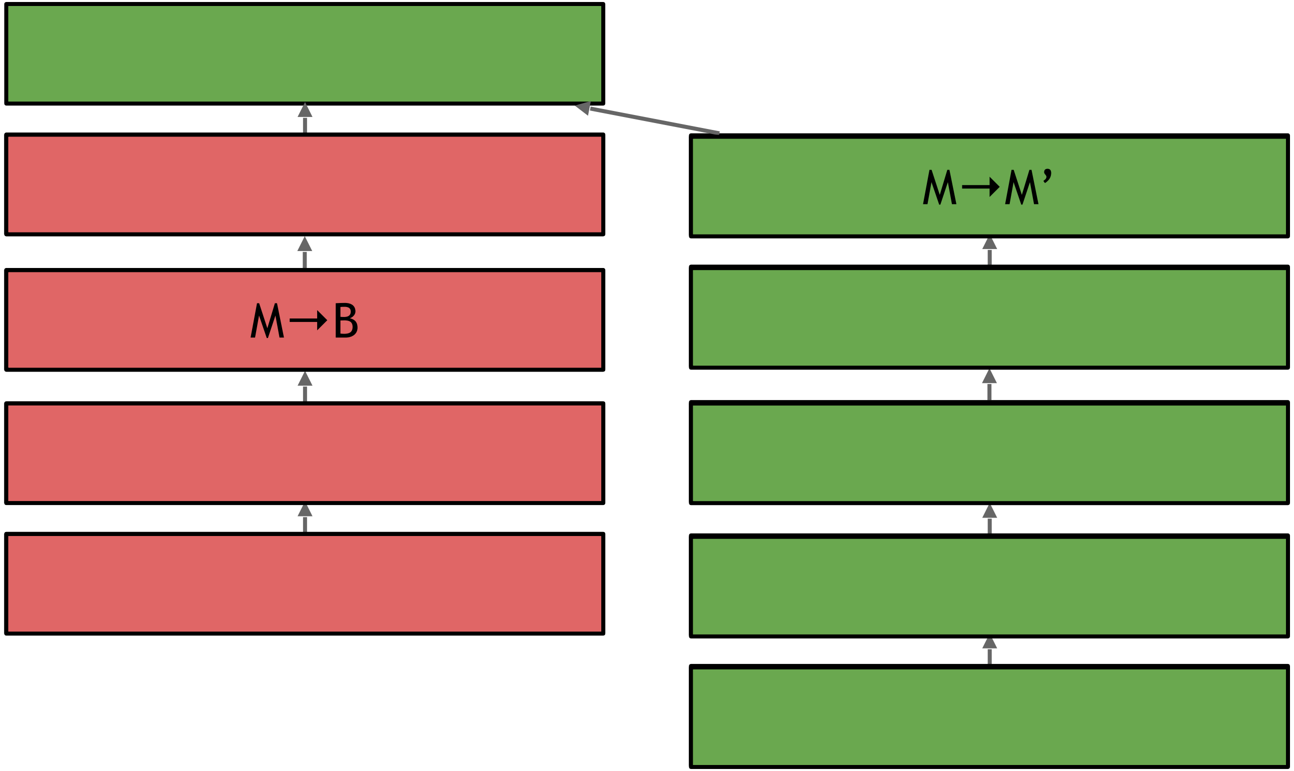
Game-theoretic analysis of mining

Assume you control $0 < \alpha < 1$ of mining power

Can you profit from a non-default strategy?

For some α, **YES!**

Forking attacks



Forking attacks

- Certainly possible if $\alpha > 0.5$
  - may be possible with less
- Attack is detectable
- Might be reversed
- Might crash exchange rate

PoW 51% Attack Cost

https://www.crypto51.app/

This is a collection of coins and the theoretical cost of a 51% attack on each network.

Learn More ⚡ Tip

Name	Symbol	Market Cap	Algorithm	Hash Rate	1h Attack Cost	NiceHash-able
Bitcoin	BTC	$212.07 B	SHA-256	156,092 PH/s	*$641,748*	0%
Ethereum	ETH	$43.26 B	Ethash	240 TH/s	*$272,454*	3%
BitcoinCashABC	BCH	$4.70 B	SHA-256	2,947 PH/s	*$12,117*	16%
Litecoin	LTC	$3.29 B	Scrypt	272 TH/s	*$20,709*	5%
Zcash	ZEC	$707.21 M	Equihash	7 GH/s	*$13,521*	3%
Dash	DASH	$686.48 M	X11	6 PH/s	*$2,070*	3%
EthereumClassic	ETC	$625.51 M	Ethash	4 TH/s	$4,075	231%
BitcoinGold	BTG	$138.61 M	Zhash	768 KH/s	*$287*	71%

What can a “51% attacker” do?

Steal coins from existing address? ✗

Suppress some transactions?

- From the block chain ✓
- From the P2P network ✗

Change the block reward? ✗

Destroy confidence in Bitcoin? ✓✓

Selfish Mining

(Block-Withholding Attack)

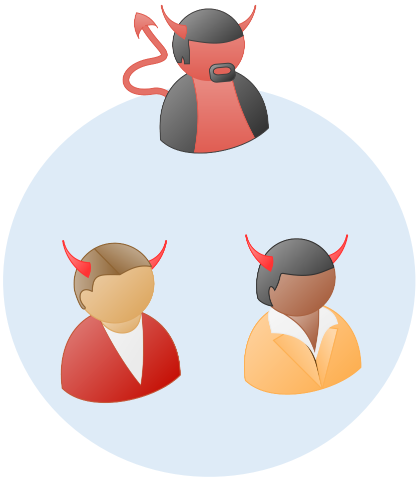
Majority is not Enough:
Bitcoin Mining is Vulnerable*

Ittay Eyal and Emin Gün Sirer

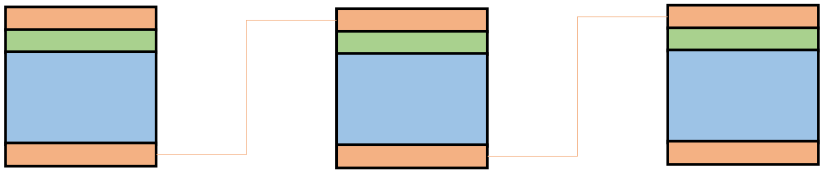
Department of Computer Science, Cornell University
ittay.eyal@cornell.edu, egs@systems.cs.cornell.edu

Selfish Mining Strategy (aka block withholding)

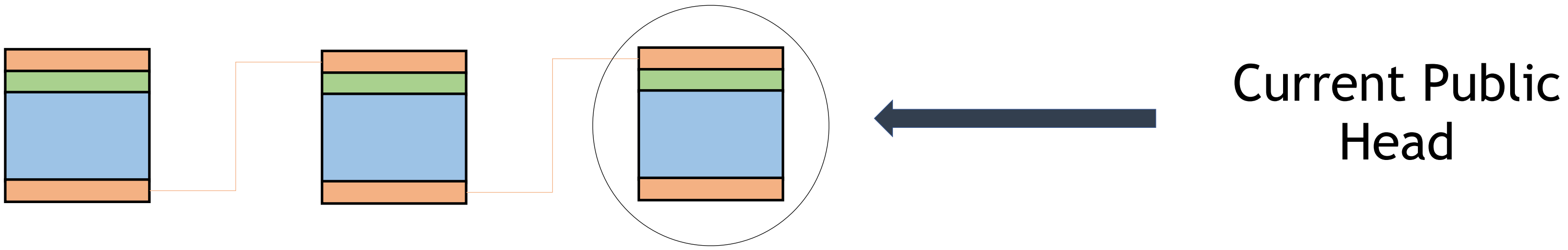
- Form a pool.
- Secretly mine blocks. Don't announce blocks right away. Try to get ahead!
- Announce as and when necessary to maintain lead, or to avoid falling behind



Public Chain

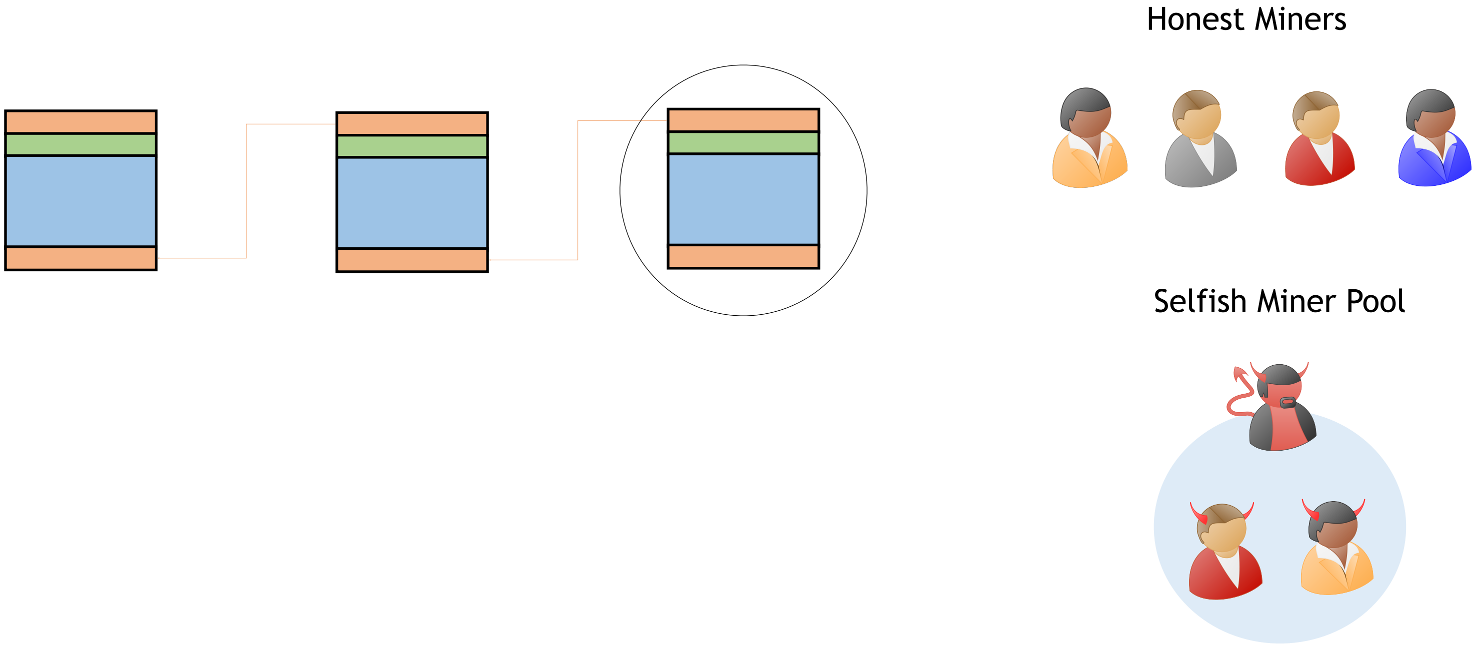


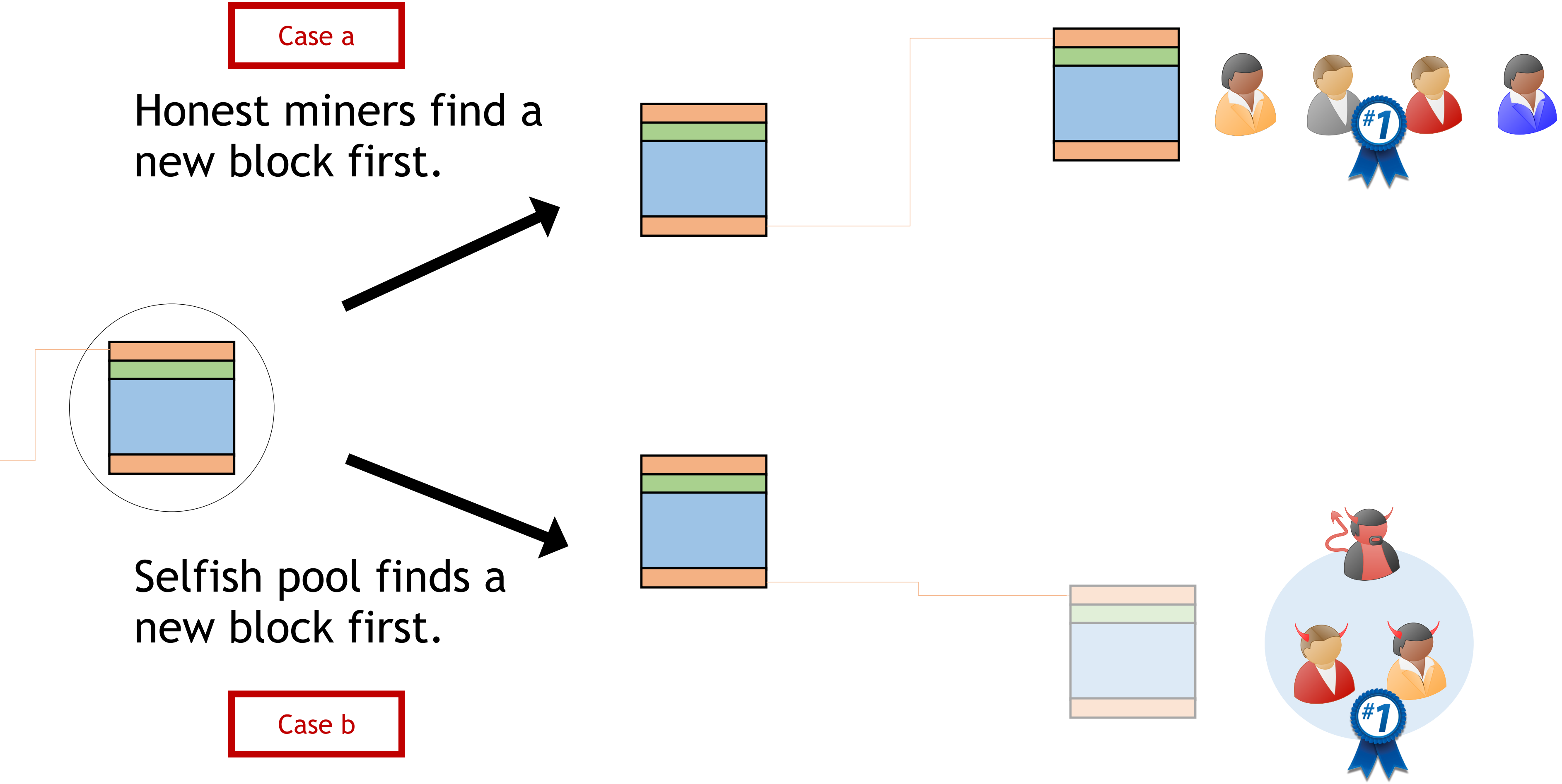
Public Chain



Public Chain

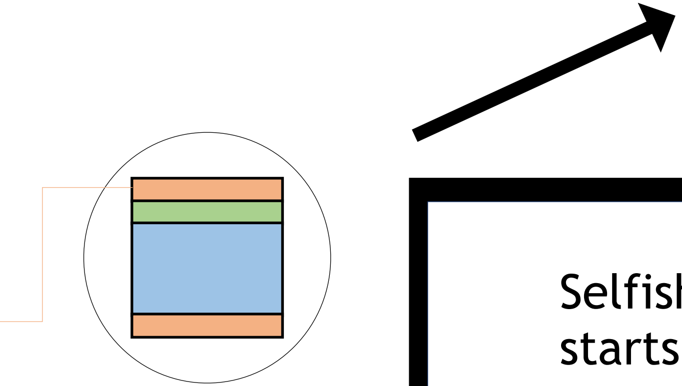
- The honest miners and the selfish miner pool start mining at the current public head.

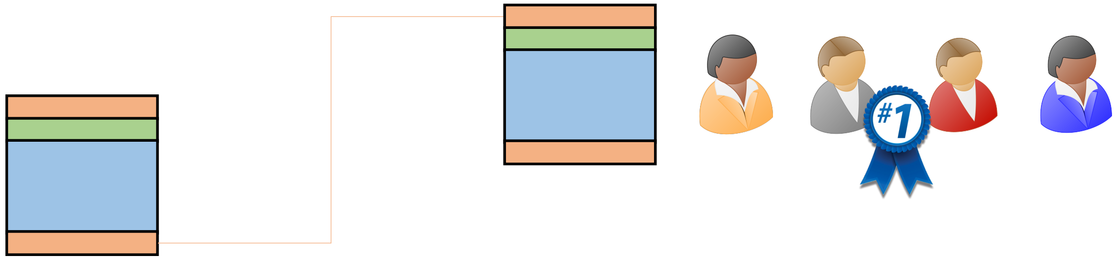



Case a
Honest miners find a new block first.
Selfish pool finds a new block first.
Case b
#1
#1

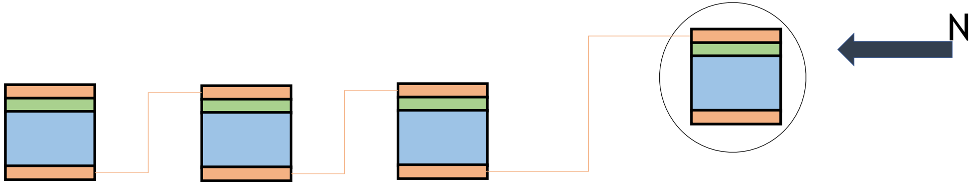
Case a

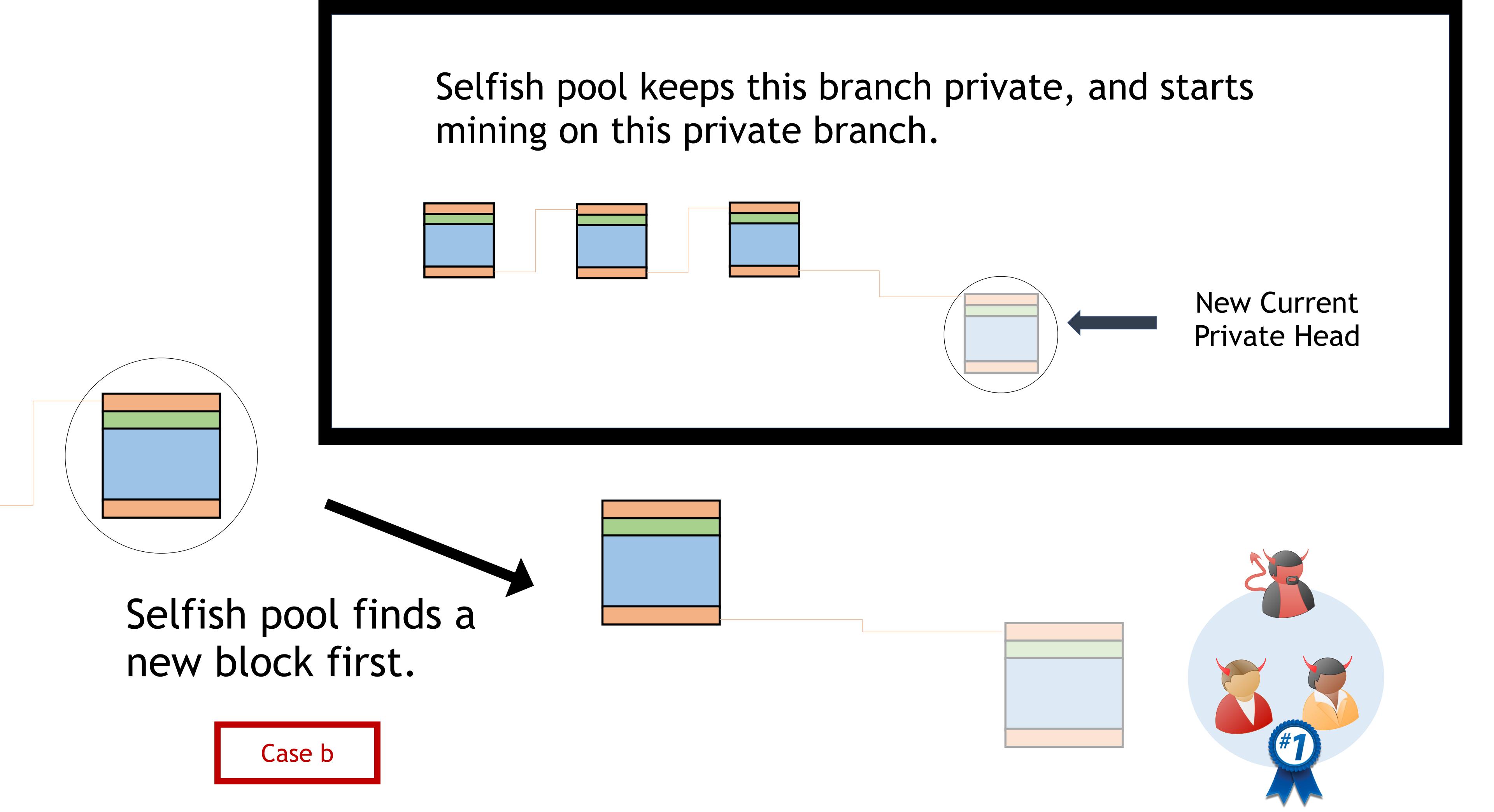
Honest miners find a new block first.



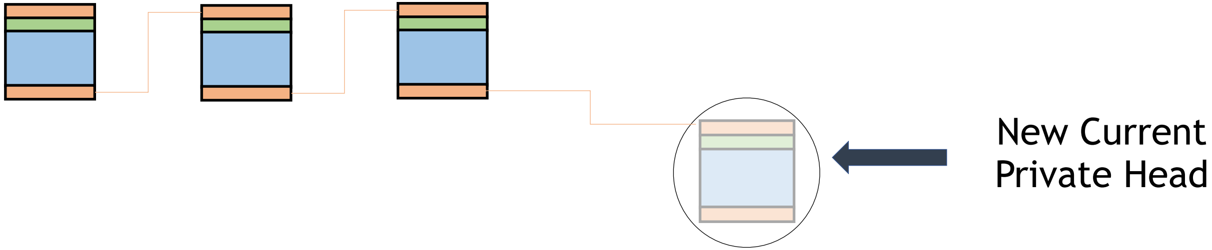


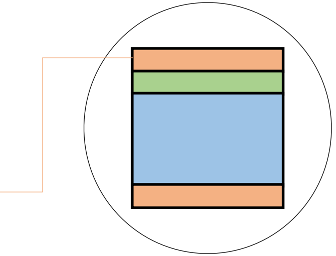
Selfish miner pool adopts the main branch and starts mining on the new current public head.

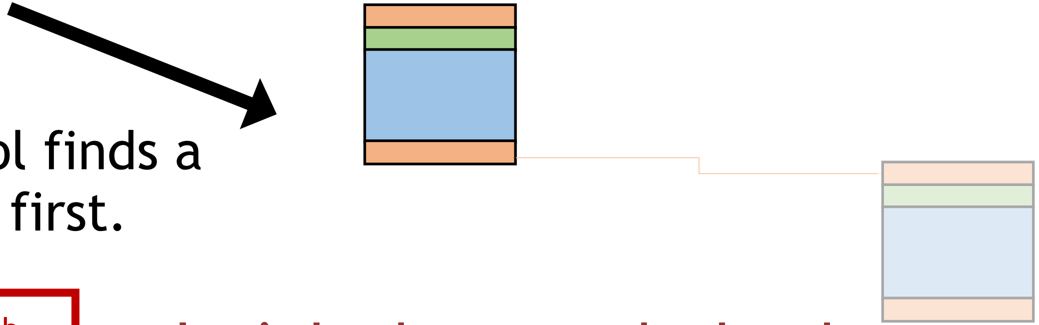



Selfish pool keeps this branch private, and starts mining on this private branch.
New Current Private Head
Selfish pool finds a new block first.
Case b
#1

Selfish pool keeps this branch private, and starts mining on this private branch.







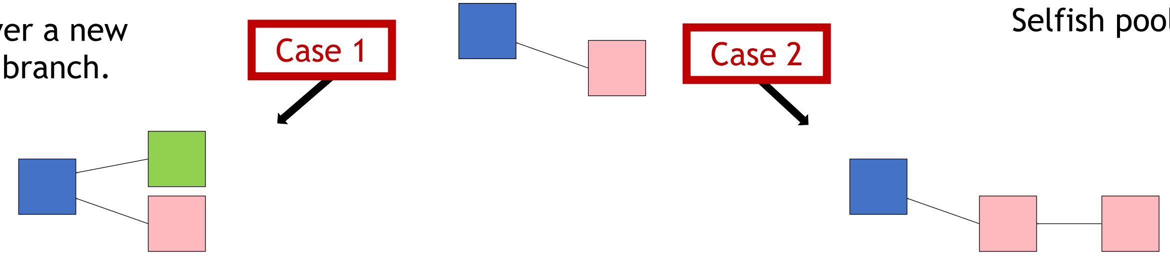
Selfish pool finds a new block first.

Case b

Let’s look at case b closely.

Honest miners discover a new block on the public branch.

Selfish pool finds a second block.

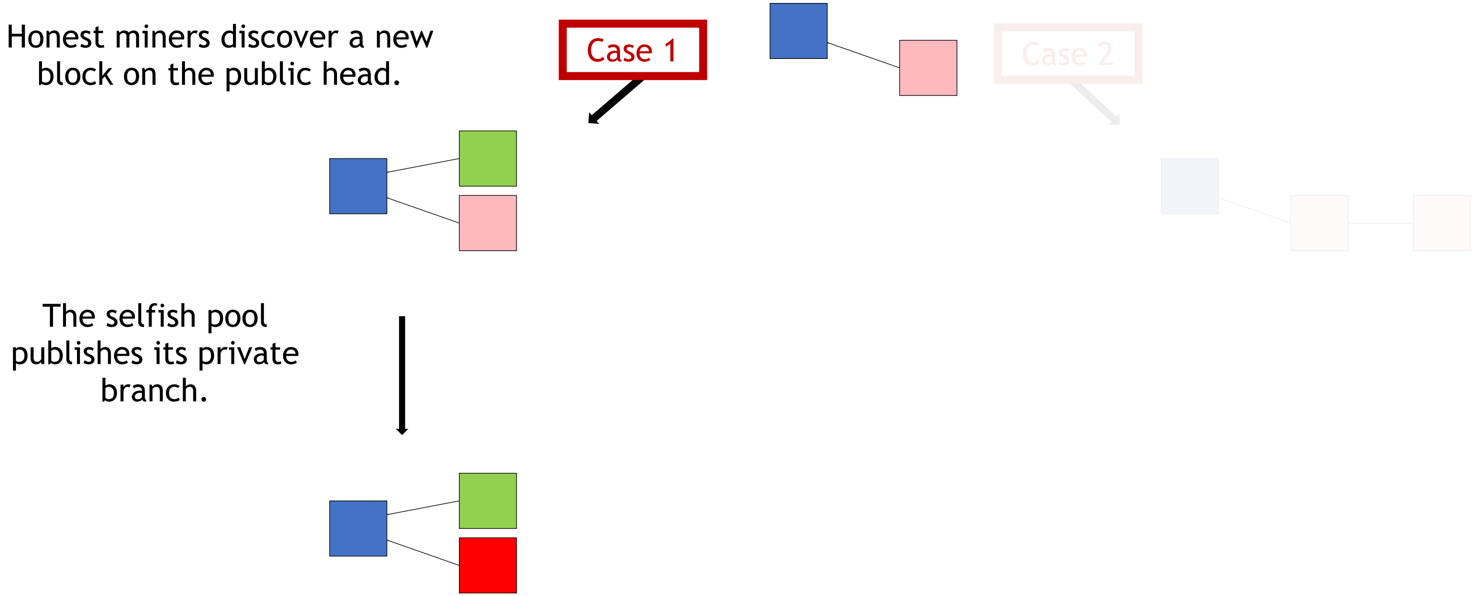


Public Head

Block mined by honest miners

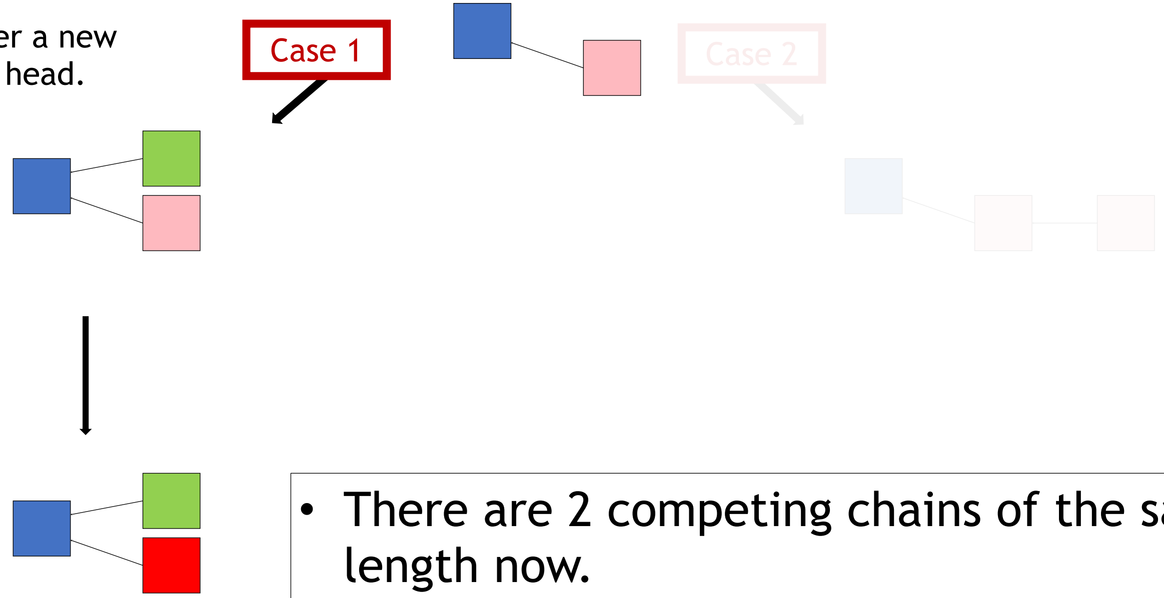
Block mined by selfish pool and kept private

Block mined by selfish pool and made public


Honest miners discover a new block on the public head.
Case 1
Case 2
The selfish pool publishes its private branch.

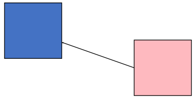
Honest miners discover a new block on the public head.

The selfish pool publishes its private branch.

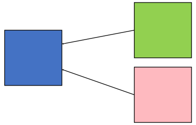


- There are 2 competing chains of the same length now.
- The selfish pool mines to extend its branch.
- Honest miners choose to mine on either branch.

Honest miners discover a new block on the public head.

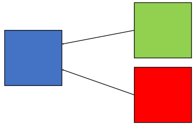


Case 1

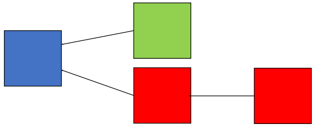


Case 2

The selfish pool publishes its private branch.

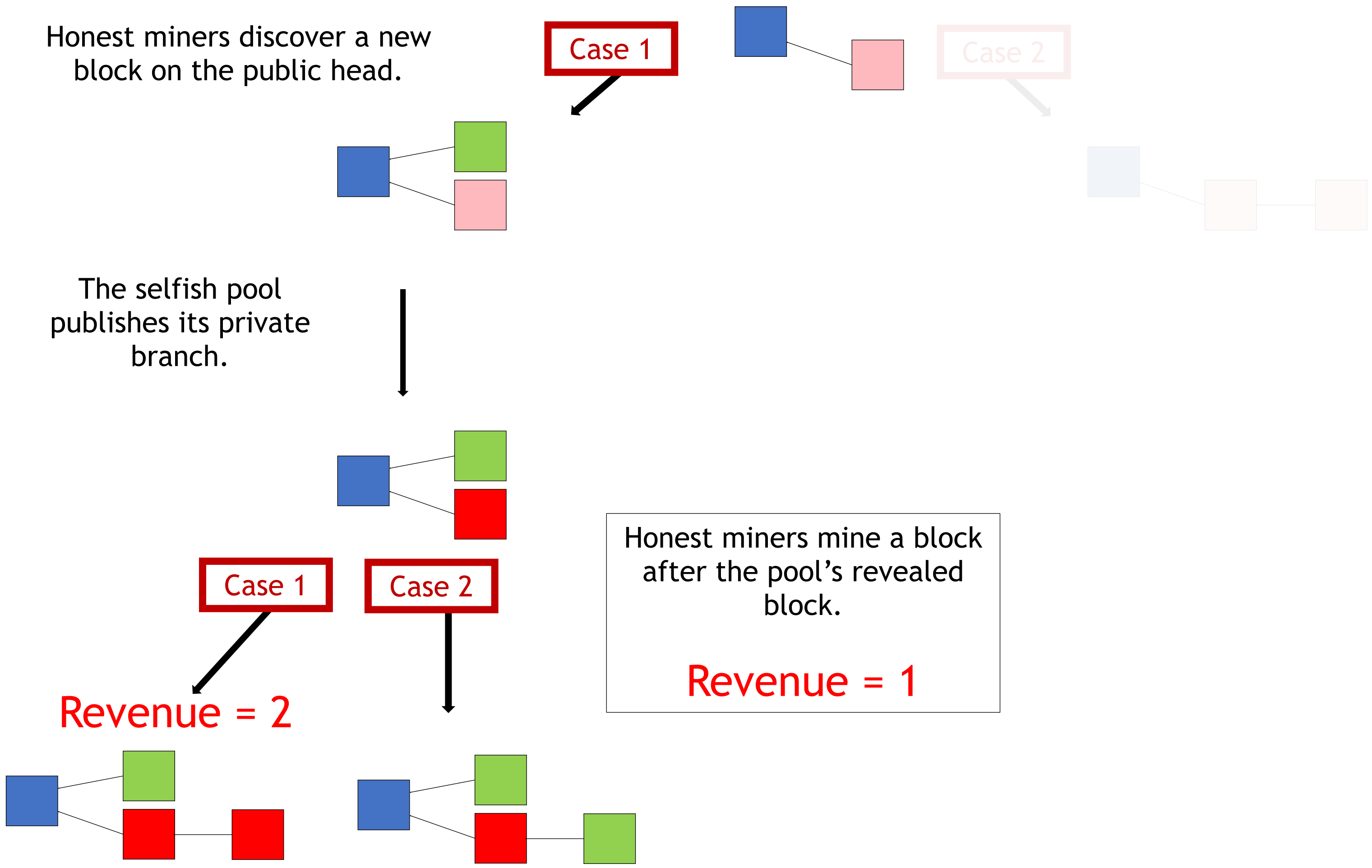


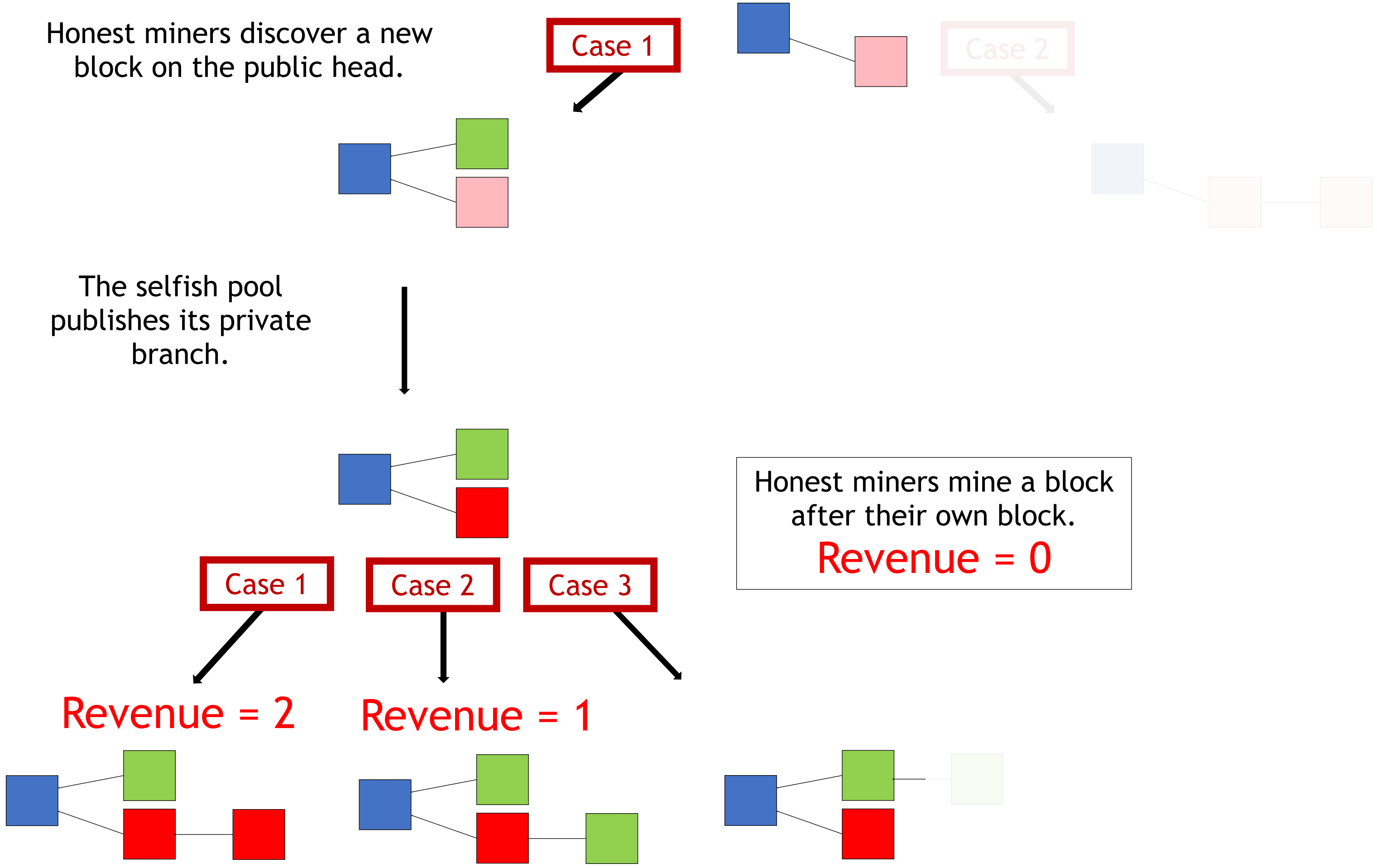
Case 1



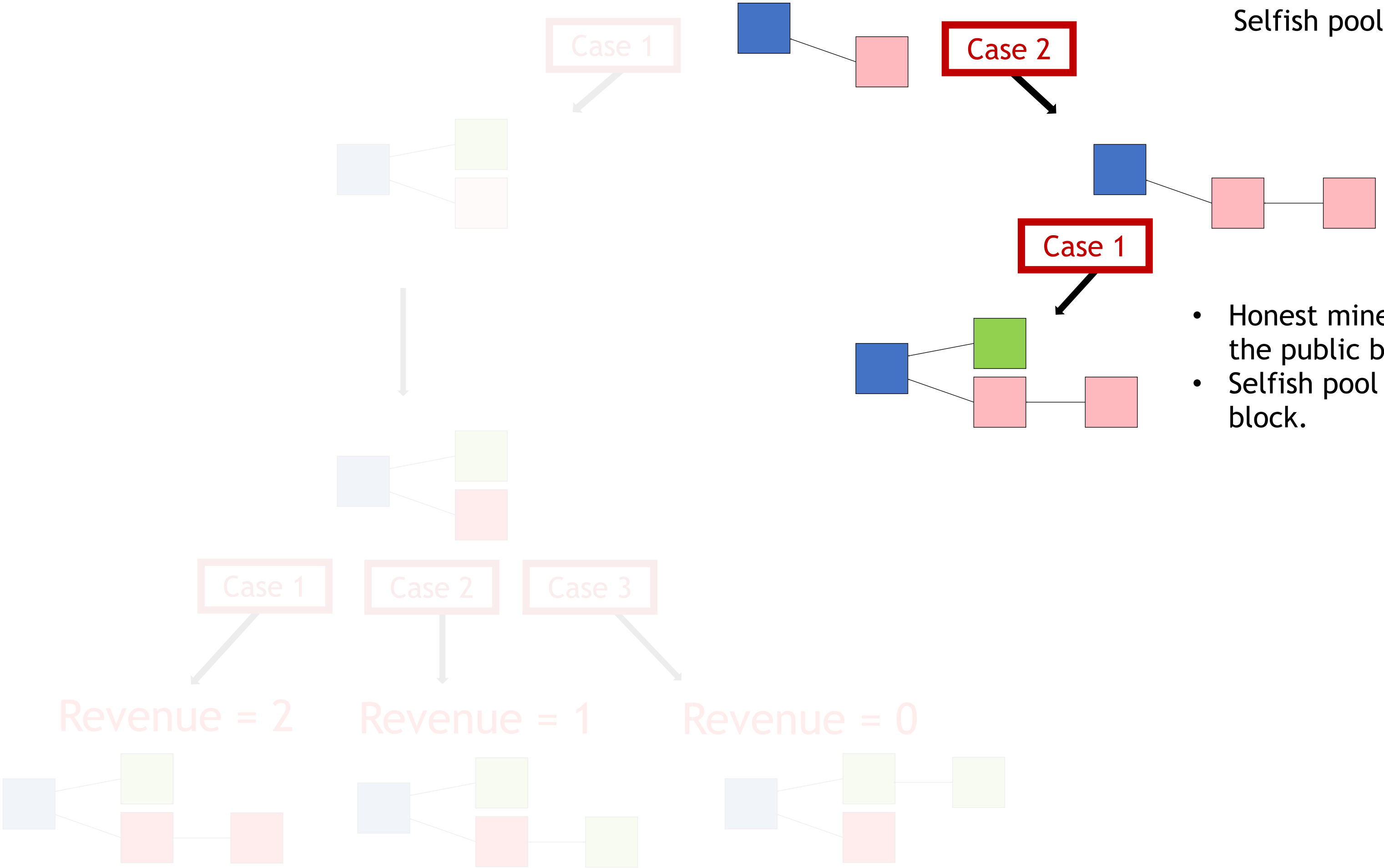
Selfish pool mines a second block and publishes it.

Revenue = 2


Honest miners discover a new block on the public head.
Case 1
Case 2
The selfish pool publishes its private branch.
Case 1
Case 2
Revenue = 2
Honest miners mine a block after the pool's revealed block.
Revenue = 1

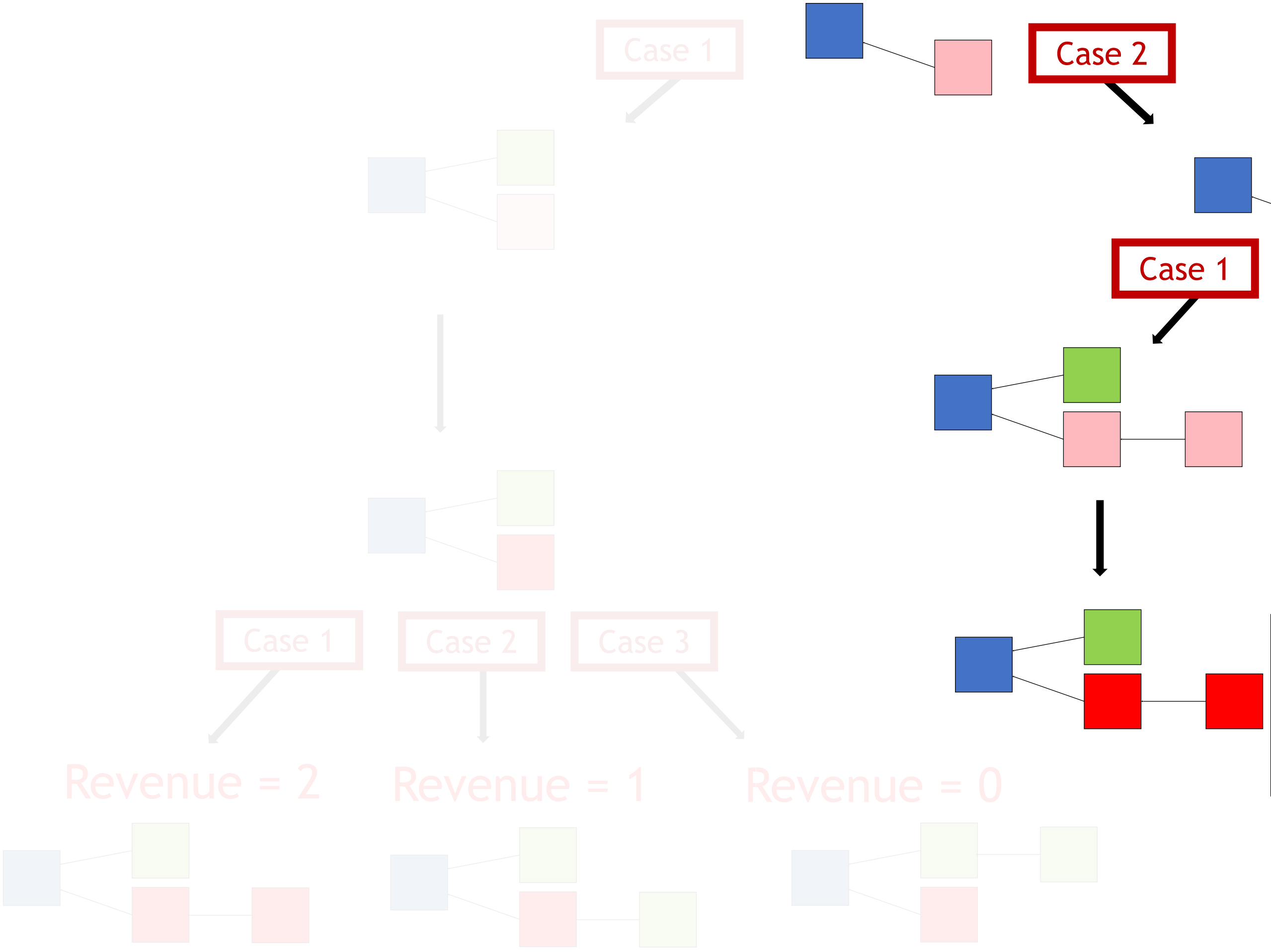

Honest miners discover a new block on the public head.
Case 1
Case 2
The selfish pool publishes its private branch.
Honest miners mine a block after their own block.
Revenue = 0
Case 1
Case 2
Case 3
Revenue = 2
Revenue = 1

Selfish pool finds a second block.



- Honest miners mine a block on the public branch.
- Selfish pool has a lead of 1 block.

Selfish pool finds a second block.

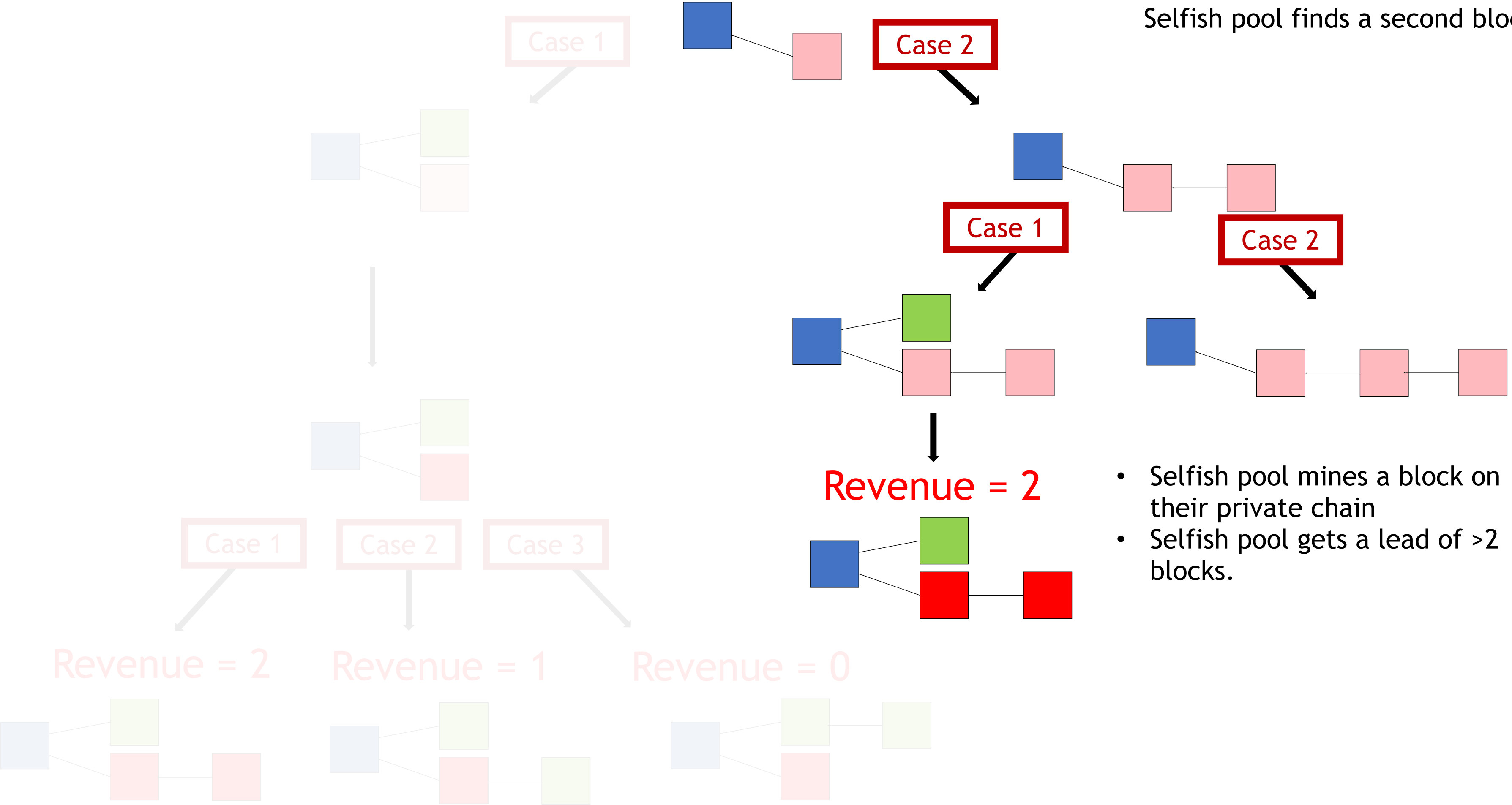


- Honest miners mine a block on the public branch.
- Selfish pool has a lead of 1 block.

Selfish Pool publishes the entire chain.

Revenue = 2

Selfish pool finds a second block.



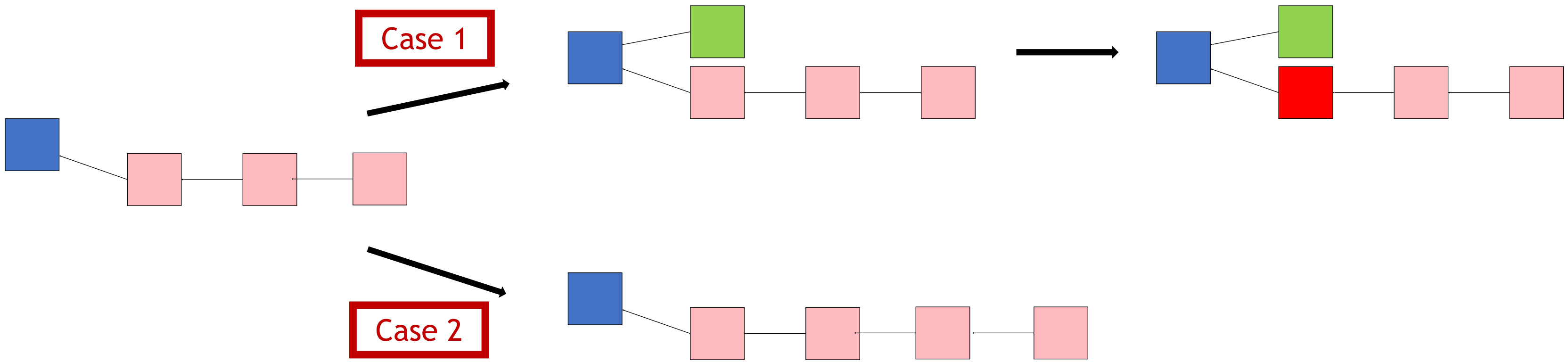
- Selfish pool mines a block on their private chain
- Selfish pool gets a lead of >2 blocks.

Selfish Pool gets a lead of >2 blocks

- Selfish pool continues to mine on its private branch.
- For each subsequent block mined by an honest party, it publishes one block from its private chain.
- Tries to maintain a lead of 2 blocks for as long as possible.
- If the lead reduces to 1, it publishes its private branch.

Earns revenue for all its blocks.

Selfish Pool gets a lead of >2 blocks



If the selfish pool is in minority, then with a very high probability this lead will eventually reduce to one block.

Analysis

- Set of miners in the system : $1, \ldots, n$
- Miner i has mining power: m_i

$$\sum_{i=1}^{n} m_i = 1$$

- Let the total mining power of selfish pool be: α
- Mining power of others: $(1 - \alpha)$
- Ratio of honest miners that choose to mine on pool's block: γ
- Ratio of honest miners that choose to mine on the other block : $(1 - \gamma)$

Analysis: Revenue Rate (Ideal Case)

- Revenue rate of each agent is the revenue earned by it for each block mined in the system.
- Let revenue rate of selfish pool be: r_{pool}
- Let total revenue rate of others be: r_{others}
- Revenue rate should be proportional to the mining power.

$$r_{pool} \propto \alpha$$

- Ideally, $r_{pool} + r_{others} = 1$

Analysis: Revenue Rate (Selfish Mining)

- Since selfish mining causes intentional branching in the blockchain, several mined blocks are not included in the blockchain.
- Total block generation rate drops.
- As a result, $r_{pool} + r_{others} < 1$

Analysis: Revenue Rate Ratio

- Actual revenue rate of each agent is the revenue rate ratio.
- Revenue rate ratio of an agent is defined as the ratio of its blocks out of the total blocks added to the main chain

$$R_{pool} = \frac{r_{pool}}{r_{pool} + r_{others}} = \frac{\alpha\left(1-\alpha^2\right)\left(4\alpha+\gamma(1-2\alpha)\right)-\alpha^3}{1-\alpha(1+(2-\alpha)\alpha)}$$

- Assuming honest majority,

$$0 \leq \alpha \leq \frac{1}{2}$$

- Selfish miners earn more revenue than their mining power if,

$$R_{pool} > \alpha$$

- For a given $\boldsymbol{\gamma}$, a selfish miners pool of size $\boldsymbol{\alpha}$ earns more revenue than its relative size for,

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

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

- Honest miners always mine on the pool's branch

  For $\gamma = 1, \quad 0 \leq \alpha \leq \frac{1}{2}$

- Honest miners randomly choose which branch to mine on

  For $\gamma = \frac{1}{2}, \quad \frac{1}{4} \leq \alpha \leq \frac{1}{2}$

- Honest miners never mine on the pool's branch

  For $\gamma = 0, \quad \frac{1}{3} \leq \alpha \leq \frac{1}{2}$

Problem with Bitcoin Protocol

- In case of multiple branches of the same length:
  - A miner mines and propagates only the first branch it received.
- There is no measure to guarantee a low γ.

- Sybil attack combined with selfish mining can lead to $\gamma \approx 1$.
  - In this case, a selfish pool of any size would earn more revenue than its mining power.
  - Rational miners will join the selfish pool.
  - The selfish pool would increase towards majority.

Solution: A simple change in the Bitcoin Protocol

- In case a miner encounters multiple branches of the same length:
  - He should propagate all the branches it receives.
  - He should choose which one to mine on uniformly at random.
- This change would yield $\gamma = \frac{1}{2}$.
- This change is backward compatible.

Selfish-mining attacks

- Surprising departure from previous assumptions
- Not yet observed in practice!
- Plausible reason: selfish-mining is detectable, could lead to a crash in exchange rates for Bitcoin

Punitive forking

- Suppose you want to blacklist transactions from address X
    - Freeze an individual's money forever
- Extreme strategy: announce that you will refuse to mine on any chain with a transaction from X

With α < 0.5, you'll soon fall behind the network

Feather-forking strategy

- To blacklist transactions from X, announce that you will refuse to mine directly on any block with a transaction from X
    - but you'll concede after ***n*** confirming blocks

- Chance of pruning an offending block, when n=1, is $\boldsymbol{\alpha^2}$

Response to feather forking

- For other miners, including a transaction from X induces an α^2 chance of losing a block
- Might be safer to join in on the blacklist
- Can enforce a blacklist with $\boldsymbol{\alpha < 0.5}$**!**

Success depends on convincing other miners you'll fork

Feather-forking: what is it good for?

- Freezing individual bitcoin owners
  - ransom/extortion
  - law enforcement?
- Enforcing a minimum transaction fee
  - Current transaction fees are low (about 2% of revenue)
  - But may become significant when mining reward becomes low

Summary

- Miners are free to implement any strategy
- Very little non-default behavior in the wild
- Game-theoretic analysis necessary
- Recent works in this direction. See, e.g.: [Badertscher-Garay-Maurer-Tshudi-Zikas, EUROCRYPT'18]

Puzzles

Puzzles are the core of Blockchains

- Determine the incentive system, and nature of puzzles determines behavior of miners
- Basic features of Bitcoin's proof-of-work puzzle (recap)
  - Puzzle is difficult to solve, so large-scale attacks are difficult
  - … but not too hard, so honest miners are compensated
- What other features could a puzzle have?

Today (and next time…)

- Alternative puzzle designs

  Used in practice, and research proposals

- Variety of possible goals

  ASIC resistance, pool resistance, environmental-friendliness, intrinsic benefits...

- Essential security requirements

Basic Puzzle Requirements

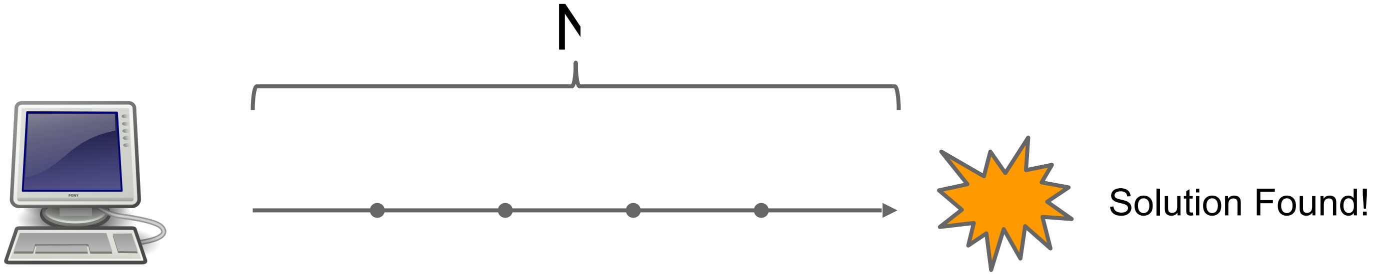
Puzzle requirements

- Cheap to Verify
  - since other users have to verify solutions
- Adjustable difficulty
  - E.g., due on increase in hash rate or more users
- In PoW puzzles, chance of winning should be proportional to computing power (e.g., hash power in Bitcoin)
  - Large players get only proportional advantage
  - Even small players get proportional compensation

Bad PoW puzzle: a *sequential* puzzle

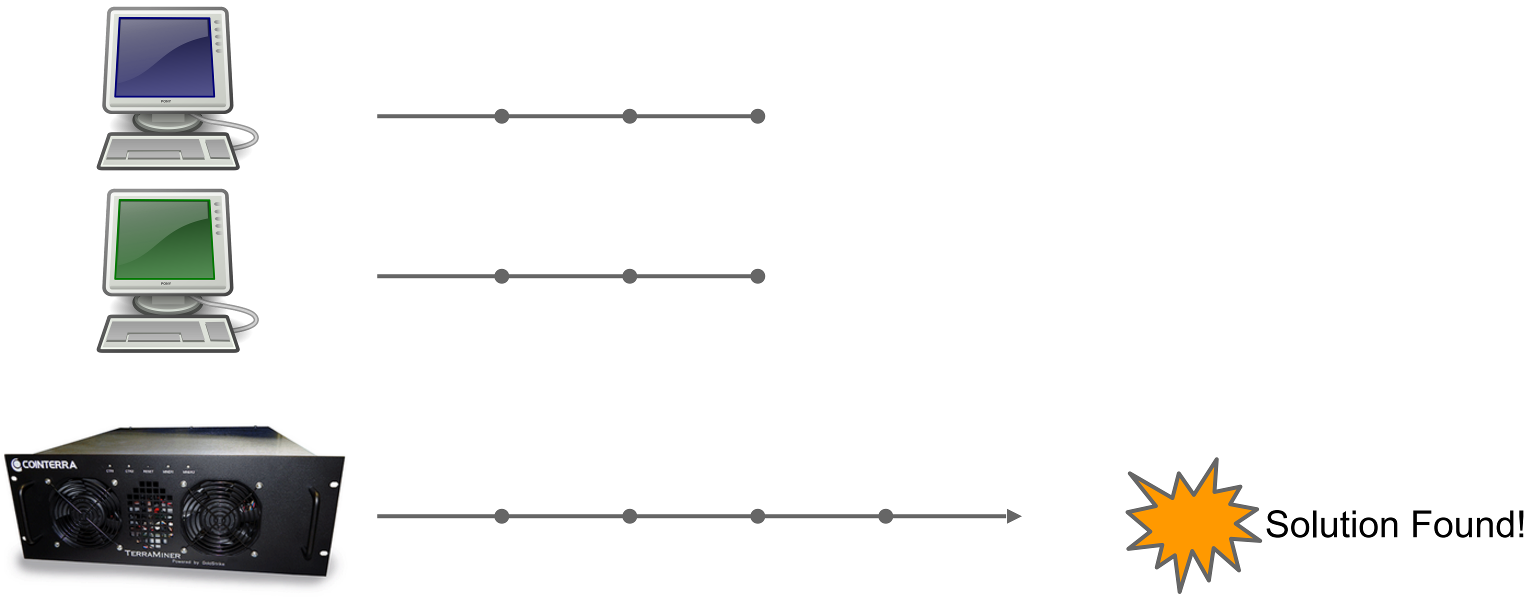
Consider a puzzle that takes N steps to solve

a "Sequential" Proof of Work

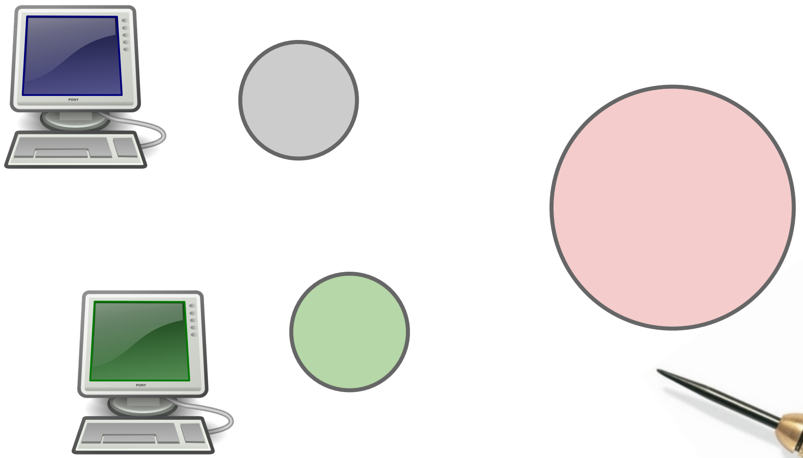


Bad PoW puzzle: a *sequential* puzzle

Problem: fastest miner **always** wins the race!



Good PoW puzzle → Weighted sample



This property is sometimes called “progress-free”

ASIC Resistant (PoW) Puzzles

ASIC resistance - Why? (1 of 2)

Goal: Ordinary people with idle laptops, PCs, or even mobile phones can mine!

Lower barrier to entry

Approach: Reduce the gap between custom hardware and general purpose equipment

ASIC resistance - Why? (2 of 2)

Goal: Prevent large manufacturers from dominating the game

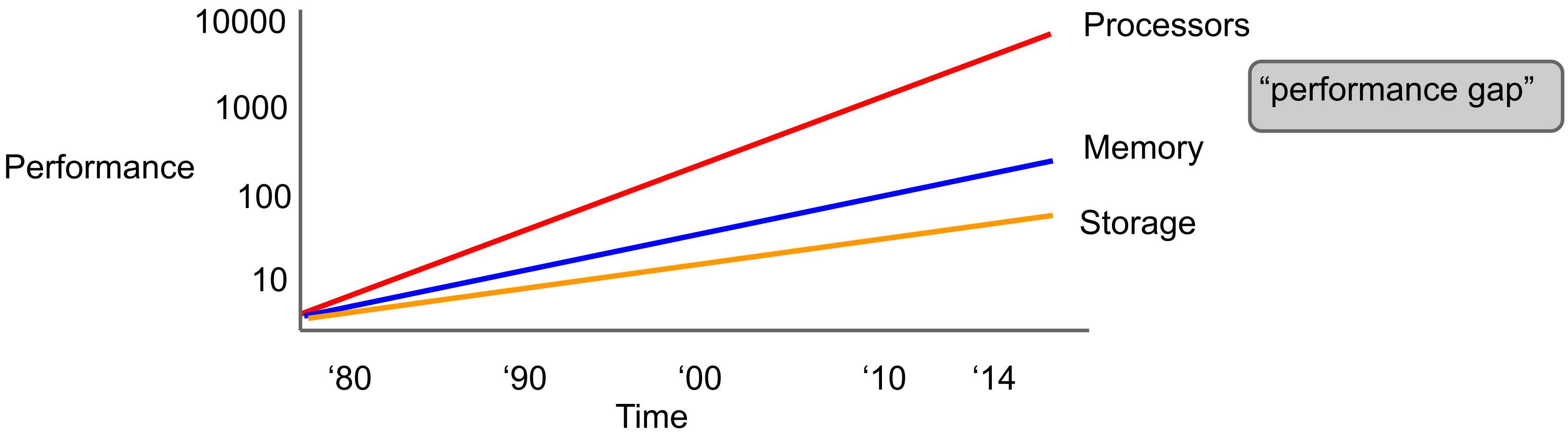
“Burn-in” advantage

In-house designs

Approach: reduce the “gap” between future hardware and the custom ASICs we already have

Memory hard puzzles

Premise: the cost and performance of memory is more stable than for processors



scrypt

Colin Percival, 2009

- Memory hard hash function
  - ***Constant time/memory tradeoff***
  - Memory consumes a large amount of on-chip area. High memory requirement => small number of hashing engines on special-purpose chips
- Widely used alternative PoW puzzle (e.g., Litecoin)
- Also used in Password-hashing

1. Fill memory with random values
2. Read from the memory in random order

scrypt - step 1 of 2 (write)

Input: **x**

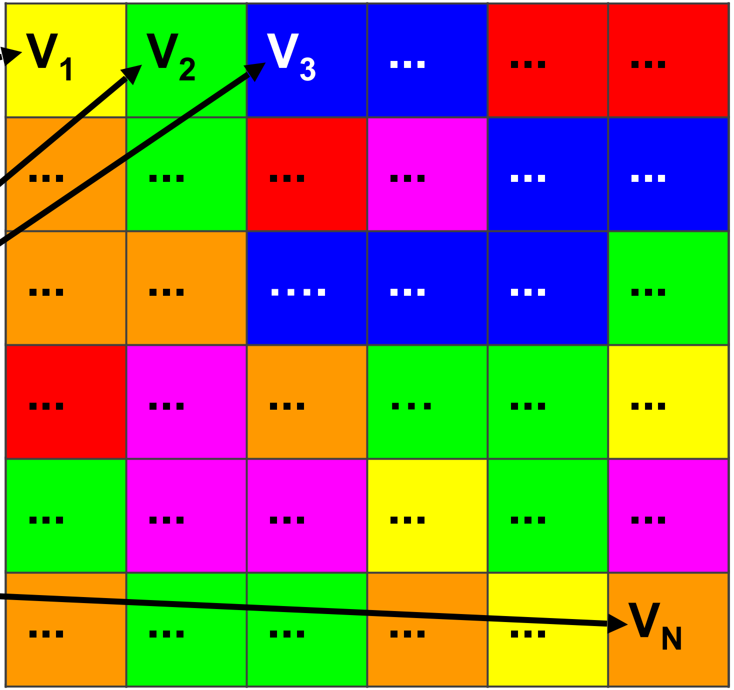
$V_1 = H(X)$

$V_2 = H(V_1) = H(H(X))$

$V_3 = H(V_2) = H^3(X)$

...

$V_N = H^N(x)$



scrypt - step 2 of 2 (read)

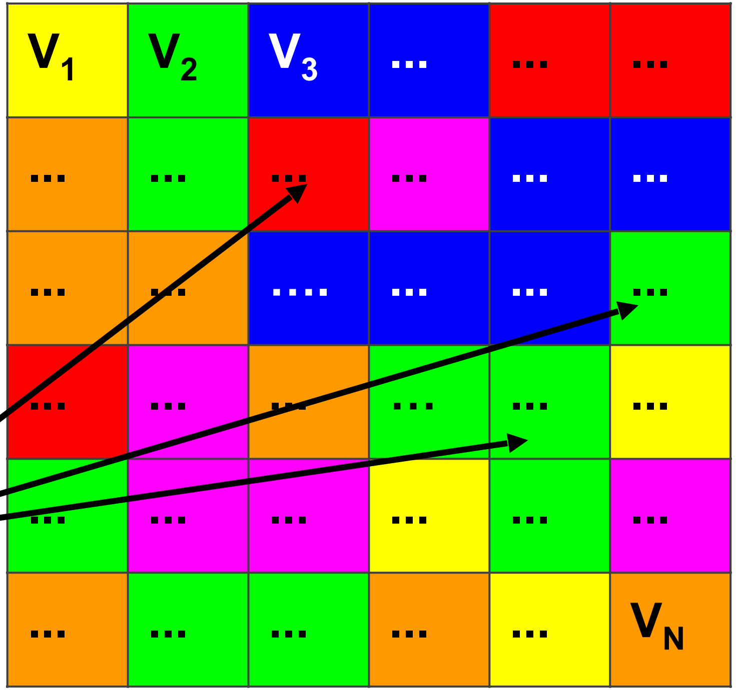
Input: **x**

`A := H^{N+1}(X)`

For N iterations:

```
    i := A     mod N
    A := H(A xor V_i)
```

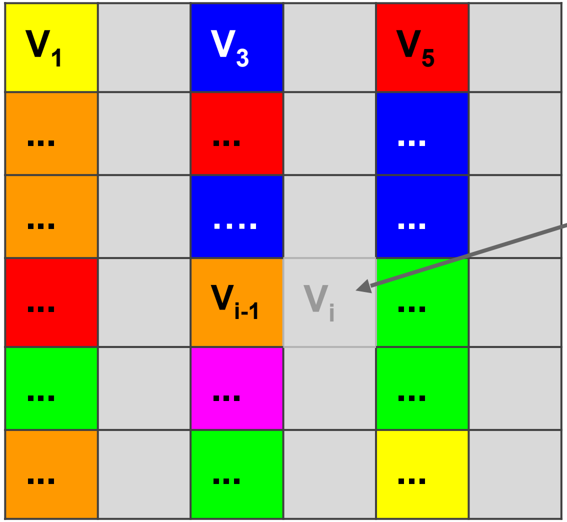
Output: A



scrypt - time/memory tradeoff

Why is this memory-hard?

Reduce memory by half, 1.5x the # steps



Need to access V_i where i is even?

Access V_{i-1}

Compute $V_i = H(V_{i-1})$

scrypt

Disadvantages: Also requires N steps, N memory to check

Is it actually ASIC resistant?

- scrypt ASICs ***are*** already available
- Exploit time-memory trade-offs, lower values of N, etc.

Academic research

- Many subsequent candidates: Argon2i (winner of PW-hashing contest), Ballon-Hashing, etc.
- Proofs of memory hardness in various models using graph pebbling techniques (see, e.g., Alwen-Serbeninko'15 and many subsequent works)