Blockchains & Distributed Ledgers

Lecture 07

Dimitris Karakostas

Slide credits: DK, Aggelos Kiayias, Dionysis Zindros, Christos Nasikas, Aikaterini-Panagiota Stouka

[Previously]

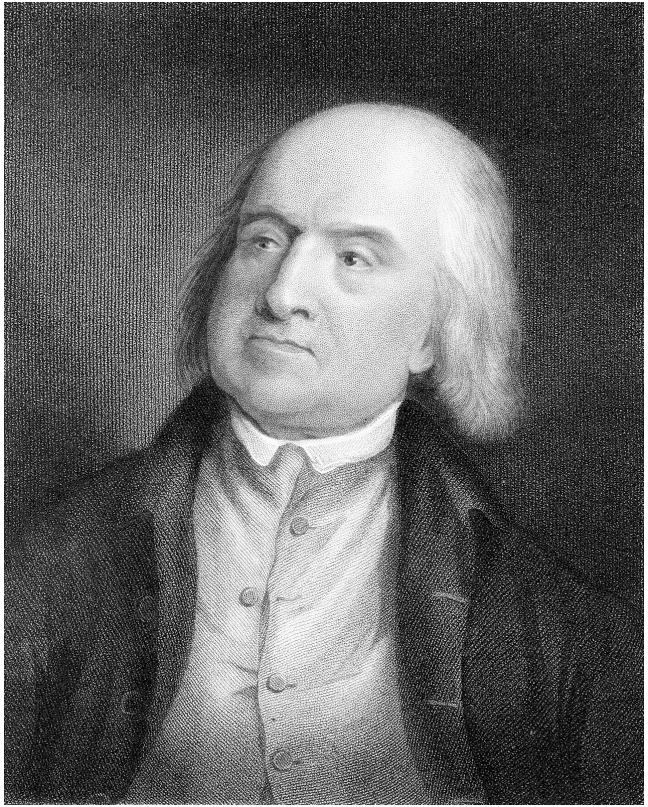
- Participating in a blockchain/distributed ledger system costs
  - Electricity
  - Hardware equipment
  - Network availability
- Security analysis so far: participants are either honest 😇 or adversarial 😈
  - **Honest** parties follow the protocol precisely
  - **Adversarial** (corrupted) parties can follow any algorithm they want
- If **majority** of power (computational/stake) is **honest**, then the ledger is **secure**
  - Persistence and liveness are guaranteed

Adam Smith (1723-1790)

The capitalist principle

- The invisible hand of the market
  - *“he intends only his own gain; and he is in this, as in many other cases, led by an invisible hand to promote an end which was no part of his intention”* (Adam Smith. *The Wealth of Nations.)*
- People chase after their own profit…
- … and in doing so can, seemingly inadvertently, create external results
  - For example, the baker makes bread and sells it to gain money, but also feeds the family that buys it

Jeremy Bentham (1748-1832)



Utilitarianism

- The principle of utility
  - *“that property in any object, whereby it tends to produce benefit, advantage, pleasure, good, or happiness” (Jeremy Bentham. An Introduction to the Principles of Morals and Legislation)*
- Utility: the property that each person strives to maximize
  - Money: people try to increase their financial wealth
  - Social acceptance: people try to increase their fame or status
  - Divine acceptance: people try to act as close as possible to what their religion dictates

From the invisible hand to incentives

- People have some **utility**
- In the capitalist system, the utility is monetary **profit**
- Therefore, profit seeking is the **rational** behavior
  - Every player is motivated to *always* maximize their own profit

How to design a system s.t. rational parties follow the prescribed protocol?

The economics of consensus

- Running a consensus protocol involves multiple participants with possibly **conflicting interests**
- What if participants, instead of being honest/malicious, instead **follow their best financial interest**?
- How are participants **incentivized** to engage?
- Are the desired properties of distributed ledgers (consistency, liveness) the result of the participants’ **rational engagement**?

Bitcoin Incentives

Mining incentives

A miner is incentivized to mine in 2 ways:

1. Transaction Fees
2. (Fixed) Block Rewards

Mining fees

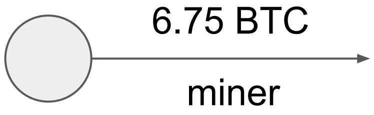
- [Recall] A transaction defines inputs and outputs
- Value conservation law:
  - *<sum of input values>* ≥ *<sum of output values>*
  - No amount value is created by a simple transaction
- *Transaction fees*: the remaining money from the conservation law of value
  - $tx\_fees = \Sigma_{i \in in(tx)} w(i) - \Sigma_{o \in out(tx)} w(o)$, where w(.) is value
- Each transaction’s fees are claimed by the miner who included the respective transaction in their block

Mining block rewards

- A miner is given a **fixed reward per block** they create
    - The only way to create *new* coins
    - In 2022 Bitcoin: 6.25 BTC
- The *block reward* and the *transaction fees* are claimed by the miner via a **coinbase transaction**
- Example:
    - aggregate tx fees = 0.5 BTC
    - block reward = 6.25 BTC
    - value of coinbase tx output = 6.75 BTC

The coinbase transaction

- The **coinbase transaction** is the transaction by which a miner is paid their rewards (tx fees + fixed block reward)
- There can only be **one coinbase transaction** per block
- It is the **first** transaction that appears in the block
- It has *no inputs*
- This is *the only way* new bitcoins are generated



The coinbase transaction

- As it does not have any inputs, a coinbase tx’s scriptSig can be anything
- scriptSig is used for certain block metadata:
    - The block height (verified for validity)
    - The name of the mining pool/user that mined the block
    - Extra entropy (nonce)
    - Signalling for protocol updates (whether a miner is in favour of an upgrade or not, e.g., a hard fork)

Coinbase transaction validity

- [Recall] A tx consumes existing outputs (UTxOs) and creates new UTxOs
  - The induction step
- Coinbase tx is the **induction basis** for transaction validity
- Has no inputs, so does not conform to the conservation law of value!
- Is not part of the mempool
  - only included in blocks directly
- When a Bitcoin block is confirmed, the coinbase is checked for validity:
  - It is the first in the block
  - There’s only one of it
  - **output value ≤ block reward + block tx fees**
- A malicious miner cannot generate more money (than determined by the protocol)

Money supply in Bitcoin

- The **money supply** in Bitcoin is **algorithmically predetermined**
  - Upper-capped total amount of tokens
  - A mechanism akin to 19th century gold standard
- Achieved with an **algorithm** known beforehand to all
- Concretely:
  - The coinbase of **genesis** has reward **50 BTC**
  - Each next block has reward equal to its previous block
  - Every 210,000 blocks (on expectation: *4 years*), the reward is **halved**
  - The duration during which rewards stay the same is known as an **era**

Money supply in Bitcoin

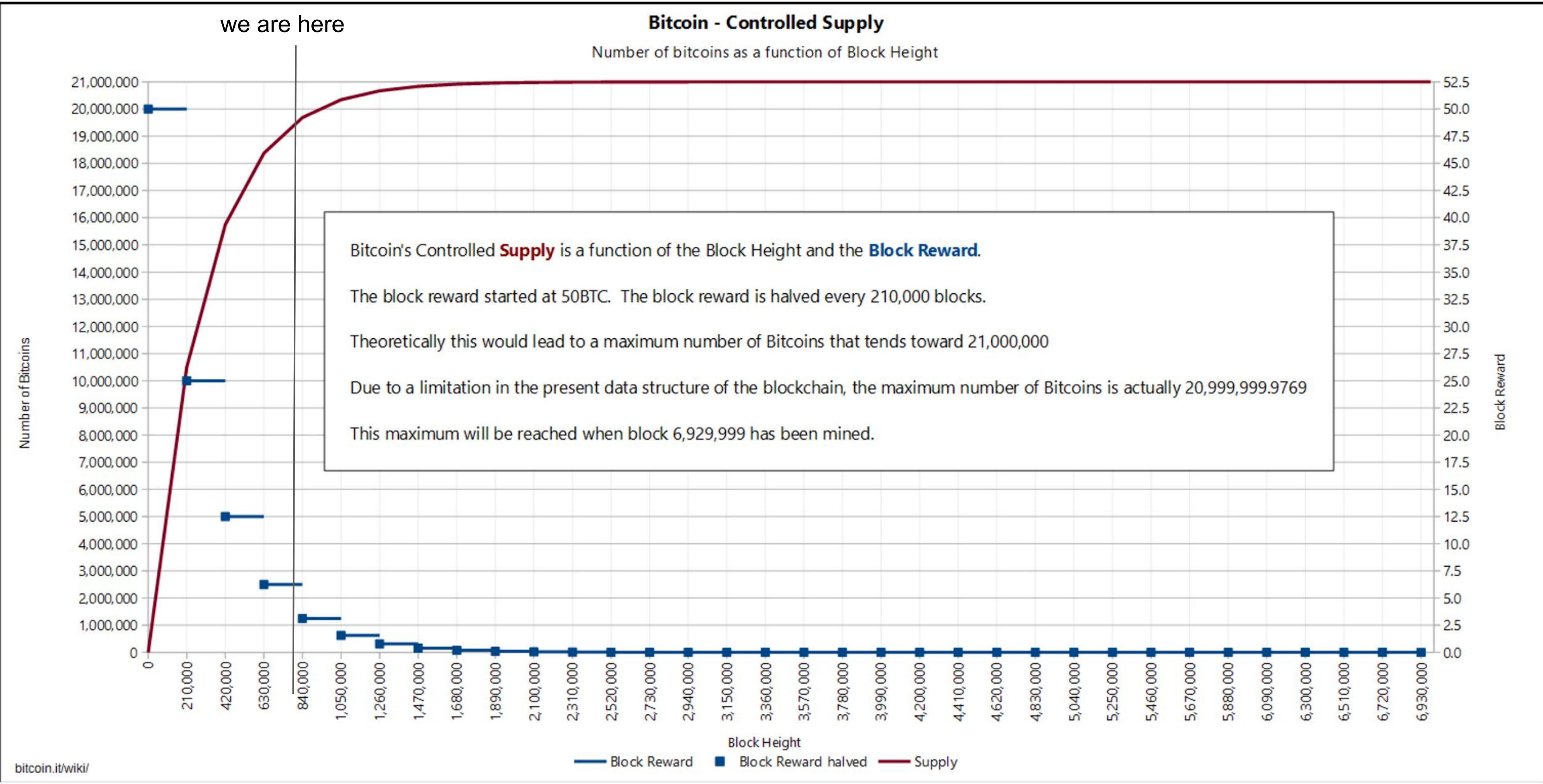
era duration in blocks

number of eras
until reward is negligible

genesis block reward

$$\frac{\sum_{i=0}^{32} 210000 \lfloor \frac{50.10^8}{2^i} \rfloor}{10^8}$$

satoshi / bitcoin



we are here

Bitcoin's Controlled **Supply** is a function of the Block Height and the **Block Reward**.

The block reward started at 50BTC. The block reward is halved every 210,000 blocks.

Theoretically this would lead to a maximum number of Bitcoins that tends toward 21,000,000

Due to a limitation in the present data structure of the blockchain, the maximum number of Bitcoins is actually 20,999,999.9769

This maximum will be reached when block 6,929,999 has been mined.

bitcoin.it/wiki/

https://en.bitcoin.it/wiki/Controlled_supply

Money distribution in Bitcoin

- Halving mechanism **favors disproportionately** the early miners
  - **50% of all bitcoins** that will ever be produced were created in **first 3 years**
- **Extremely centralized** ownership distribution
  - (2013) 2,300 addresses controlled 50% of all tokens
  - (2022) 2,021 addresses (0.0047%) control 41% of all tokens
    - In the (extremely unequal) real world, 520,000 people (0.01%) control 11% of all wealth
  - (2022) Bitcoin Gini coefficient w.r.t. all addresses: 0.956
    - The worst real-world economy (in terms of Gini): 0.512

https://bitinfocharts.com/top-100-richest-bitcoin-addresses.html

Bitcoin denominations

- **1 bitcoin** is divisible up to **10^{-8}**
- 10^{-8} bitcoin = 1 **satoshi**
- 1 satoshi = 0.00000001 BTC
- 1 BTC = 100,000,000
- The bitcoin implementation stores **integers** in the output edges, representing the number of satoshis
    - no floating-point errors

Ways to mine

- **CPU**: standard processors
- **GPU**: graphics card (high parallelization)
- **ASIC**: specialized hardware for mining

Is it profitable to mine? Probably not...

- November 2022:
  - CPU Intel i7-8700K:
    - Initial hardware cost: *$360*
    - Profit: -*$0.23 / day*
  - GPU NVIDIA GTX 1050 Ti:
    - Initial hardware cost: *$160*
    - Profit: -*$0.26 / day*
  - Specialized hardware AntMiner S17+:
    - Initial hardware cost: *$1,500*
    - Profit: -*$8.49 / day*
  - AntMiner Z15:
    - Initial hardware cost: *$5,600*
    - *Profit: $0.59 / day*
- Even in 2013:
  - A high-end (24/7 running) Nvidia GPU could yield in practice 1BTC (~$100) in 6 months

Electricity: $0.2/kwh
Bitcoin: $20,000
https://www.nicehash.com/profitability-calculator

Mining Games

- Miners are incentivised (via rewards) to follow the protocol
- Does this ensure that they choose to execute the (honest) protocol?
  - Is the reward mechanism *“incentive compatible”*?
- Protocol can be
  - Dominant strategy
  - Nash equilibrium

Dominant Strategy example

Split or Steal Game (payoff table: *<payoff of A> / <payoff of B>)*

	Split (B)	Steal (B)
Split (A)	50 / 50	0 / 100
Steal (A)	100 / 0	1 / 1

Dominant Strategy example

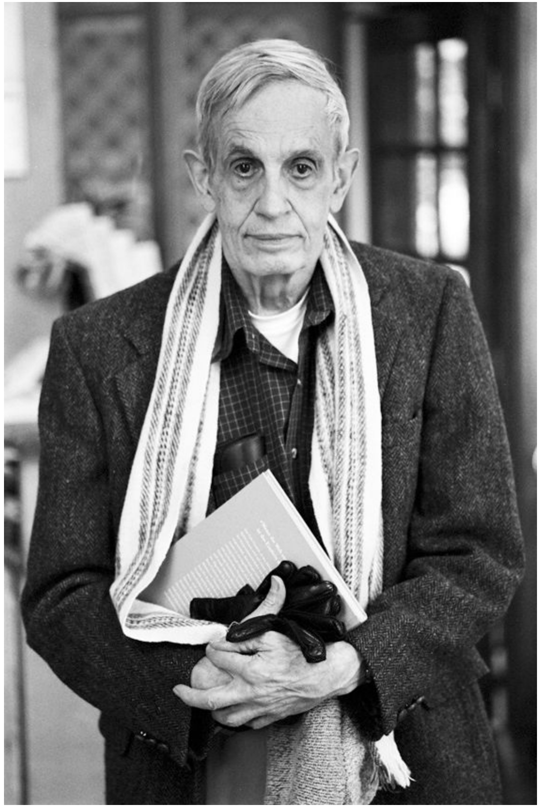
Split or Steal Game (payoff table: *<payoff of A> / <payoff of B>)*

	Split (B)	Steal (B)
Split (A)	50 / 50	0 / 100
Steal (A)	100 / 0	1 / 1

- Stealing is dominant strategy
  - For player A: 100 > 50 (if B splits), 1 > 0 (if B steals)
  - Same for B
- Steal/Steal is *sub-optimal* strategy
  - Split/Split yields higher rewards for both
- See also: prisoner's dilemma

Nash Equilibrium

John Forbes Nash Jr. (1928-2015)

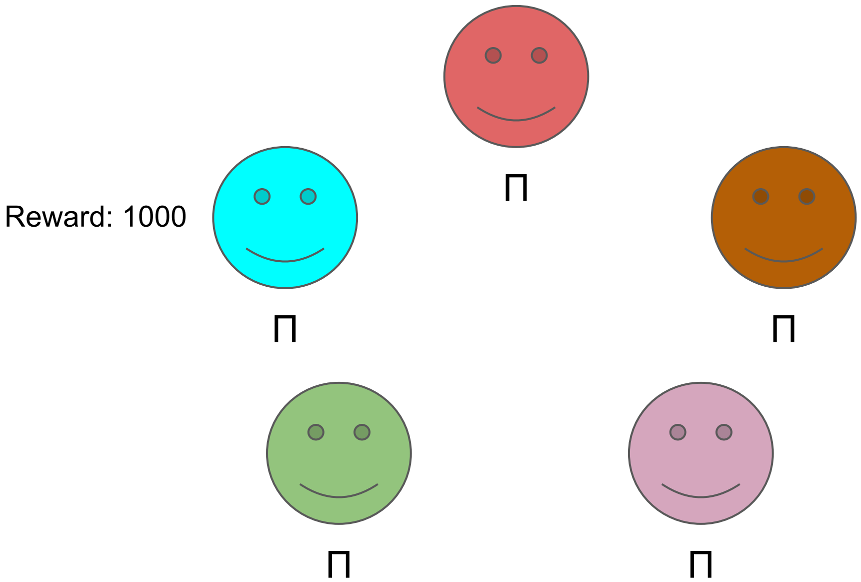


Nash Equilibrium

- Utility of a participant:
  - function
  - input: a vector of strategies of all participants
  - output: a real number that represents the gains of this participant at the end of the execution
- Participants are rational:
  - they want to maximize the utility they obtain at the end of the execution

Nash Equilibrium

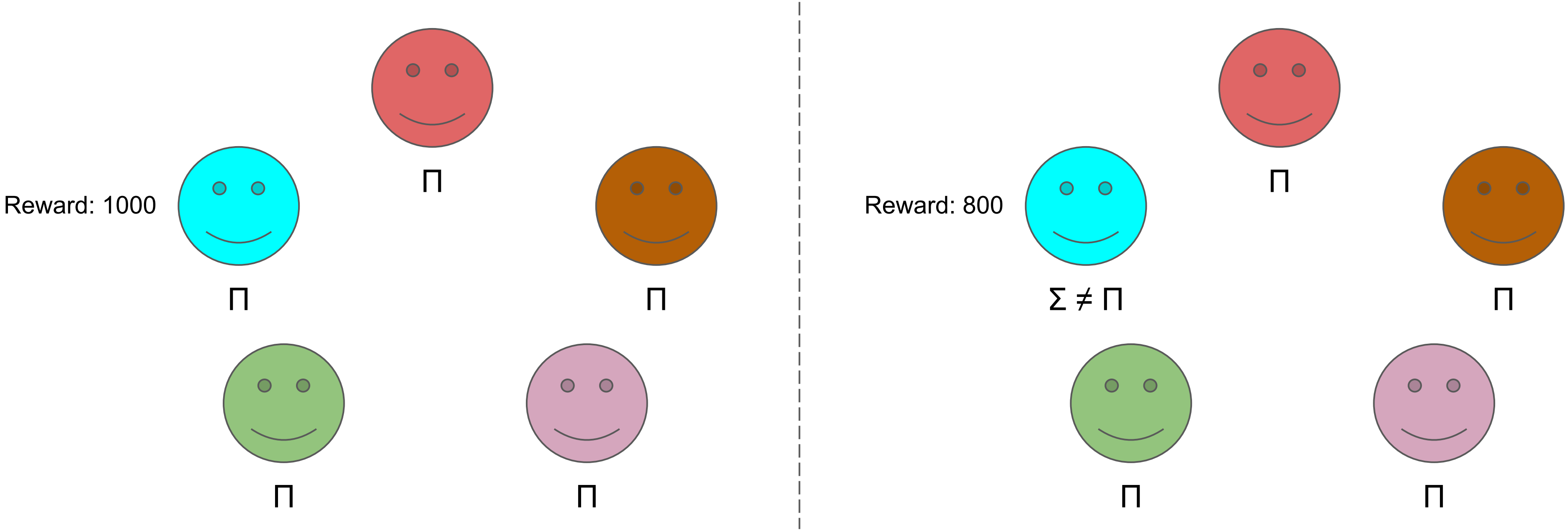
All participants are **rational**; they want to *maximize their utility*.



Nash Equilibrium

All participants are **rational**; they want to *maximize their utility*.

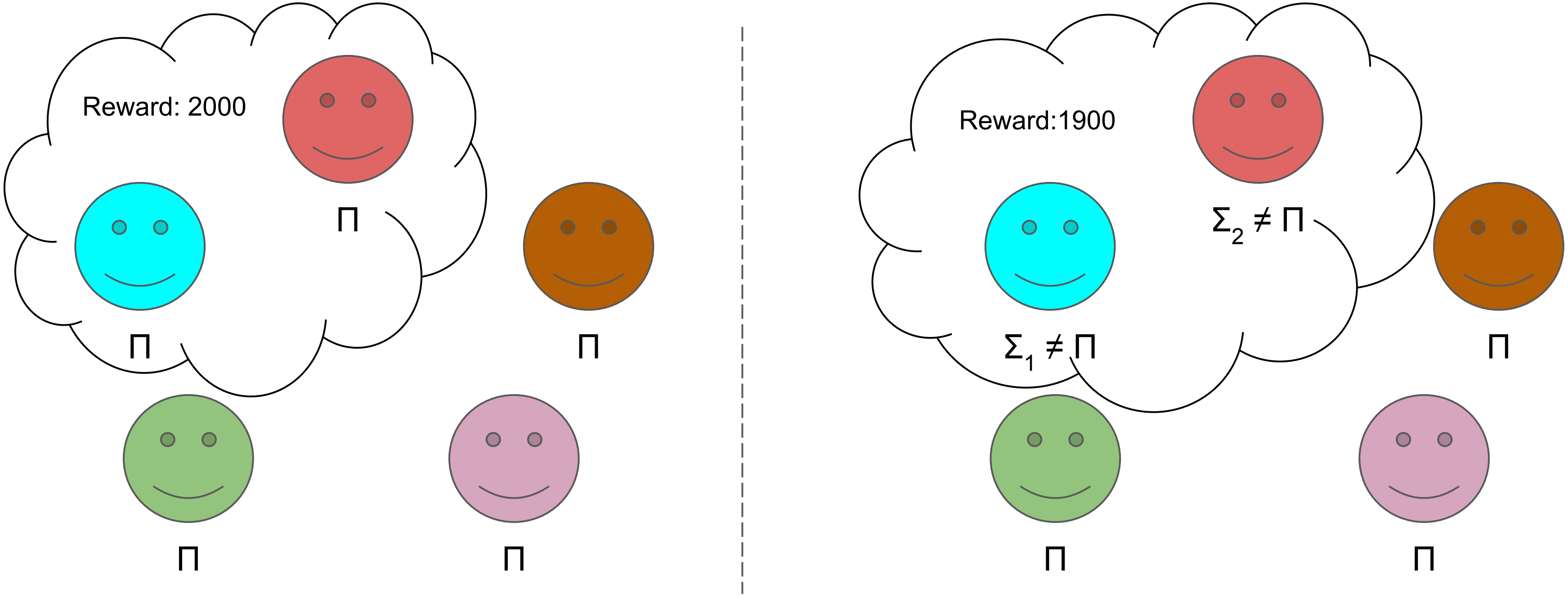
Nash Equilibrium: No party can increase its utility by deviating from Π.



Generalisation to Coalitions

All participants are **rational**; they want to *maximize their utility*.

Nash Equilibrium: No coalition can increase its aggregate utility by deviating from Π.

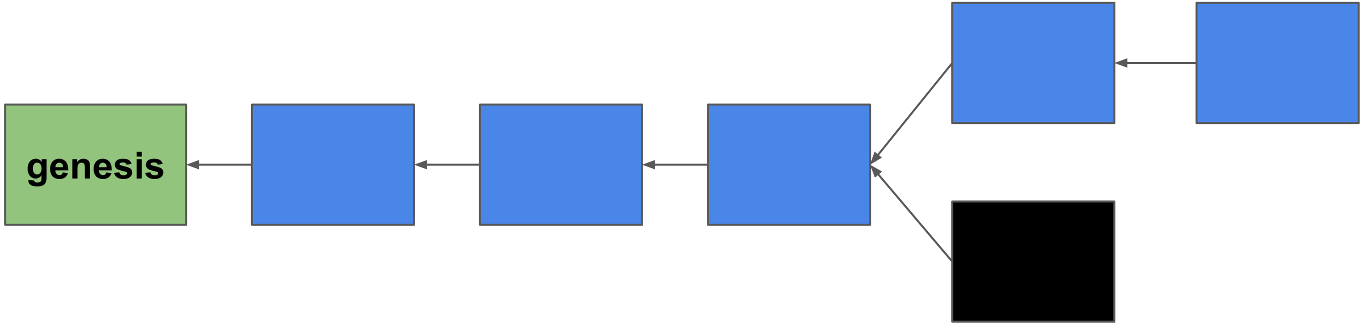


Is Bitcoin a Nash Equilibrium ?

- What can be the utility in Bitcoin?
  - Absolute rewards
  - Relative rewards
- How can utility be defined in a probabilistic protocol?
  - expectation
  - events of high probability

Absolute Rewards, I

- Fix:
  - the algorithms followed by all the participants
  - the outcome of all the randomness used by participants
  - a time limit (finite execution) of the Bitcoin protocol
- Obtain a unique outcome of the protocol
- Each block of the adopted chain gives a reward to its producer

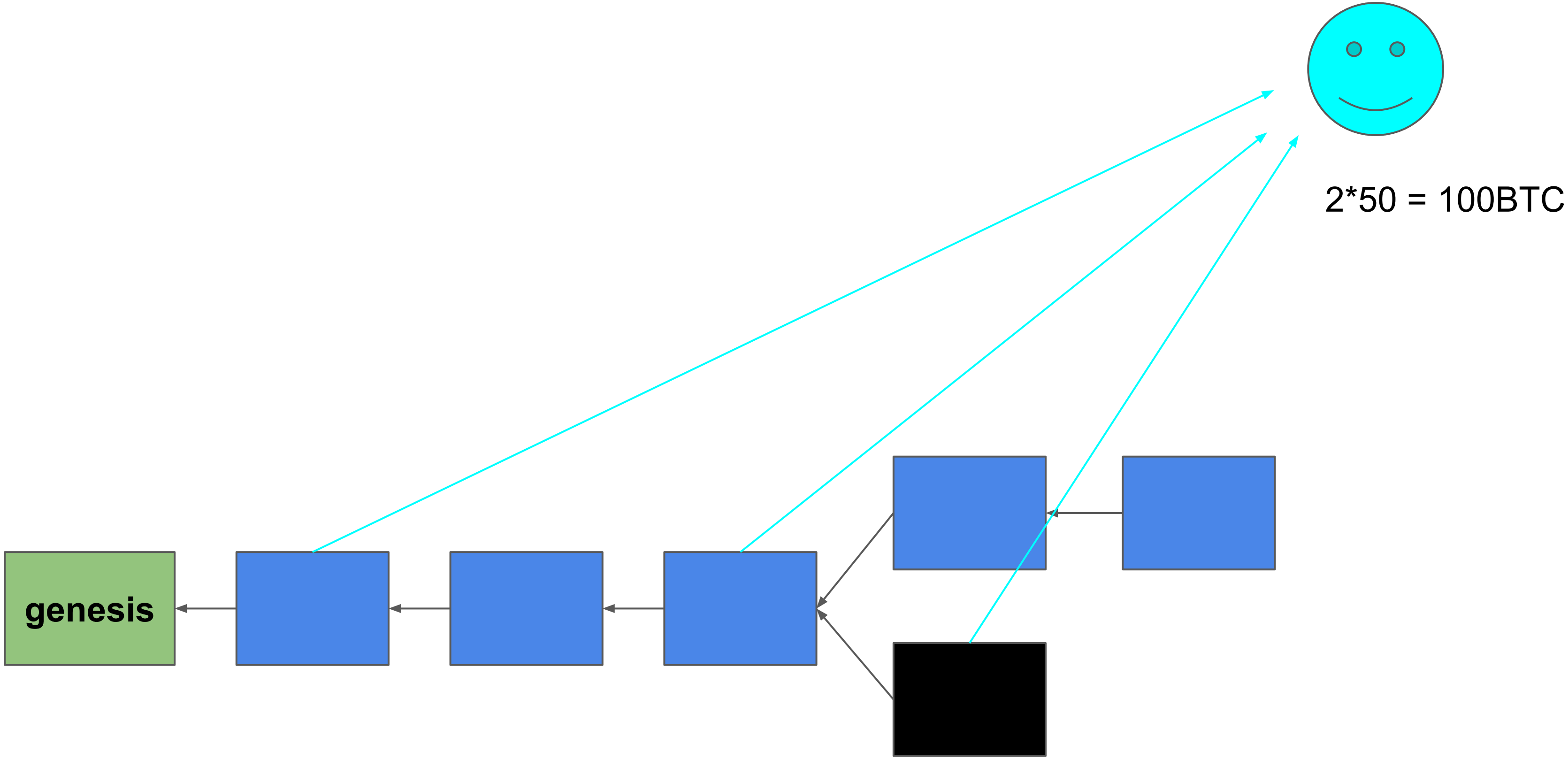


Absolute Rewards, II

- **Absolute rewards**: The utility of a coalition is equal to *the number of BTC* that it has obtained at the end of the execution

$$U_i = \langle \textit{sum rewards of } P_i \rangle$$

Absolute Rewards, III

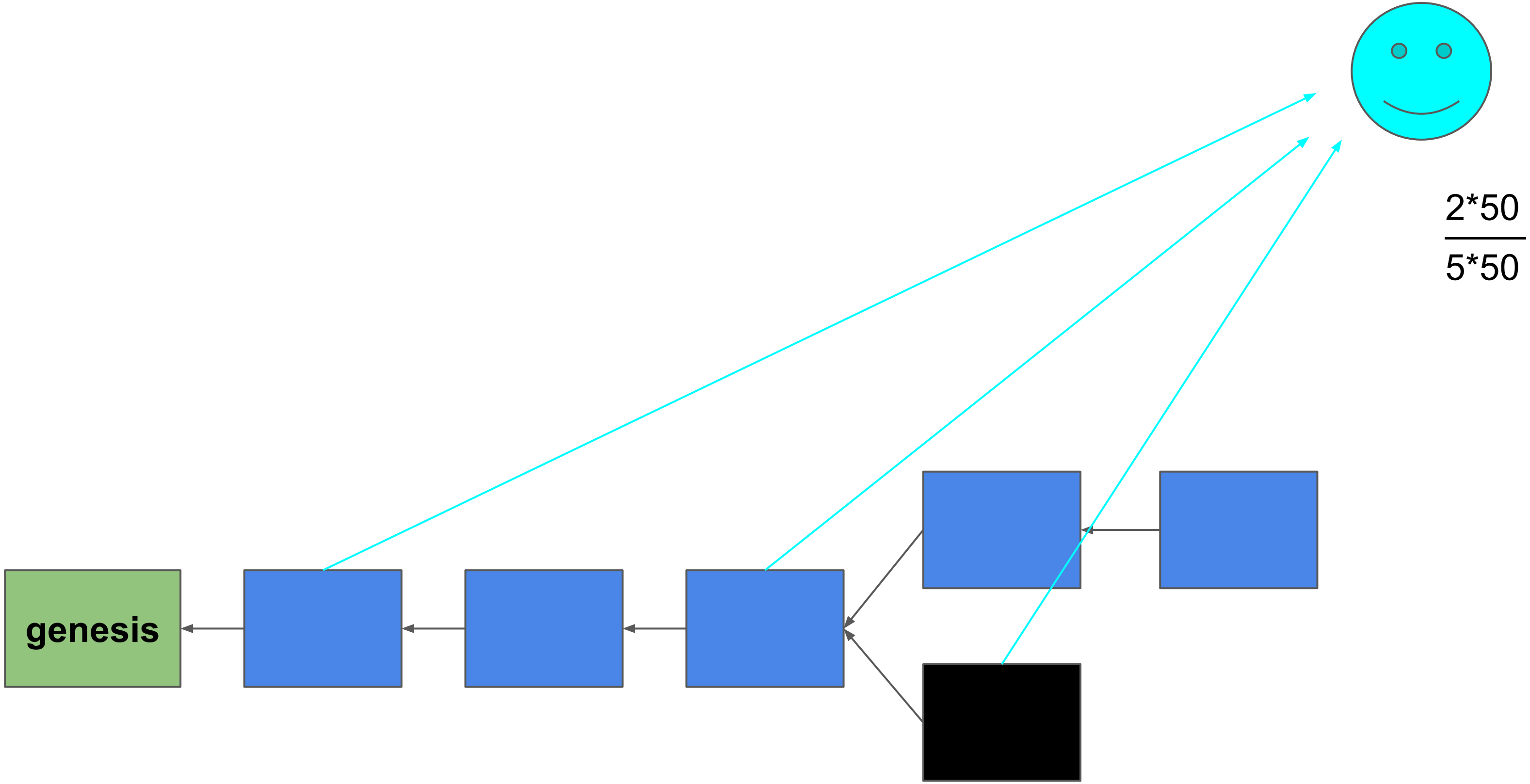


Relative Rewards, I

- **Relative Rewards**: The utility of a coalition in Bitcoin is equal to the *amount of BTC* that it earns, *divided* by the total *amount of BTC that all participants* receive at the end of the execution

$$U_i = \langle \textit{sum rewards of } P_i \rangle \; / \; \langle \textit{sum rewards of all parties} \rangle$$

Relative Rewards, II



$$\frac{2*50}{5*50}$$

Utility in probabilistic protocols

- Given the strategies of all the participants, the outcome of the Bitcoin execution is a random variable
- The utility of a coalition (parameterized by an execution) is also a random variable
- How to resolve this?
  - expectation that determines the expected value of utility
  - events that happen with high probability

Bitcoin and Equilibria

- A certain modelling of the Bitcoin protocol is a Nash equilibrium
  - **Utility** is equal to expected number of **absolute rewards**
  - Block **difficulty** is **fixed**
  - Expected number of blocks is proportional to mining power (delivered by a Bitcoin execution)
- Bitcoin is not a Nash equilibrium w.r.t another modelling
  - **Utility** is equal to expected number of **relative rewards**
  - Block **difficulty** is **fixed**
  - Expected number of blocks is proportional to mining power (delivered by a Bitcoin execution)
  - **Selfish mining** strategy is more profitable

Kroll et al. in “The economics of Bitcoin mining, or Bitcoin in the presence of adversaries” (2013)
Ittay Eyal and Emin Gün Sirer. "Majority is not enough: Bitcoin mining is vulnerable."(2014)

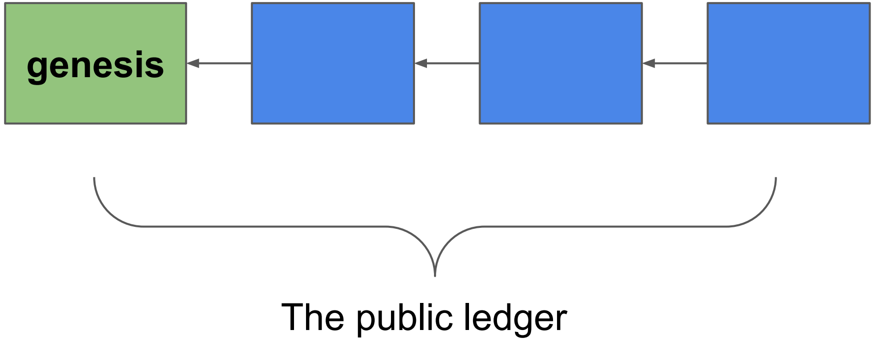
Game Theoretic Attacks

Selfish Mining

- A strategy that enables a coalition to collect more (expected) relative rewards by deviating from the honest protocol
- Attacker maintains a private chain, strategically releasing its blocks to deny honest parties' blocks from being adopted to the "main chain"

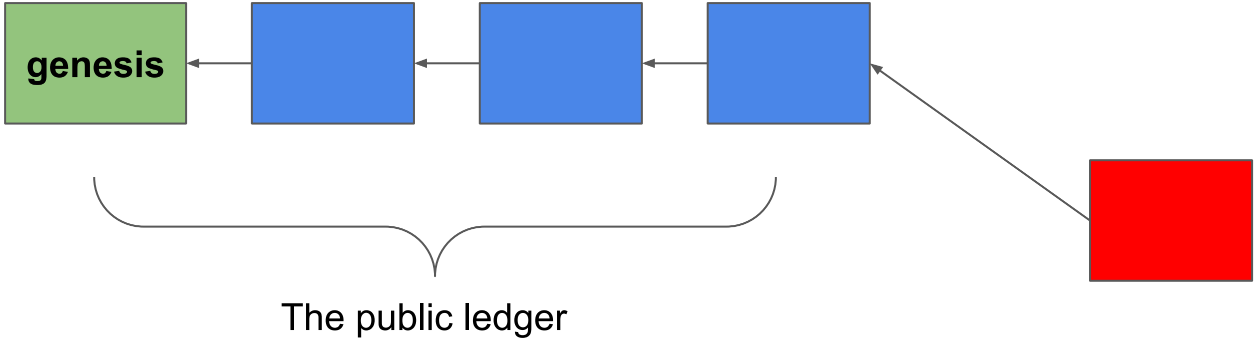
Selfish Mining, step 1

- $\mathcal{A}$ adopts the longest chain and tries to extend it.



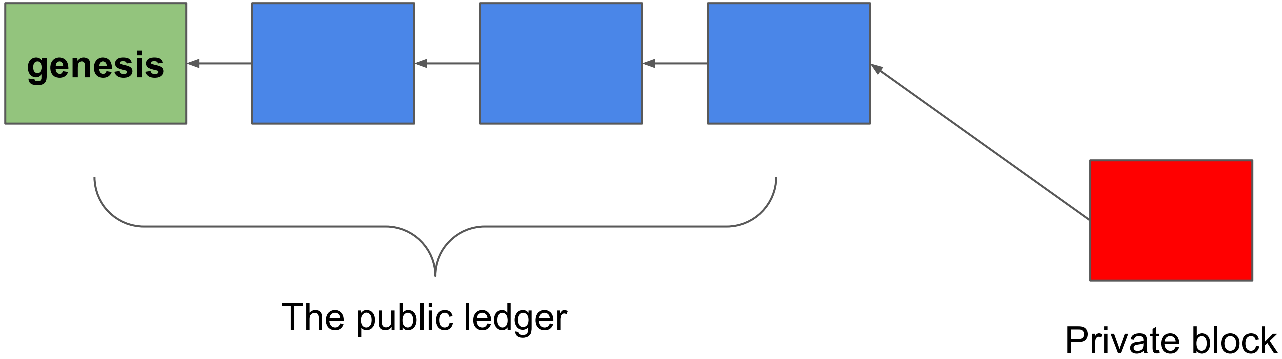
Selfish Mining, step 2a

- $\mathcal{A}$ produces a block before the honest parties.



Selfish Mining, step 2a

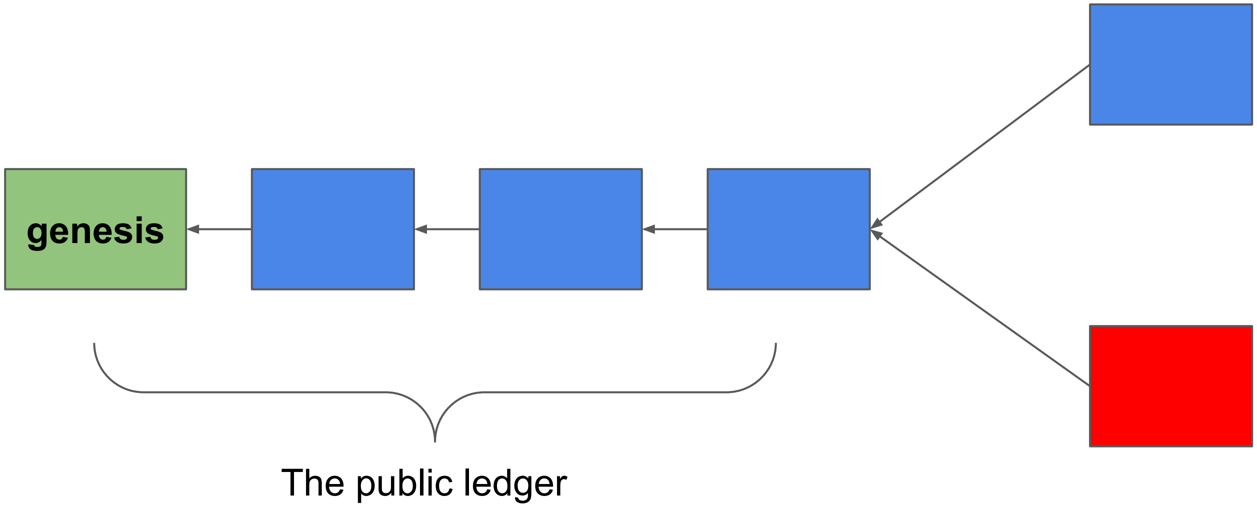
- $\mathcal{A}$ produces a block before the honest parties.
  - $\mathcal{A}$ keeps the block private…



*Depending on the attacker's network dominance, A might publish it when the honest chain is one block behind.

Selfish Mining, step 2a

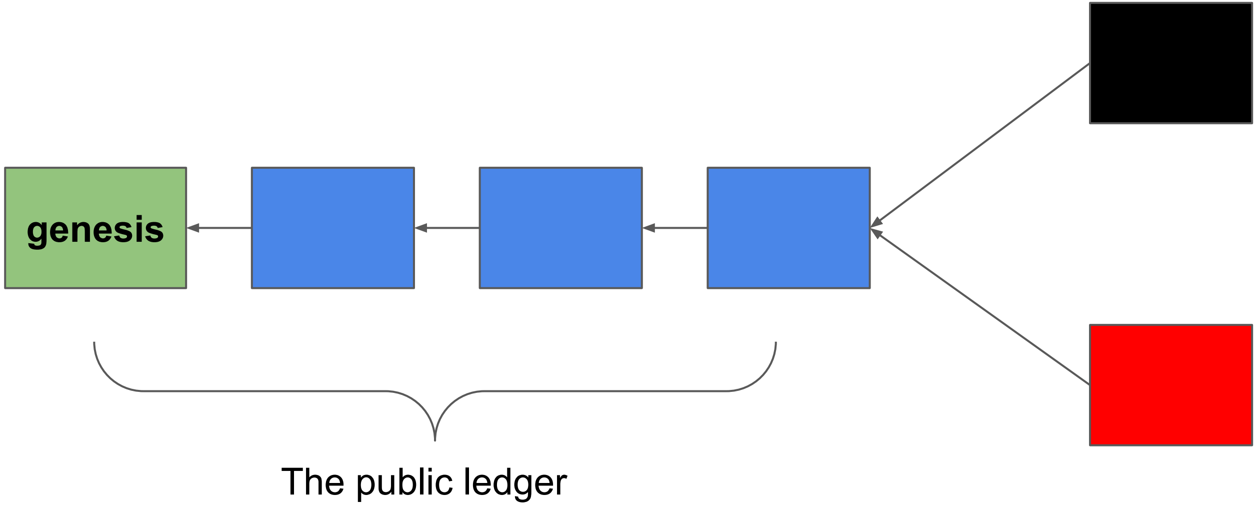
- $\mathcal{A}$ produces a block before the honest parties.
- $\mathcal{A}$ keeps the block private… until a competing honest block is created.*



Selfish Mining, step 2a

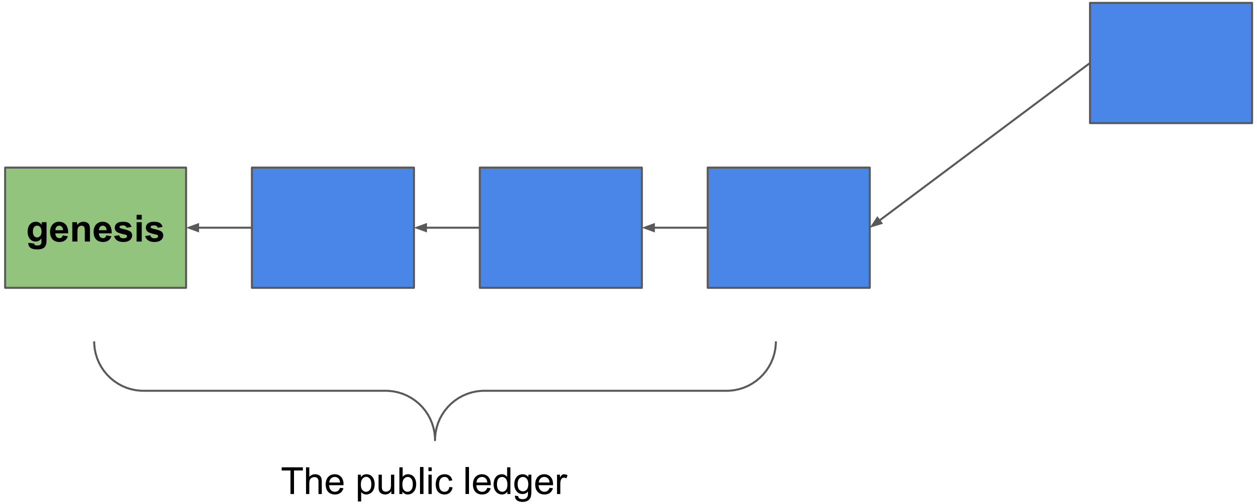
*Depending on the attacker’s network dominance, A might publish it when the honest chain is one block behind.

- $\mathcal{A}$ produces a block before the honest parties.
- $\mathcal{A}$ keeps the block private… until a competing honest block is created.*
- If the other parties adopt the adversarial block, selfish mining attack is successful.



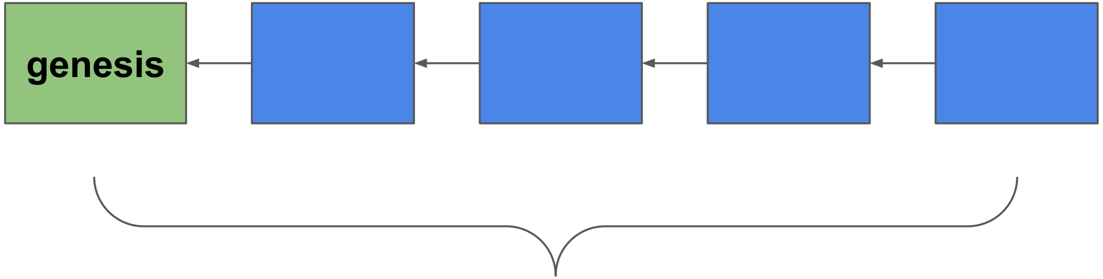
Selfish Mining, step 2b

- The honest parties produce a block before $\mathcal{A}$.



Selfish Mining, step 2b

- The honest parties produce a block before $\mathcal{A}$.
  - $\mathcal{A}$ adopts the block and goes to step 1.



The public ledger

Selfish Mining

- More generally: $\mathcal{A}$ will be capable of censoring blocks, if
  a. $\mathcal{A}$'s chain gets two blocks ahead of the public chain
  b. $\mathcal{A}$ manages to deliver its block to the other parties first
- In principle, when an honest party receives two chains of the same length, it chooses the first that it received

Selfish Mining, Analysis

- $\mathcal{A}$ contributes only towards censoring blocks and not towards extending the public ledger
- During the attack, the total (expected) number of blocks in the public ledger is less than the expected number of blocks when $\mathcal{A}$ follows the protocol
- If $\mathcal{A}$ does not manage to deliver its block first, $\mathcal{A}$ loses the block’s rewards
    - However, it is assumed that $\mathcal{A}$ has *some* control over the message delivery schedule

Selfish Mining, Analysis

- Consider an execution that consists of “block rounds"
- Assume that $\mathcal{A}$ always wins the network race vs. public blocks
- $\mathcal{A}$ has probability α to produce the next block:
  - Honest play:
    1. *n* rounds → *n* blocks
    2. attacker creates α·n blocks in expectation
    3. Utility: (Relative Rewards) = α, (Absolute Rewards) = α·*n*

Selfish Mining, Analysis

- Consider an execution that consists of “block rounds"
- Assume that $\mathcal{A}$ always wins the network race vs. public blocks
- $\mathcal{A}$ has probability α to produce the next block:
  - Honest play:
    1. *n* rounds → *n* blocks
    2. attacker creates α·n blocks in expectation
    3. Utility: (Relative Rewards) = α, (Absolute Rewards) = α·*n*
  - Selfish play:
    1. *n* rounds → (1-α)·*n* blocks
    2. attacker creates α·*n* of those blocks
    3. Utility: (Relative Rewards) = α / (1-α), (Absolute Rewards) = α·*n*
- In a static difficulty setting absolute rewards are unaffected... but relative rewards increase!

Bitcoin and Equilibria, III

- If difficulty adjusts…
- Selfish mining will impact chain growth
- The difficulty recalculation mechanism will lower the difficulty
- Block production per actual unit of time will increase
- Attacker will also receive higher number of (absolute) rewards compared to honest play

Block Reward Zero Attack

- When the block reward becomes zero, incentives come only from tx fees
- The following deviation may be profitable:
    - When a miner receives two blocks of the same height, instead of choosing the first one it chooses the block that leaves the most transaction fees unclaimed
- A selfish miner can take advantage of this deviation by creating a fork with a block with less transaction fees than the block in the head of the public ledger
    - The attacker sacrifices part of their tx fees to attract others to join the fork

Carlsten, Miles, et al. "On the instability of bitcoin without the block reward." (2016)

Bribery Attack

- The attacker creates a fork and includes in the first block a transaction τ_0 that gives *bribe* money to miners who will adopt the fork and will extend this block
- The input of τ_0 is also transmitted in the public ledger and double spends the bribe money
- If the chain of the attacker does not become longer than the public ledger then the attacker does not lose the bribe money
  - In this case, miners who adopted this fork will have spent computational power without gaining anything

Bonneau, Joseph, et al. "Why buy when you can rent? bribery attacks on Bitcoin consensus." (2016).

Mining Pools

Mining pools

- Mining
  - gives a high reward per block
  - has a small probability of success
  - variance is high
- Miners typically collaborate in **mining pools**
  - **temporal discounting**: the tendency to disfavor rare or delayed rewards
  - “I prefer to get $1,600 per month than $80,000 after 4 years”
- If a miner in a pool finds a block, rewards are split among the pool members
- Splitting is *pro rata* according to the computational power contributed by each
- Miners outside of pools (very rare) are called **solo**
- Pools are maintained by a *trusted* **pool leader**

Mining inside a pool

- The pool maintains a different **internal** target for proof-of-work $T_{pool} > T_{bitcoin}$
- If a block satisfies $T_{bitcoin} < H(B) < T_{pool}$, it is called a **share**
- The miners of the pool mine as follows:
  - They include a coinbase tx with output the pool leader’s address
  - If $H(B) < T_{bitcoin}$, they broadcast the block to the **bitcoin network**
  - If $H(B) < T_{pool}$, they broadcast the share **inside the pool**
- Pool leader **verifies shares** by:
  - Checking that PoW is satisfied with T_{pool}
  - Checking that coinbase tx pays to the **pool’s address** and not some other address

Pool rewarding

- When a bitcoin block is created, each node in the pool is rewarded *proportionally* to the pool blocks they have recently generated
- Node participants pay a *participation fee* to the pool leader
- Pools are a trusted scheme:
  - Miners trust the pool leader, but the pool leader does not trust the miners
  - Miners don’t trust the other miners in the pool
- The pool leader **can steal** money, but they will be **detected**
- Why can’t a pool miner mine shares with the pool’s address, but blocks with his own address?
  - They don’t know if it will be a share or a block during mining!
  - After mining is completed, changing the address will invalidate the PoW.

Mining Pool Games

- To create a pool or join an existing one?
- Assuming cost of verification and pool maintenance is non-negligible:
    - Optimal solution is a single dictatorial pool
    - Reason: offset costs with the player that has the lowest service cost

Block withholding attack

- Consider there exists just two pools A, B with hashing power α and β resp.
- A segment of pool A (α') “infiltrates” pool B:
    - participates in pooled mining
    - receives rewards
    - does not share the solutions it finds
- Assuming no other deviations, over a period of n steps:
    - Pool A will produce $(\alpha-\alpha')*n$ blocks.
    - Pool B will produce $\beta*n$ blocks (same as before)
- In the same period of time, the shares of pool B will be distributed as follows:
    - Members of pool A will obtain $\alpha' / (\beta+\alpha')$ of such shares
    - Members of pool B will obtain $\beta / (\beta+\alpha')$ of such shares
- Pool A’s rewards in n steps are $(\alpha-\alpha')n + \beta\ n\ \alpha'/(\beta+\alpha')$ of total rewards $(\alpha - \alpha' + \beta)\ n$
    - In terms of *relative rewards*, this is better than $\alpha/(\alpha+\beta)$ (from honest behaviour)

Real-world Utility

Real utility ≠ Cryptocurrency utility

- The previous analyses measure utility in terms of **BTC received**
- **Real utility** depends also on the **exchange rate** (price) BTC/USD, BTC/GBP and other real-world (fiat) currencies
- Miners’ **costs** (e.g., energy bills) are in real-world money, not cryptocurrency
- **Market friction** (e.g., exchange fees) may further impact real utility
- **Detectable deviations** from the protocol may impact the price
    - If the protocol is perceived to be insecure or attackable → demand for BTC and trust in the network drops → BTC price drops → miner’s utility drops → counter-incentive to deviation
    - *however*, historical data show that the market typically does not respond in such manner (price does not drop significantly after an attack)

Will parties attack their own system?

- **Detectable deviations** from the protocol may impact the price
- If the protocol is perceived to be insecure or attackable
  - demand for token and trust in the network drops
  - → price drops
  - → miner’s (or stakeholder’s) utility drops
- Miners may be **dis-incentivized** to attack
  - This argument is even more present in PoS, where participation is via the tokens themselves
- However, **historical data** show that the **market does not respond** adequately
  - Prices don’t drop significantly after an attack
  - E.g., Ethereum Classic suffered two double-spending attacks in August 2020 and its price remained practically unaffected