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
- People would participate if financially rewarded (Homo Economicus)
  - If rational, people try to optimize their financial gain
- Users would pay participants (miners) to process their transactions
- Free market (everywhere)
- Security analysis: participants are either honest 😇 or adversarial 😈

The economics of consensus

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

Mining incentives

A miner is incentivized to mine in 2 ways:

1. Transaction Fees
2. (Fixed) Block Rewards

Mining fees

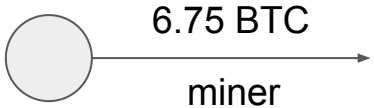
- [Recall] A transaction defines inputs and outputs
- Value conservation law:
  - *<sum of input values>* ≥ *<sum of output values>*
  - No amount value is created by a simple transaction
- *Transaction fees*: the remaining money from the conservation law of value
  - $tx\_fees = \Sigma_{i \in in(tx)}\ w(i) - \Sigma_{o \in out(tx)}\ w(o)$, 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 *newly-issued* coins
  - In 2021 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**
- 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**
- Halving mechanism is significantly better for early miners (adopters)

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

number of eras until reward is negligible

era duration in blocks

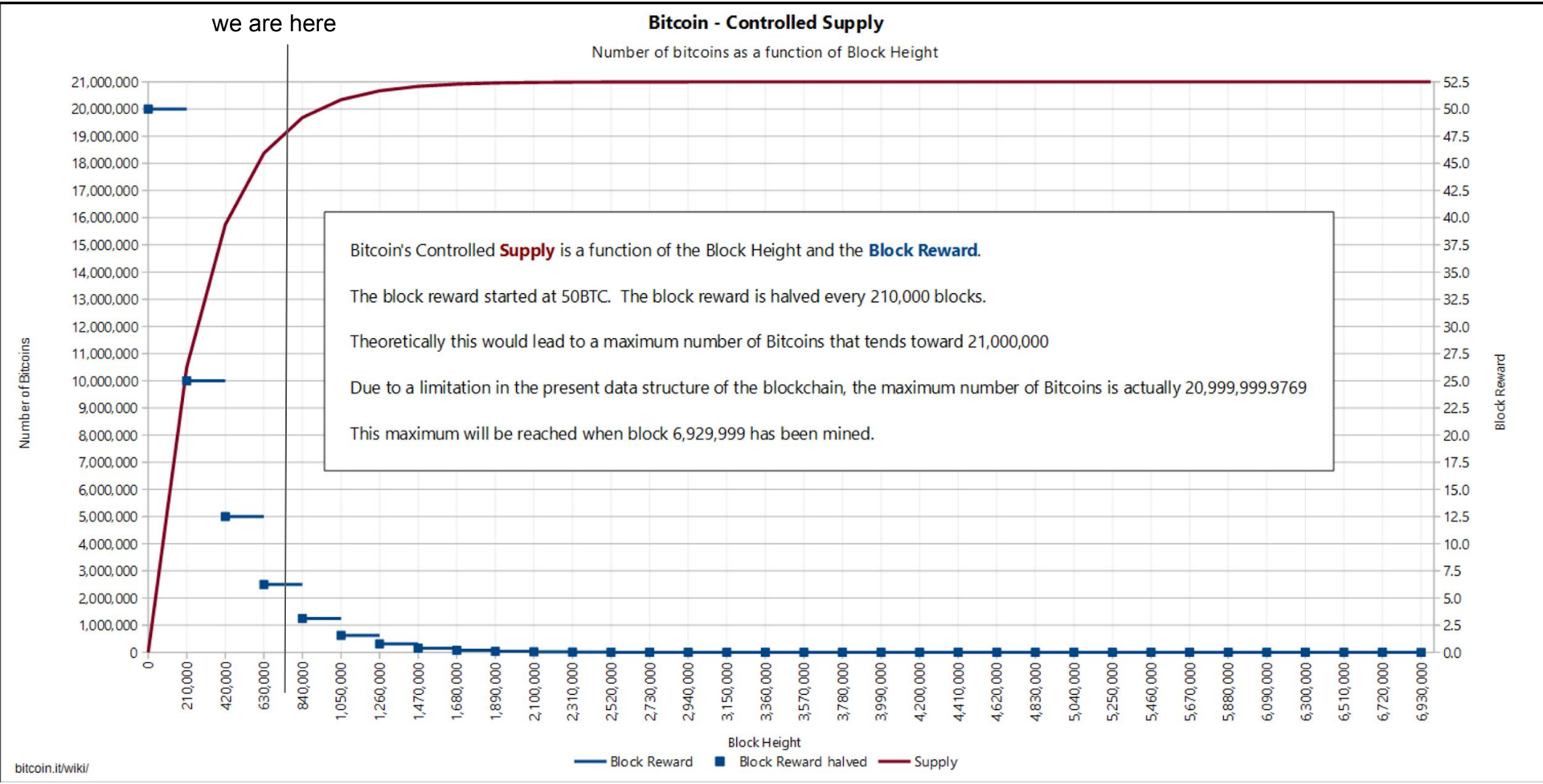
genesis block reward

$$\text{total\_btc\_supply} = \frac{\sum_{i=0}^{32} 210000 \left\lfloor \frac{50 \cdot 10^8}{2^i} \right\rfloor}{10^8}$$

satoshi / bitcoin

Date reached	Block	Reward Era	BTC/block	Year (estimate)	Start BTC	BTC Added	End BTC	BTC Increase	End BTC % of Limit
2009-01-03	0	1	50.00	2009	0	2625000	2625000	infinite	12.500%
2010-04-22	52500	1	50.00	2010	2625000	2625000	5250000	100.00%	25.000%
2011-01-28	105000	1	50.00	2011*	5250000	2625000	7875000	50.00%	37.500%
2011-12-14	157500	1	50.00	2012	7875000	2625000	10500000	33.33%	50.000%
2012-11-28	210000	2	25.00	2013	10500000	1312500	11812500	12.50%	56.250%
2013-10-09	262500	2	25.00	2014	11812500	1312500	13125000	11.11%	62.500%
2014-08-11	315000	2	25.00	2015	13125000	1312500	14437500	10.00%	68.750%
2015-07-29	367500	2	25.00	2016	14437500	1312500	15750000	9.09%	75.000%
2016-07-09	420000	3	12.50	2016	15750000	656250	16406250	4.17%	78.125%
2017-06-23	472500	3	12.50	2018	16406250	656250	17062500	4.00%	81.250%
2018-05-29	525000	3	12.50	2019	17062500	656250	17718750	3.85%	84.375%
2019-05-24	577500	3	12.50	2020	17718750	656250	18375000	3.70%	87.500%
2020-05-11	630000	4	6.25	2021	18375000	328125	18703125	1.79%	89.063%
	682500	4	6.25	2022	18703125	328125	19031250	1.75%	90.625%
	735000	4	6.25	2023	19031250	328125	19359375	1.72%	92.188%
	787500	4	6.25	2024	19359375	328125	19687500	1.69%	93.750%

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


we are here
Bitcoin - Controlled Supply
Number of bitcoins as a function of Block Height
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.
Number of Bitcoins
Block Reward
Block Height
Block Reward
Block Reward halved
Supply
bitcoin.it/wiki/

Money distribution in Bitcoin

- Halving mechanism is significantly better for early miners
- 50% of *all* bitcoins that will ever be produced were created in first 3 years
- Extremely centralized money distribution
  - In 2013, 2,300 addresses (i.e., a couple thousand people *at most*, in reality much less) controlled 50% of all supply
  - In 2021, 16,190 addresses control 63% of all bitcoins

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

Is it worth it to mine? Probably not...

- Today:
  - CPU Intel i7-8700K:
    - Initial hardware cost: *$360*
    - Profit: *-$0.1 / day*
  - GPU NVIDIA GTX 1050 Ti:
    - Initial hardware cost: *$160*
    - Profit: *$0.29 / day*
    - 1st year profit: *-$54*
  - Specialized hardware AntMiner S17+:
    - Initial hardware cost: *$7,000*
    - Profit: *$16.89 / day*
    - 1st year profit: *-$835*
- 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: $66,000
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 is dominant strategy:
  - a party will fare best by following the protocol
- Protocol is Nash equilibrium:
  - if all parties follow the protocol, nobody can do better by deviating from it

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

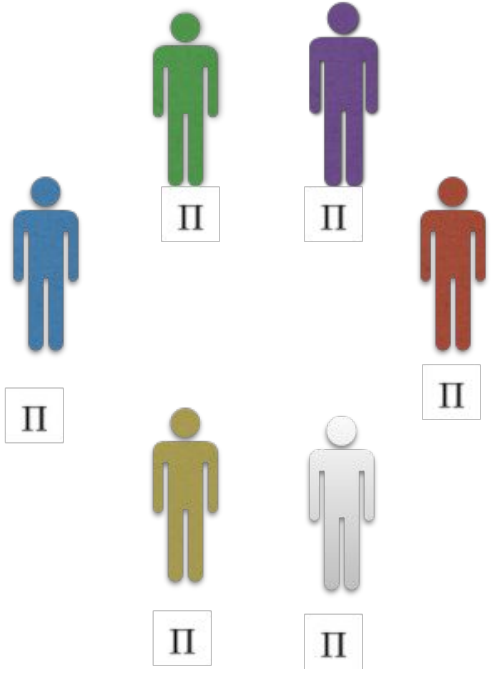
- Stealing is dominant strategy
  - For player A: 100 > 50 (if B splits), 1 > 0 (if B steals)
  - Same for B
- Steal/Steal is *sub-optimal* strategy
  - Split/Split yields higher rewards for both
- See also: prisoner's dilemma

Protocol is Nash Equilibrium, I

- 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

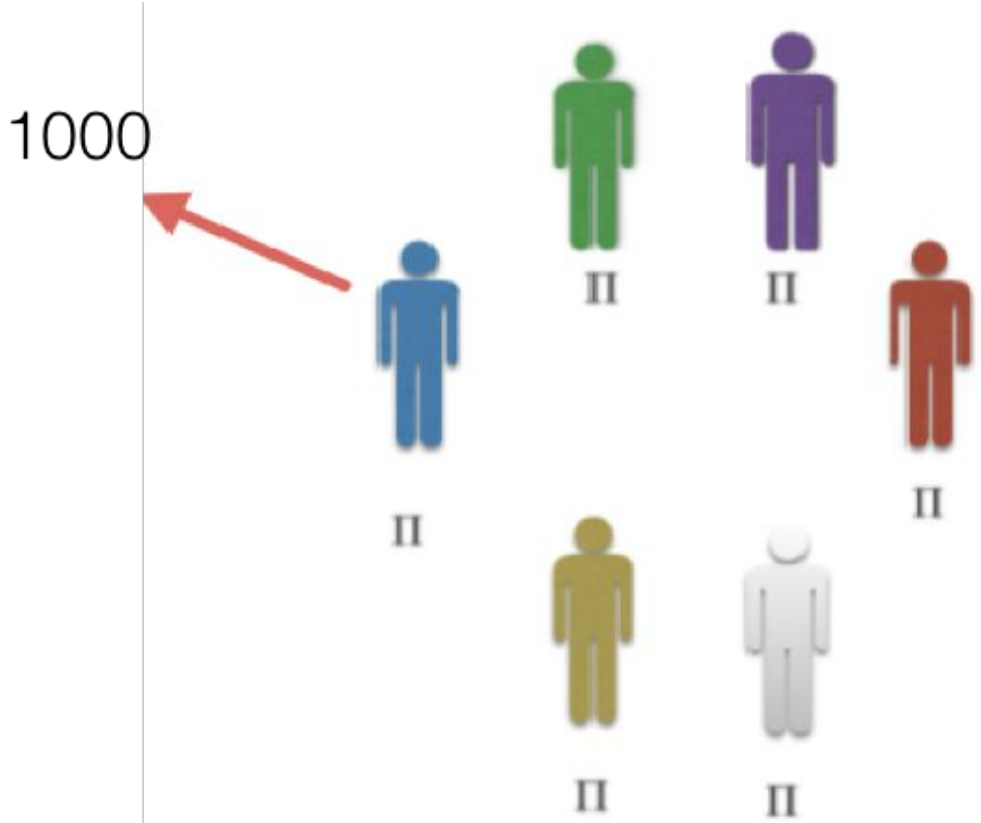
Protocol is Nash Equilibrium, II

All the participants are rational: they want to maximize their utility.

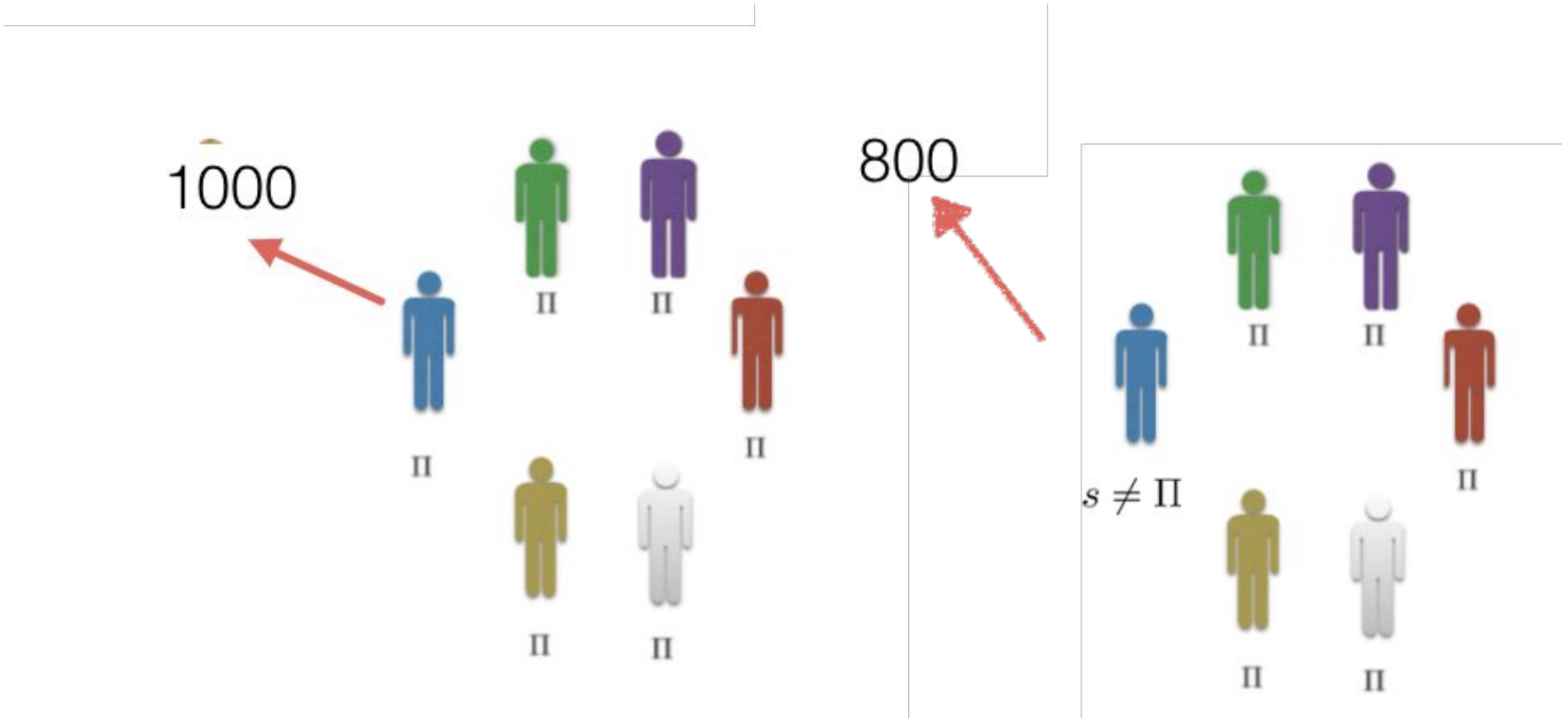


Protocol Π is Nash equilibrium

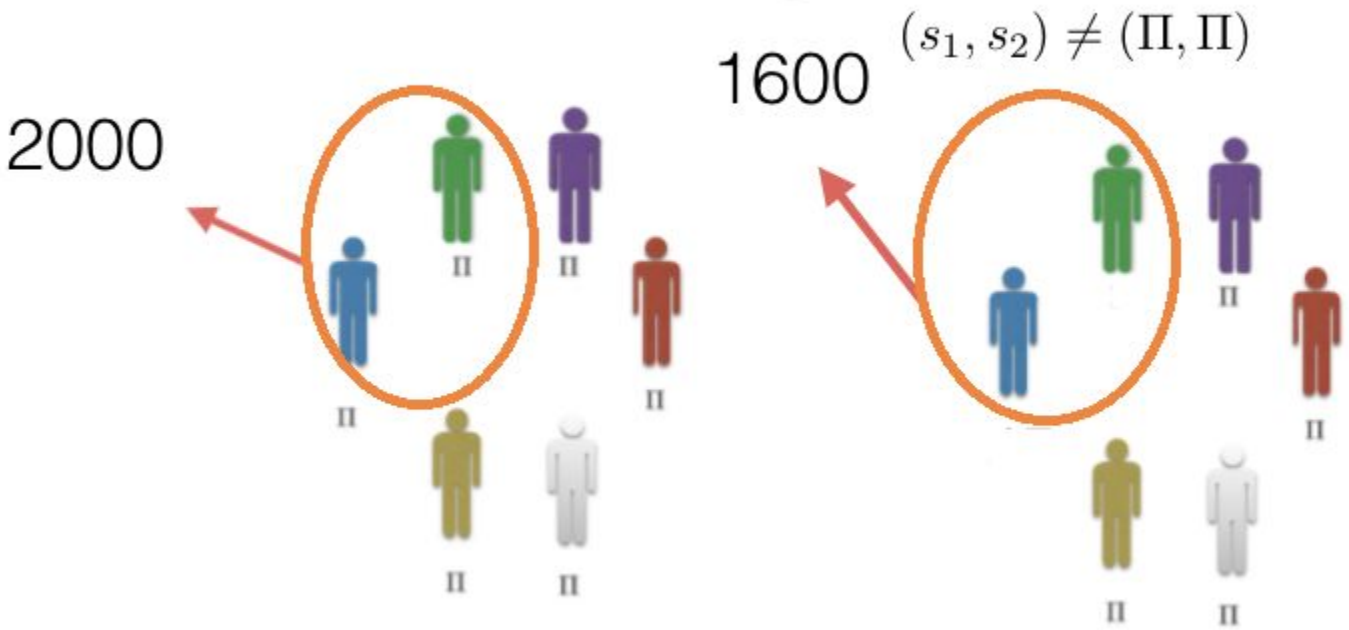
Protocol is Nash Equilibrium, III



Protocol is Nash Equilibrium, IV



Generalisation to Coalitions

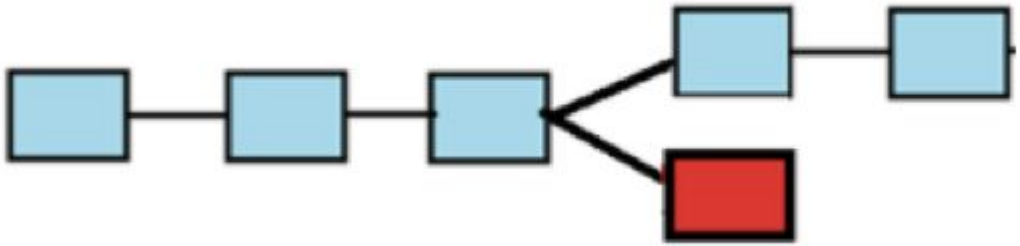


Is Bitcoin a Nash Equilibrium ?

- 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:
  a. the algorithms followed by all the participants
  b. the outcome of all the randomness used by participants
  c. 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

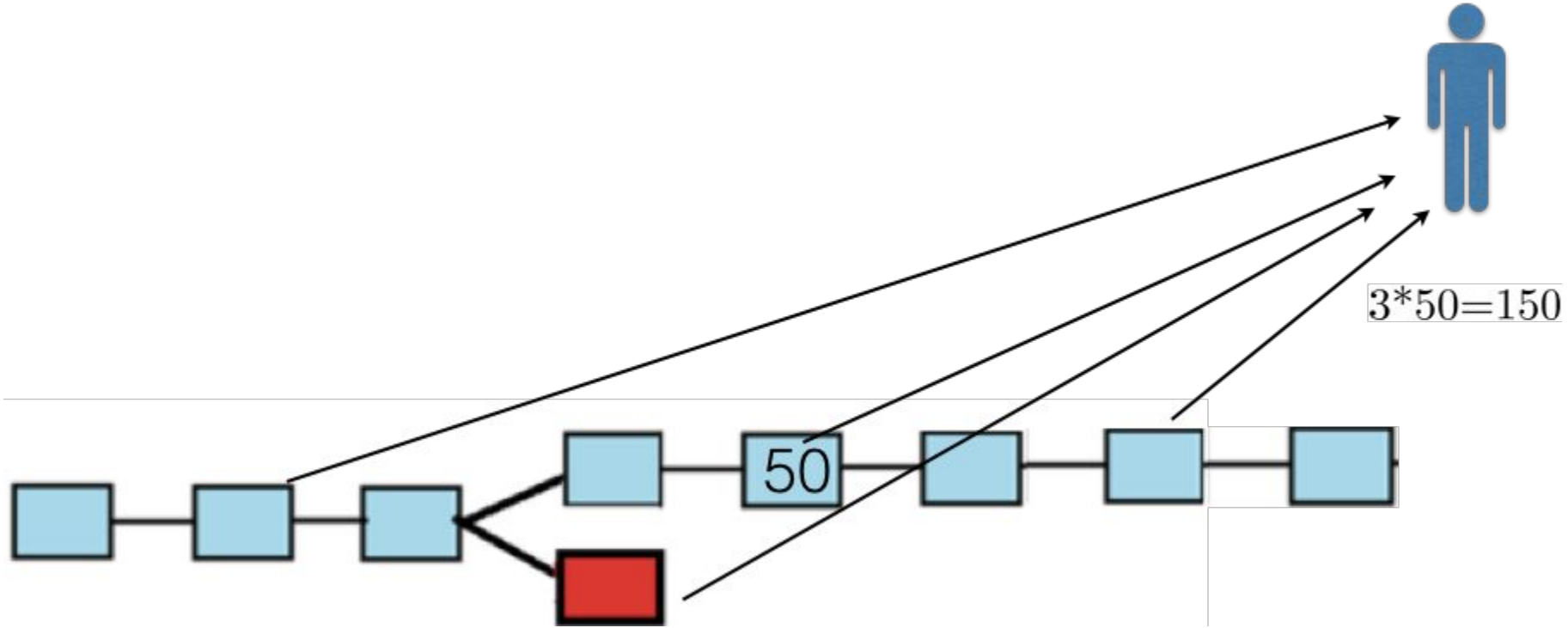


Absolute Rewards, II

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

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

Absolute Rewards, III

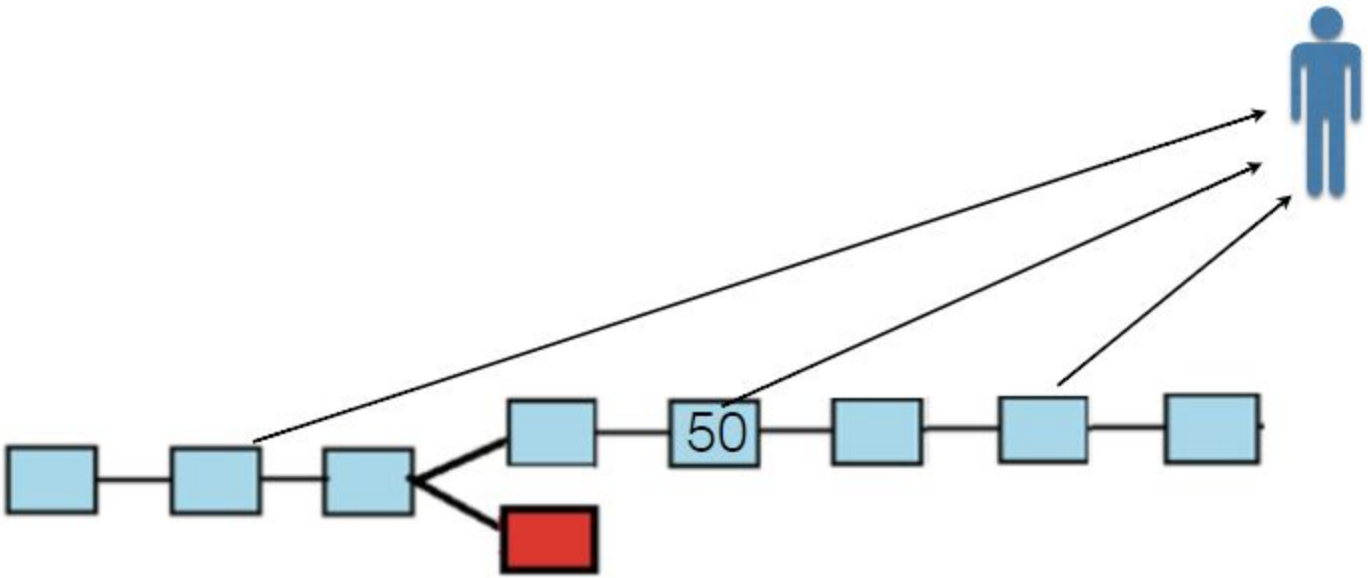


Relative Rewards, I

- **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

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

Relative Rewards, II



$$\frac{3 * 50}{8 * 50}$$

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 modeling of the Bitcoin protocol is a Nash equilibrium w.r.t. absolute rewards
  - **utility** is equal to the **expected value 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. relative rewards
  - *selfish mining* attack

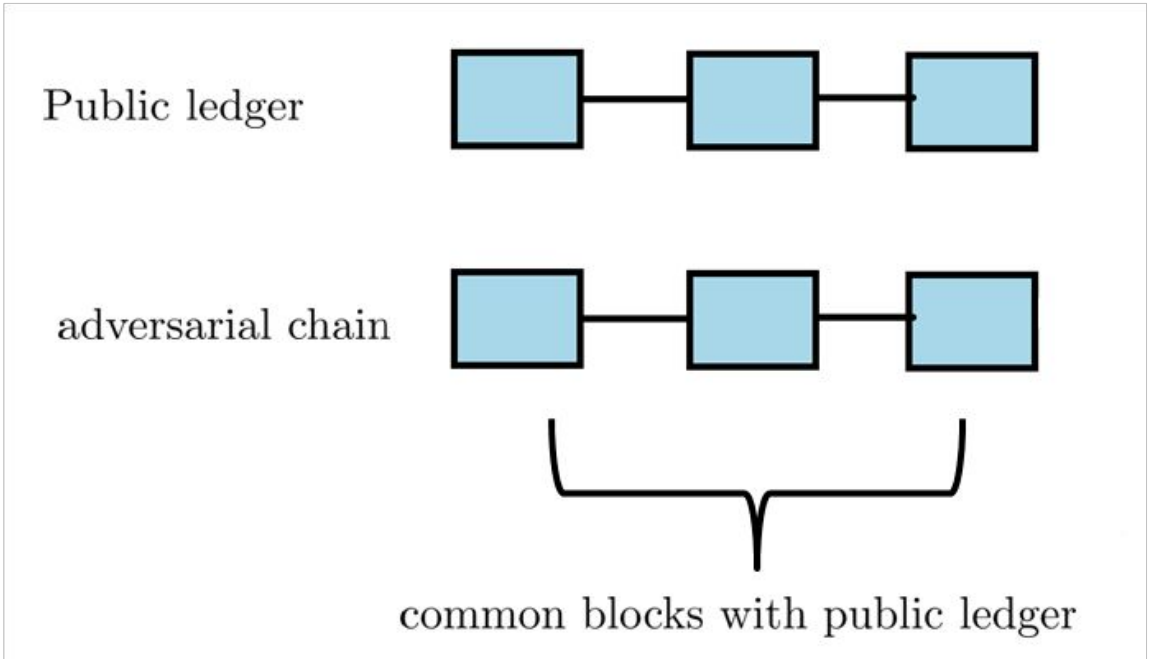
Kroll et al. in "The economics of Bitcoin mining, or Bitcoin in the presence of adversaries" (2013)
Eyal, Ittay, and Emin Gün Sirer. "Majority is not enough: Bitcoin mining is vulnerable."(2014)

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

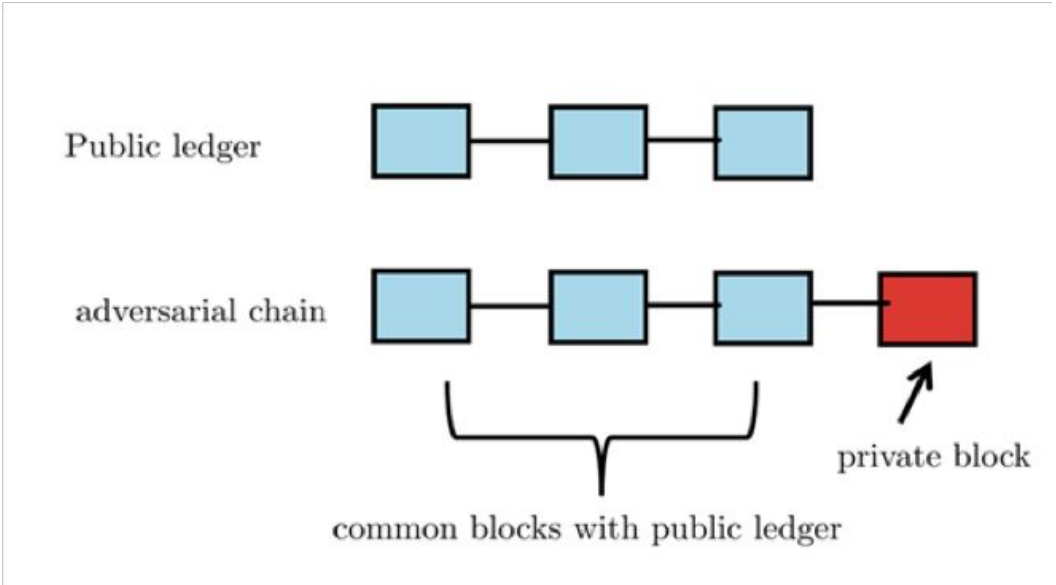
1)The attacker adopts the longest chain and tries to extend it.



Selfish Mining, case 2a

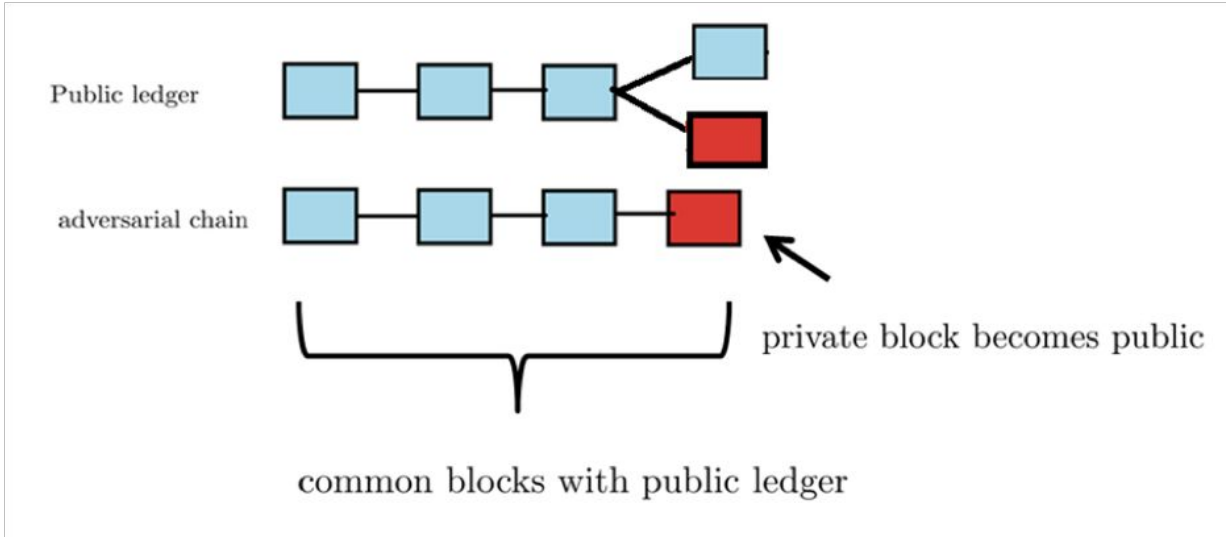
We have the following two cases : 2a or 2b

2a)The attacker is first to produce a block.



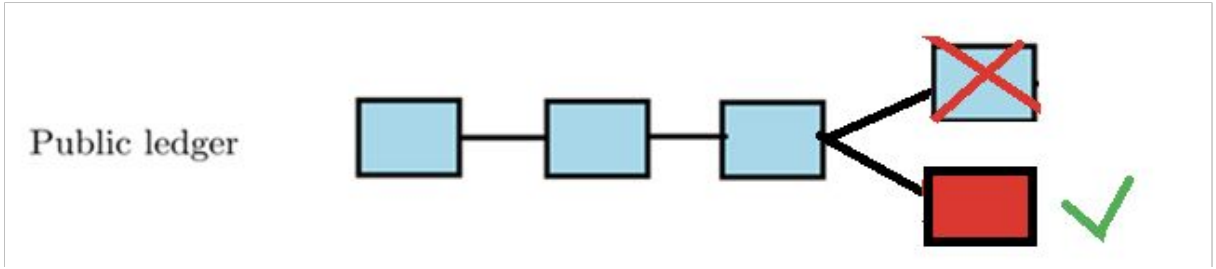
Selfish Mining, case 2a

The attacker withholds its block(s) private until the public chain comes to be (i) one block behind the private one, (ii) equal in length to the attackers. (choice can depend on “network dominance” of the adversary)



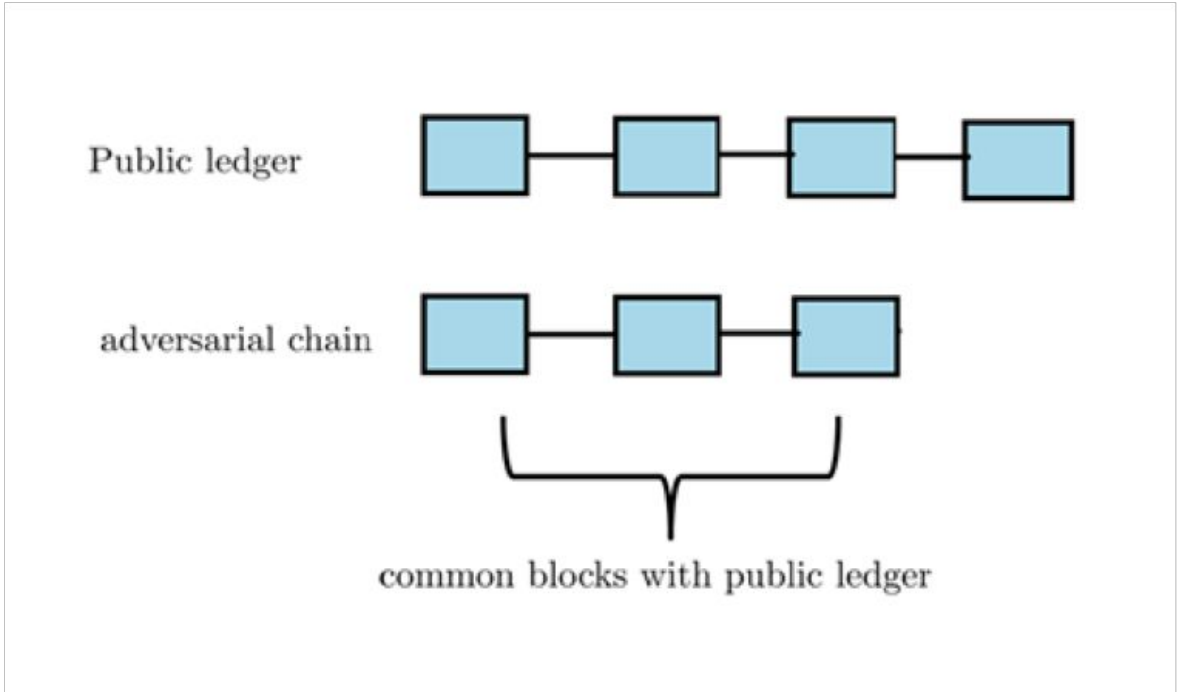
Selfish Mining, case 2a

If the other parties choose to extend the adversarial chain then the adversary has managed to censor legitimate blocks from the public ledger.



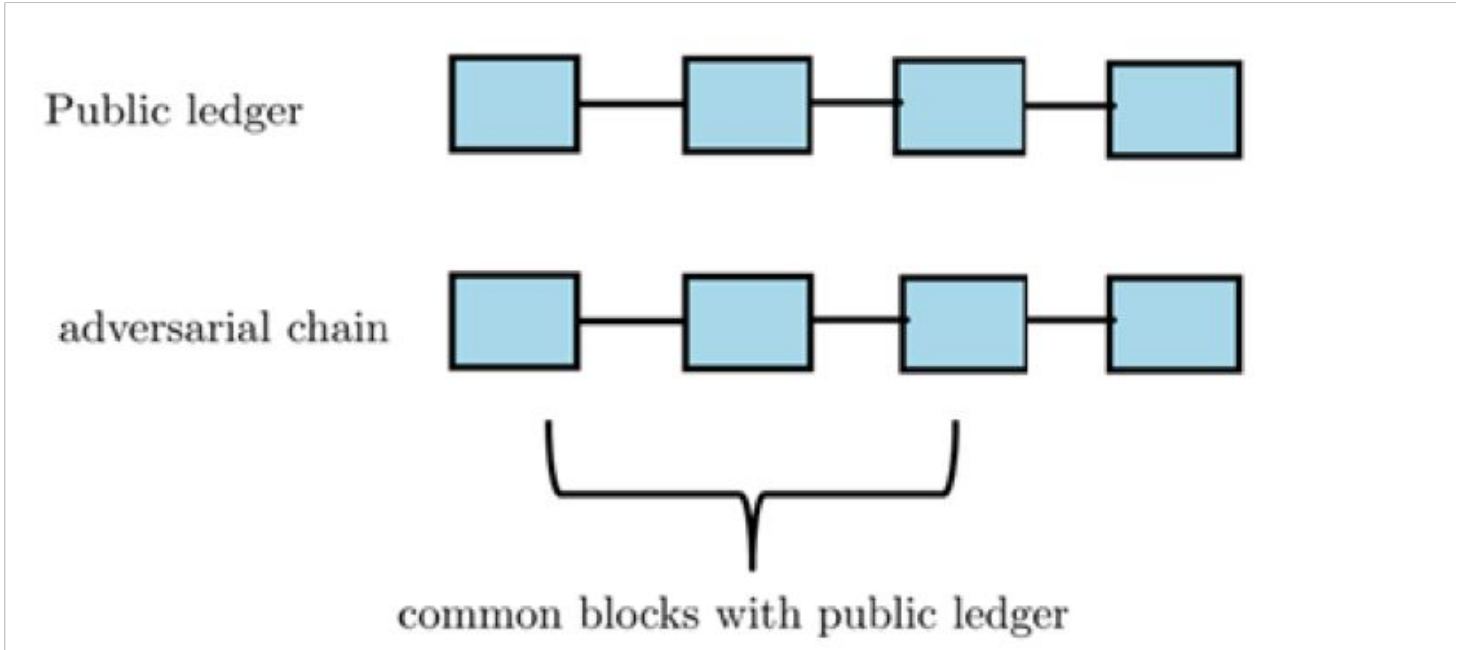
Selfish Mining, case 2b

2b)The attacker does not manage to produce first a block.



Selfish Mining, case 2b

The attacker in this case adopts the public ledger.



Selfish Mining

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

Selfish Mining, Analysis, I

- The computational power of the attacker contributes only towards censoring blocks and not towards extending the public ledger.
- So when it implements the attack, the total (expected) number of blocks in the public ledger is smaller compared to the total number of blocks in the case where it follows the protocol.
- If the attacker does not manage to deliver its block first, it loses the rewards from this block.

Selfish Mining, Analysis, II

- Consider a process operating in “block rounds"
- Attacker has probability α to produce the next block (and always wins the network race vs. public blocks):
  - Honest play:
    i. *n* rounds -> *n* blocks
    ii. attacker owns α*n blocks in expectation
    iii. Utility: (Relative Rewards) = α, (Absolute Rewards) = α**n*
  - Selfish play:
    i. *n* rounds result to (1-α)**n* blocks
    ii. attacker owns α*n* of those blocks
    iii. 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

- Difficulty adjustment setting
  - 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 the following deviation may arise:
  - When a miner receives two blocks of the same height instead of choosing the first one, it has incentives to choose the block that leaves the most transaction fees unclaimed
  - A selfish miner can take advantage of this behaviour and create a fork with a block including fewer transaction fees compared to the transaction fees in the head of the public ledger

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
  - gives a high reward per block
  - has a small probability of success
  - variance is high
- Miners typically collaborate in **mining pools**
  - temporal discounting, i.e., 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* centralized **pool owner**

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 (owner'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 share verification

Pool owner verifies shares:

- Check that PoW is achieved with $\mathbf{T_{pool}}$
- Check that coinbase tx pays to the **pool 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 (compared to the total number of pool blocks generated)
- Node participants pay a *participation fee* to the owner of the pool
- Pools are a trusted scheme:
  - Miners trust the pool owner, but the pool owner does not trust the miners
  - Miners don’t trust the other miners in the pool
- The pool owner 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 (α-α')**n* blocks.
  - Pool B will produce β**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 α' / (β+α') of such shares
  - Members of pool B will obtain β / (β+α') of such shares
- Pool A's rewards in *n* steps are (α-α')*n* + β *n* α'/(β+α') of total rewards

  (α - α' + β) *n*
  - In terms of *relative rewards*, this is better than α/(α+β) (from honest behaviour)

Real utility ≠ Cryptocurrency utility

- The previous analyses measure utility in terms of *BTC* received in absolute or relative terms
- *Real world utility* depends also on the exchange rate BTC/USD, BTC/GBP and other real-world (fiat) currencies
- Detectable deviations from the protocol *may* impact the exchange rate, generating a strong counter incentive to deviation
    - e.g., if the protocol is perceived to be insecure or attackable, demand for BTC will drop, leading to an unfavourable exchange rate
    - *however*, historical data show that the market typically does not respond in such manner (price does not drop significantly after an attack)