Blockchains and Distributed Ledgers
Lecture 05

Aggelos Kiayias & Aikaterini-Panagiota Stouka

THE UNIVERSITY *of* EDINBURGH

Lecture 05

- Bitcoin's Reward Mechanism.
- Incentive Compatibility of Bitcoin.
- Selfish Mining Attack.
- Fairness.
- Bribery Attack.
- Transactions' Verification and Broadcast : Incentives
- BWH attack, FAW Attack

Incentives for the Ledger

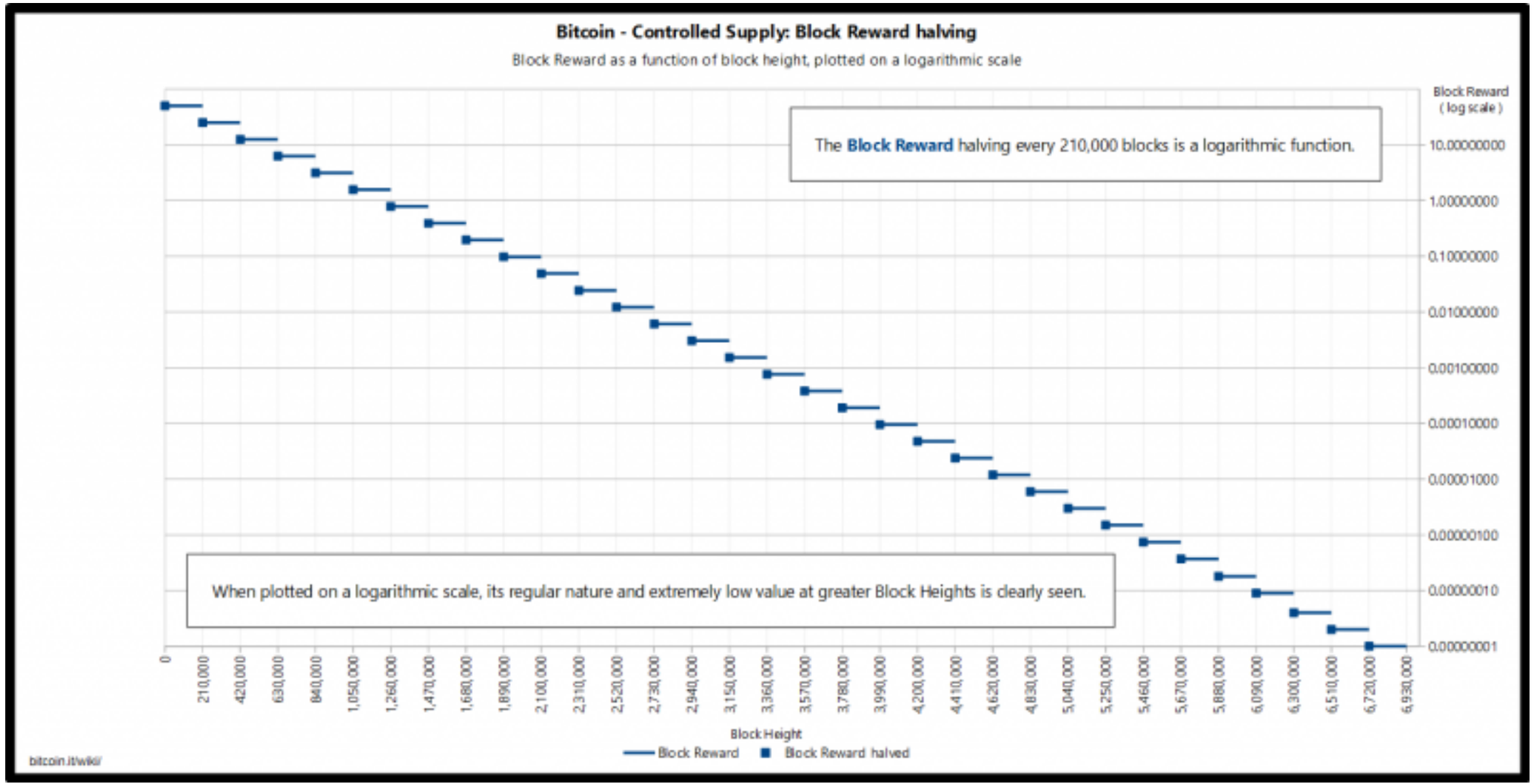
- So far : honest majority in hashing power implies robustness of transaction ledger.

- .. But why should miners follow the protocol at all?

  - Game theoretic analysis: the miners are rational participants that would like to maximize their utility.

Bitcoin's Reward Mechanism I

- Recall : for producing a block, a miner
  - collects a flat fee (that is decreasing over time)



Bitcoin's Reward Mechanism II

- This flat fee reward is mined the time the block is produced.

- 50% reduction of the flat fee every 210,000 blocks or every approximately four years.

- Total number of Bitcoin will not exceed 21 million.

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

https://bitcoin.org/en/developer-guide
https://en.bitcoin.it/wiki/Controlled_supply

Bitcoin's Reward Mechanism, III

- A miner also collects all the *transaction fees* of all the transactions included in the block that it produced.

- Transaction fee of a transaction is the amount that remains if we subtract the value of the output from the value of the input.

- Transaction fees at least 1,000 Satoshi equal to 0.00001 BTC.

https://bitcoin.org/en/developer-guide
https://en.bitcoin.it/wiki/Controlled_supply

Miner's payment

- Each block includes a coinbase transaction that sends the flat fee and the transaction fees to the public key of the miner.

- So even if two blocks include the same transactions, the coinbase transaction is different -> necessarily the blocks have different hash values.

Is the reward mechanism “incentive compatible”?

- Incentive compatibility
  - Protocol is dominant strategy: a party will fare best by following the protocol.
  - Protocol is Nash equilibrium: if all parties follow the protocol, you cannot do better by deviating.

Example of Dominant Strategy

Participants want to minimize years in prison

Prisoner A \ Prisoner B	**Prisoner B stays silent (*cooperates*)**	**Prisoner B betrays (*defects*)**
Prisoner A stays silent (*cooperates*)	Each serves 1 year	Prisoner A: 3 years Prisoner B: goes free
Prisoner A betrays (*defects*)	Prisoner A: goes free Prisoner B: 3 years	Each serves 2 years

https://en.wikipedia.org/wiki/Prisoner%27s_dilemma

Defection is a dominant strategy

Definition: A Protocol is Nash Equilibrium I

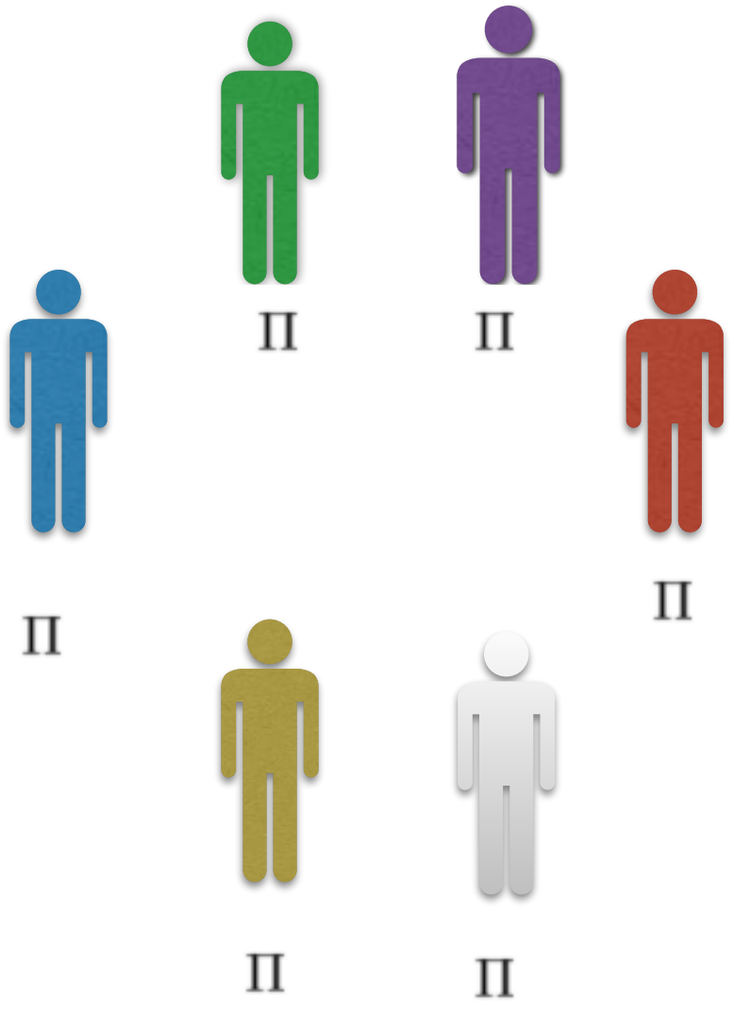
- We will describe when a deterministic protocol is Nash equilibrium.

- All the participants are rational and want to maximize the money that they will earn at the end of the protocol.

- Utility of a participant is a function that takes as input the strategies of all the participants and has as output the money that this participant will earn at the end of the protocol.

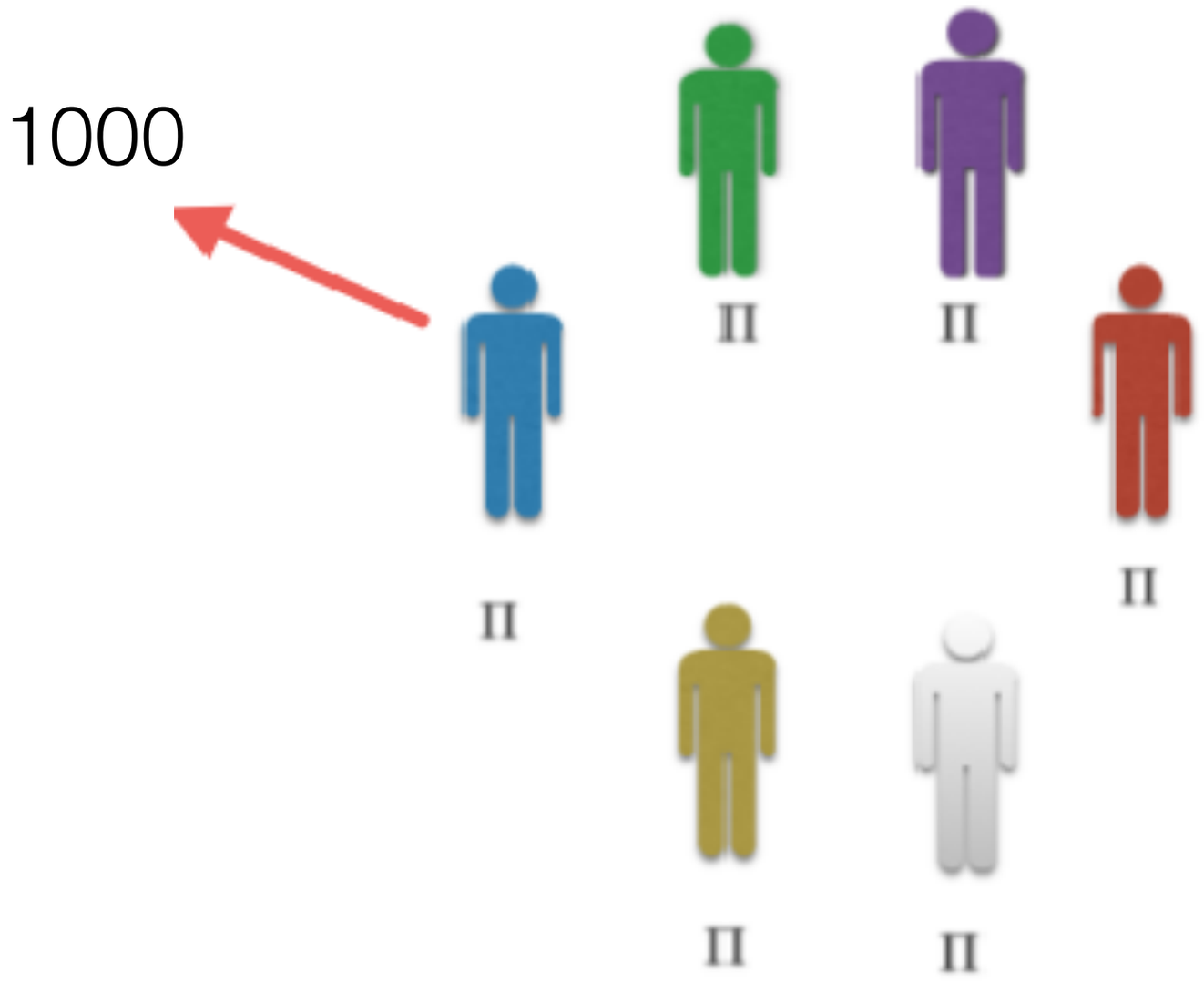
Definition: A Protocol is Nash Equilibrium II

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

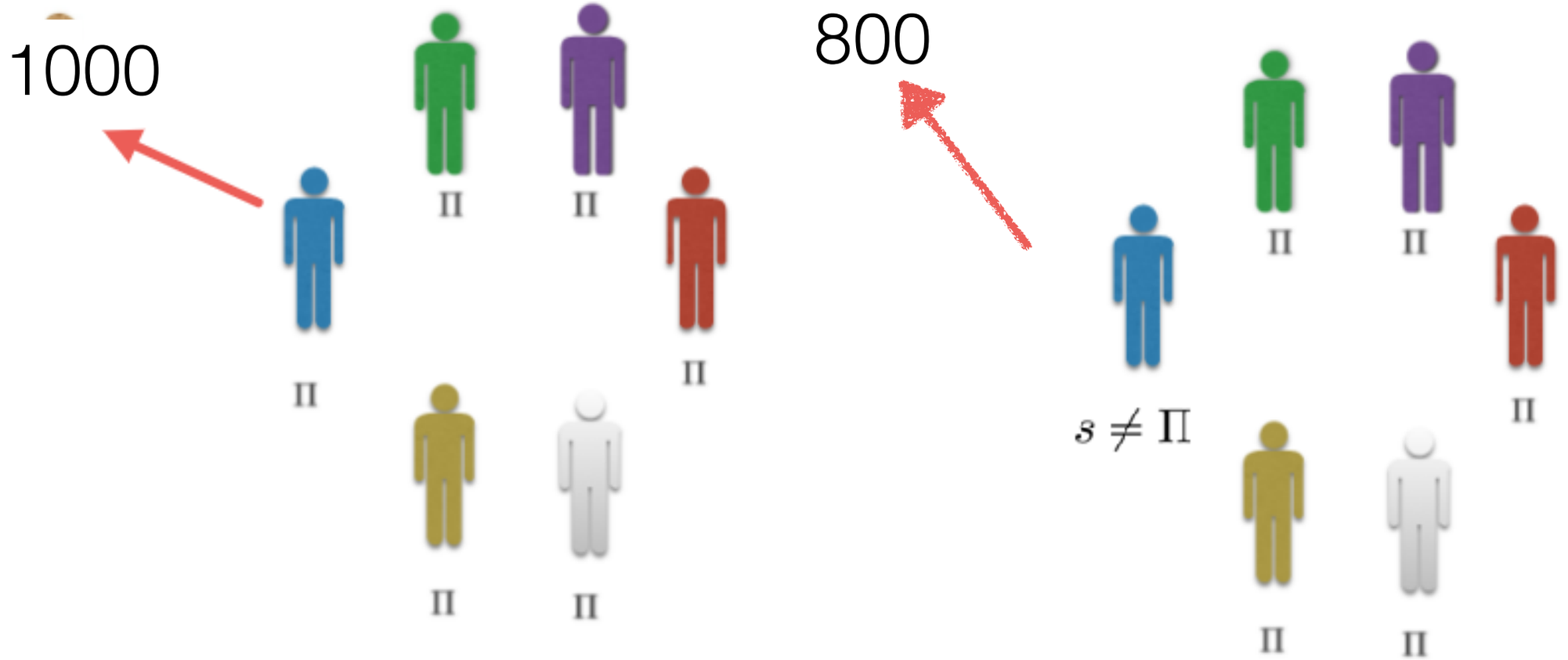


Protocol Π is Nash equilibrium

Definition: A Protocol is Nash Equilibrium III



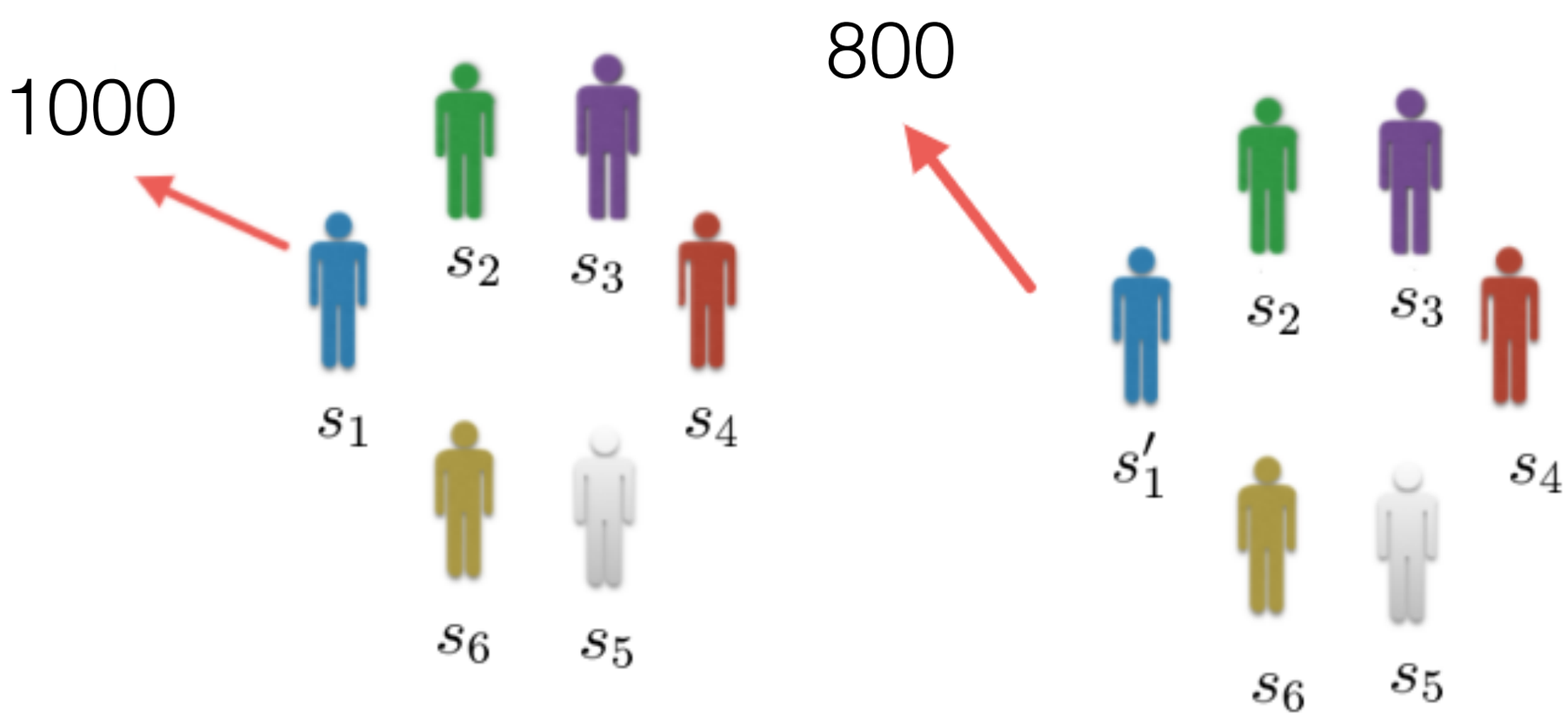
Definition: A Protocol is Nash Equilibrium IV



The same image holds for all the participants.

Definition : A Strategy Profile is Nash Equilibrium

A strategy profile $(s_1, s_2, s_3, s_4, s_5, s_6)$ that is Nash equilibrium.



The same image holds for all the participants.

Example of Nash Equilibrium

Participants want to minimize years in prison

Prisoner A \ Prisoner B	**Prisoner B stays silent (*cooperates*)**	**Prisoner B betrays (*defects*)**
Prisoner A stays silent (*cooperates*)	Each serves 1 year	Prisoner A: 3 years Prisoner B: goes free
Prisoner A betrays (*defects*)	Prisoner A: goes free Prisoner B: 3 years	Each serves 2 years

https://en.wikipedia.org/wiki/Prisoner%27s_dilemma

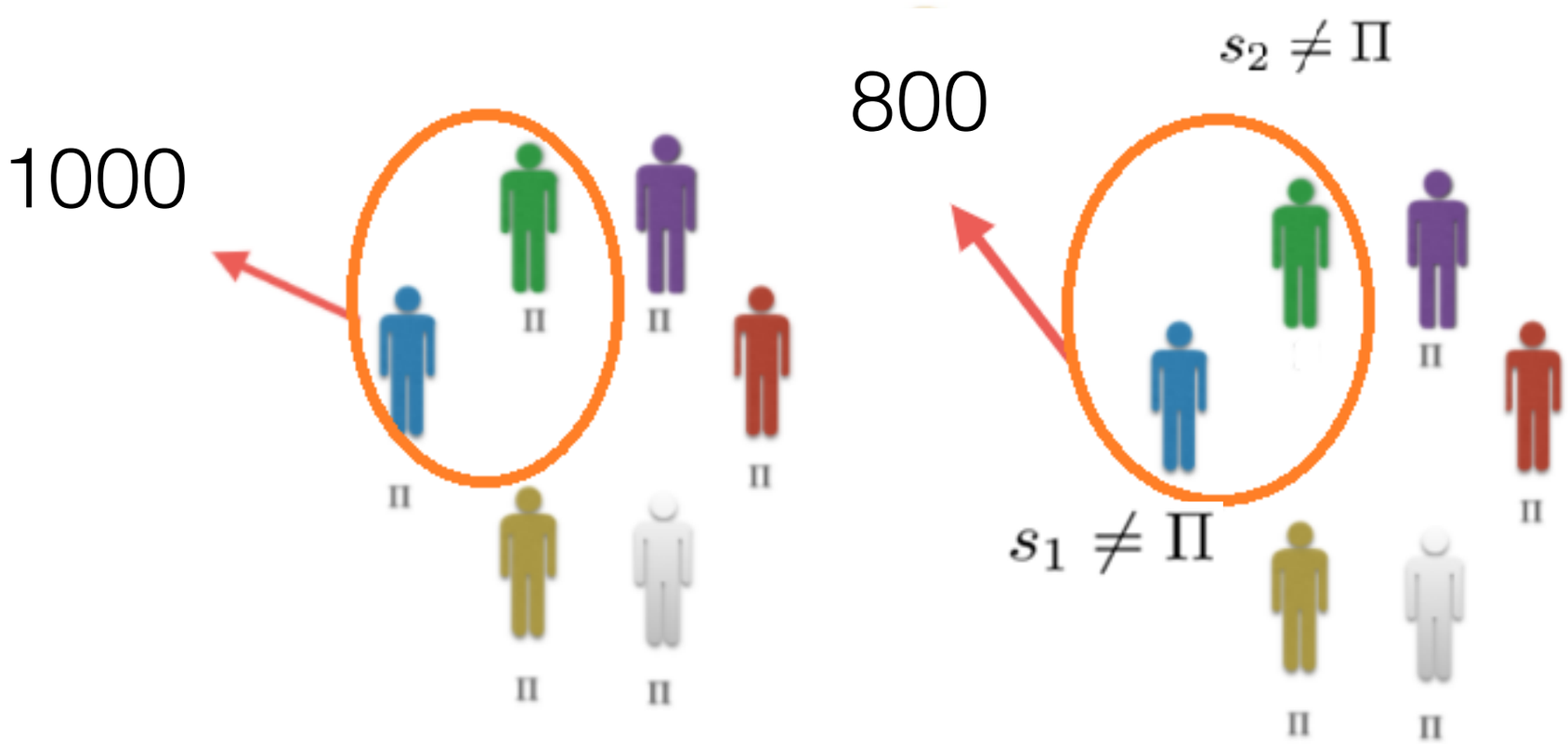
Mutual defection is Nash Equilibrium.

Participants Form Coalitions

- What happens when two or more participants can collaborate which means that they form a *coalition* and share the money that they receive at the end of the protocol?

Protocol Π is Nash equilibrium

in the case of coalitions of at most two participants



The same image holds for all the coalitions of at most two participants.

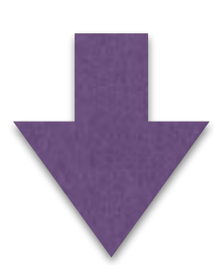
Utility I

- Utility of a coalition in the previous example was the money that it will earn at the end of the protocol.

- However utility in a protocol could be anything a coalition could maximize.

Utility II

- Coalitions want to earn more money compared to other participants.



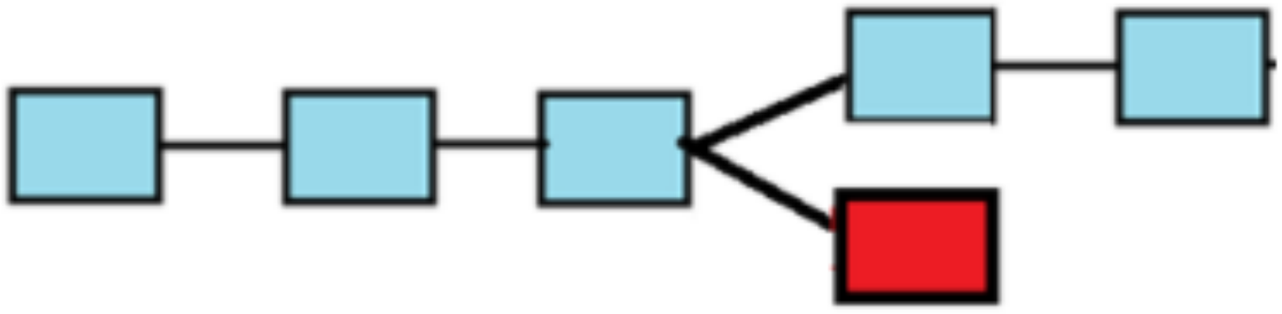
- utility of a coalition : the money that it earns divided by the total money that all the participants receive at the end of the protocol.

Types of Utility in Bitcoin

- What could the utility be in Bitcoin?
- How could utility be defined in a probabilistic protocol?
- Remember: Bitcoin is a probabilistic protocol.

Absolute Rewards I

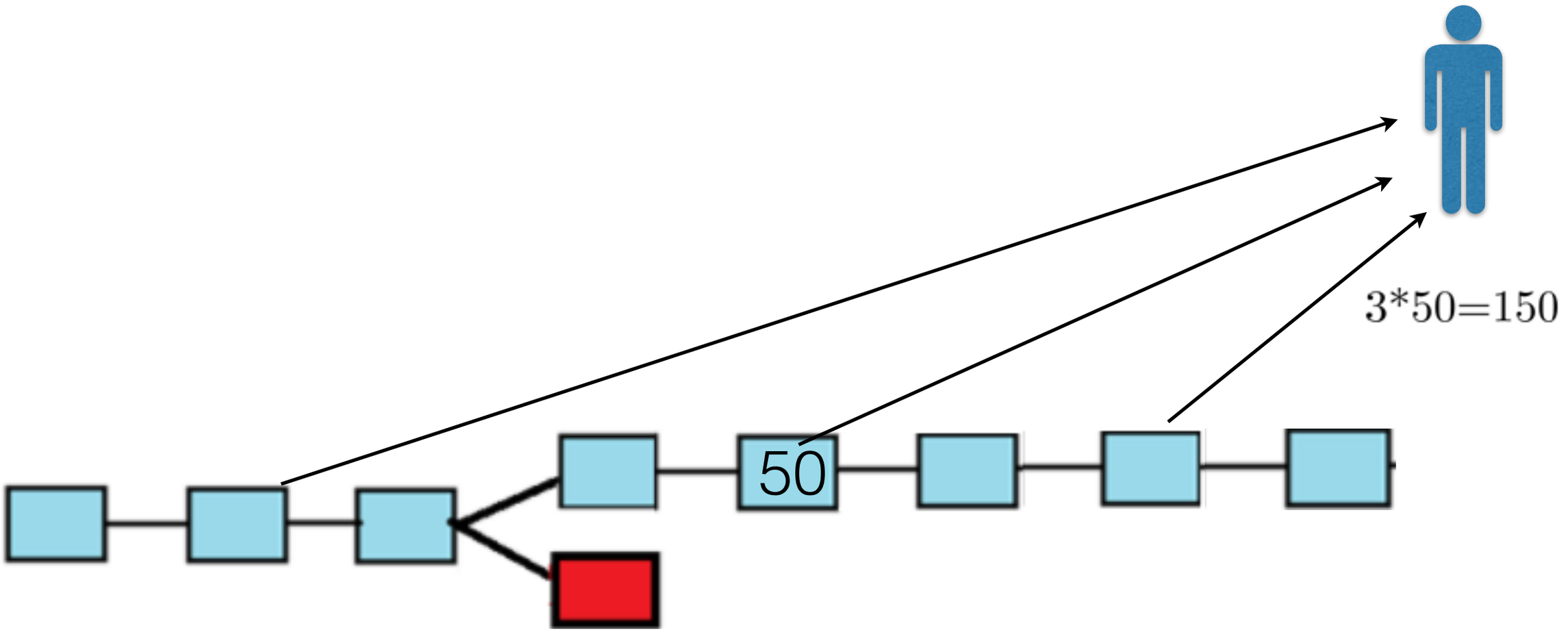
- If we specify the algorithms of all the participants and the outcome of the dice in a finite execution of the Bitcoin protocol then we have a unique outcome.



Absolute Rewards II

- Each block of the longest chain gives a flat reward to its producer. For instance, this reward can be 50 BTC.
- The utility of a coalition is equivalent to the absolute rewards when it wants to maximize the actual number of BTC that it receives at the end of the execution.

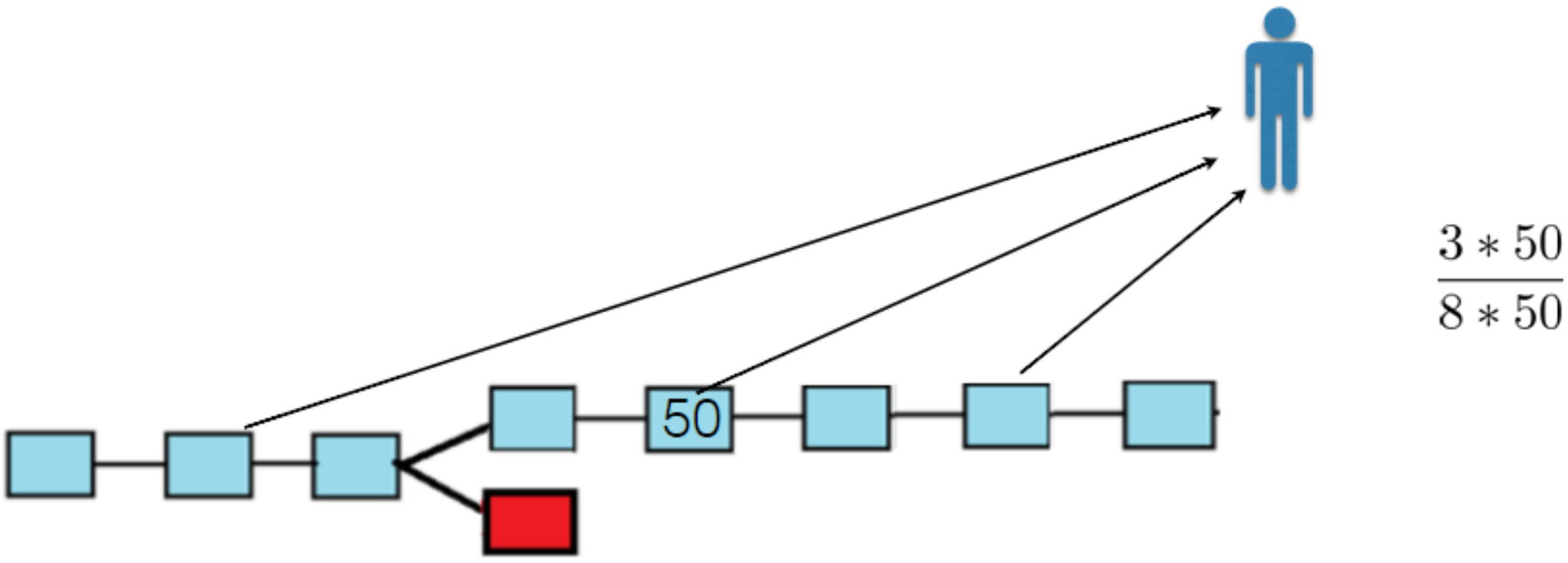
Absolute Rewards III



Relative Rewards I

- The utility of a coalition in Bitcoin is equivalent to relative rewards when it wants to maximize the amount of BTC that it earns divided by the total amount of BTC that all the participants receive at the end of the execution.

Relative Rewards II



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

Utility in Probabilistic Protocols

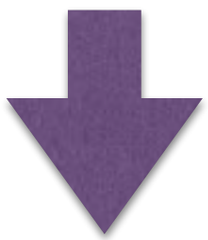
- Given the strategies of all the participants the outcome of the Bitcoin execution is a random variable, as Bitcoin is probabilistic.

- So absolute rewards and relative rewards of a coalition are also random variables.

- As types of utilities we will determine the expected value of the absolute rewards and the relative rewards respectively.

Bitcoin Incentives, I

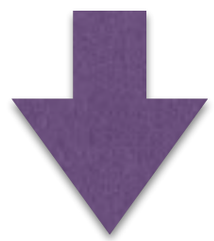
- Kroll et al. in ``The economics of Bitcoin mining, or Bitcoin in the presence of adversaries" (2013) show that a certain modeling of the Bitcoin protocol is a Nash equilibrium.



- Utility is equivalent to the expected value of absolute rewards.

Bitcoin Incentives, II

- Eyal and Sirer in `` Majority is not enough: Bitcoin mining is vulnerable" (2014) show that Bitcoin is susceptible to a type of attack called selfish mining and the protocol is not a Nash equilibrium.



- Utility equivalent to the expected value of relative rewards.

Bitcoin Incentives, III

- Kiayias et al. in `` Blockchain mining games" (2016) show that there are thresholds of hashing power where certain games that abstract Bitcoin have following the protocol as a Nash equilibrium.

- Utility equivalent to expected value of relative rewards

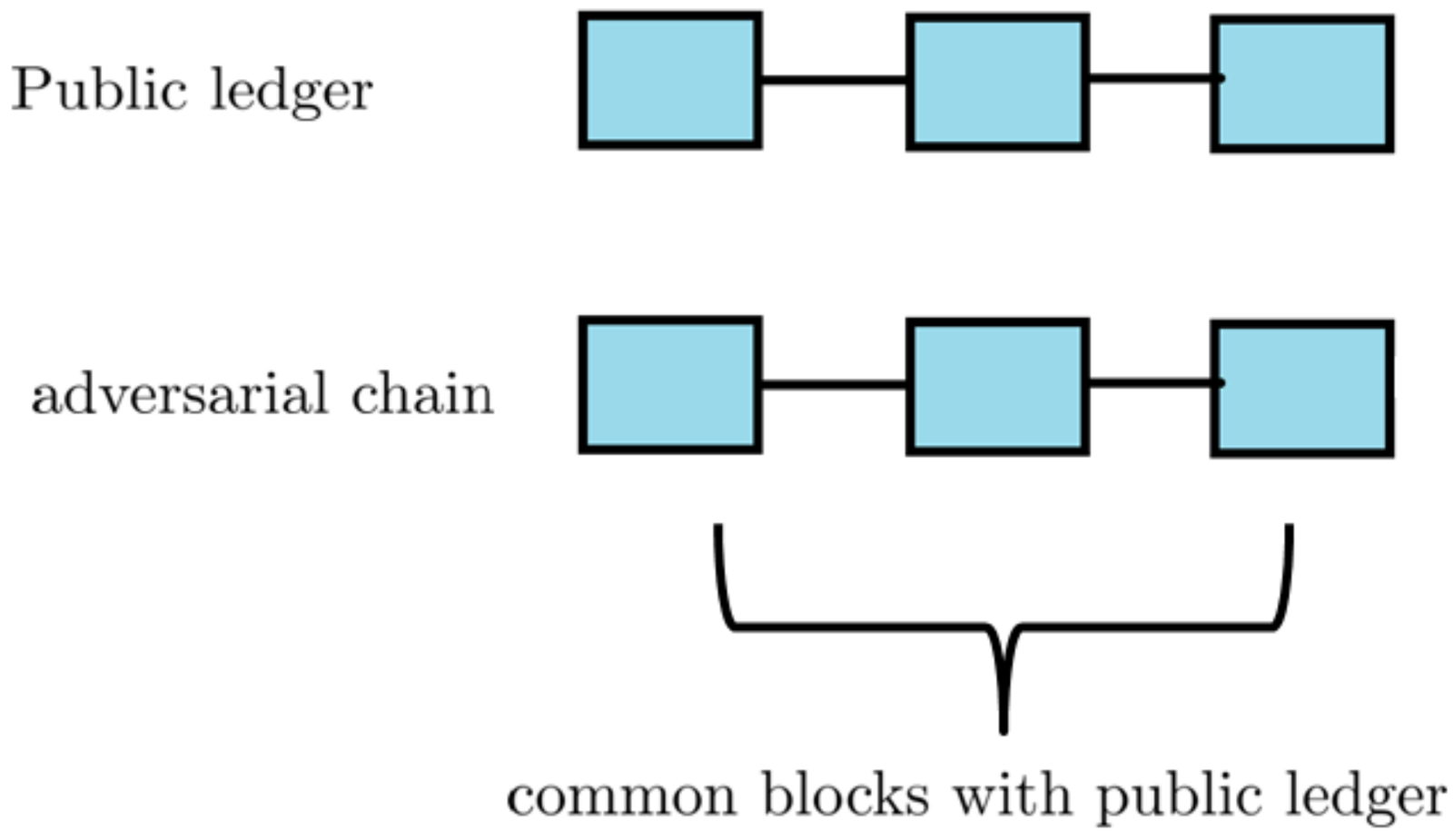
Selfish Mining, I

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

Eyal, Ittay, and Emin Gün Sirer. "Majority is not enough: Bitcoin mining is vulnerable."(2014)

Selfish Mining, II

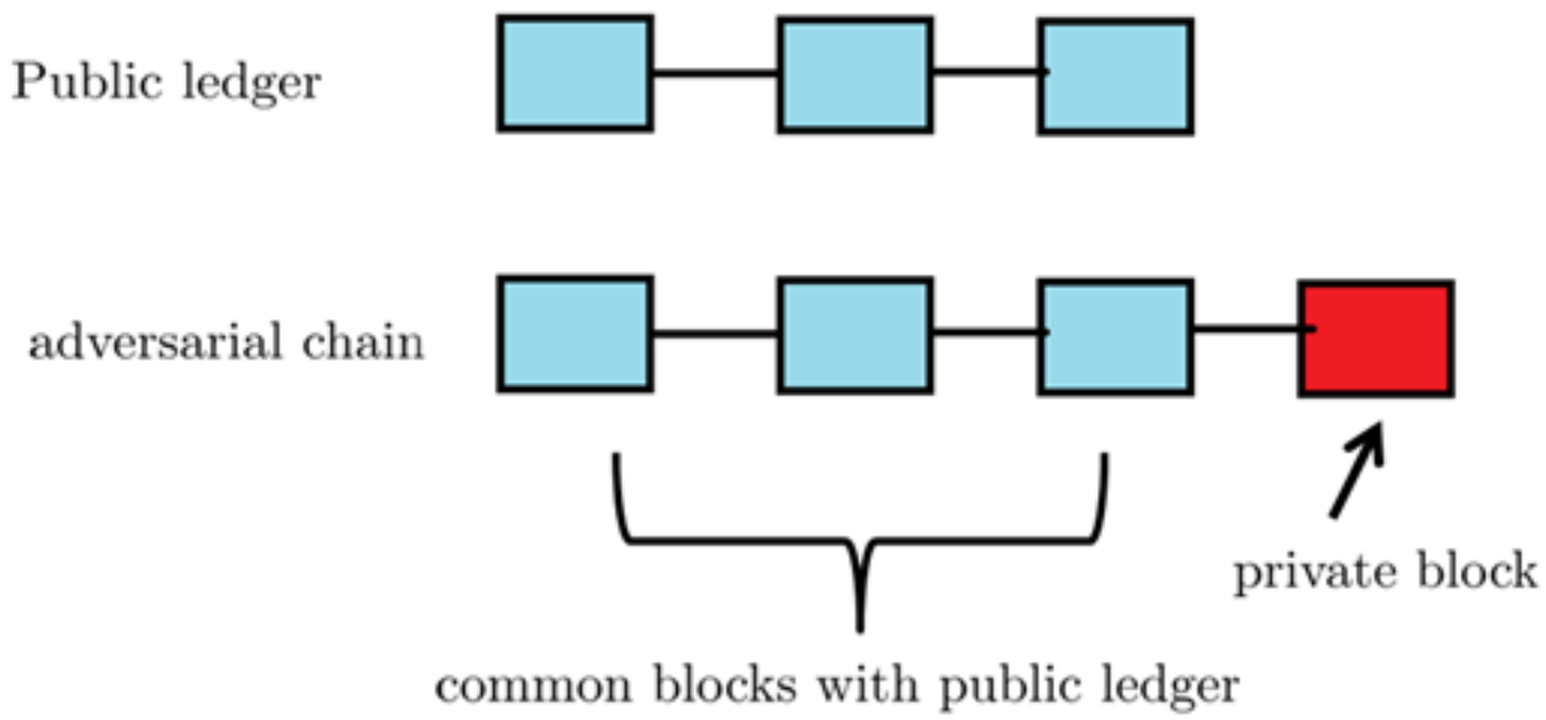
1)The attacker adopts the longest chain and tries to extend it.



Selfish Mining, III

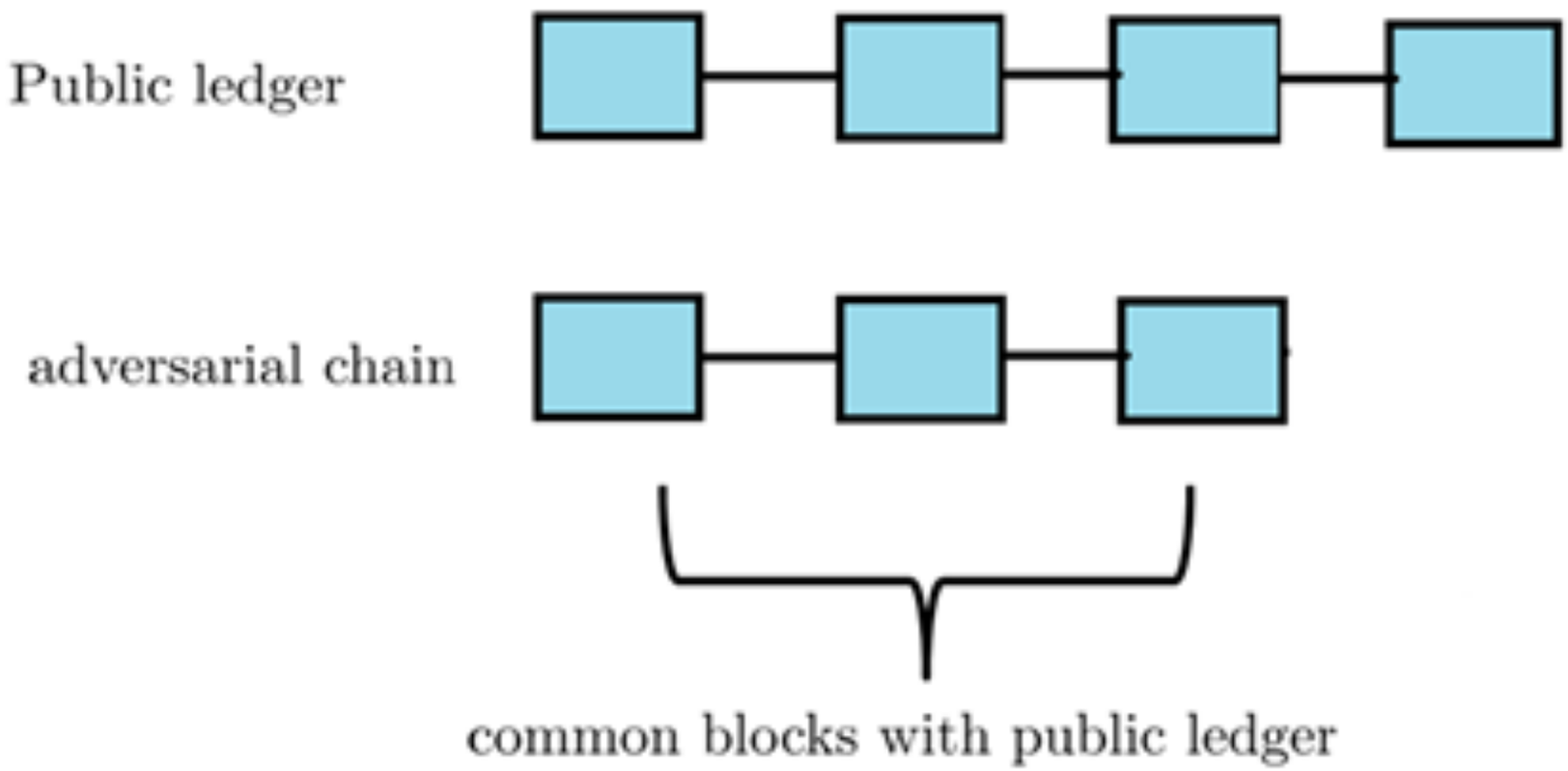
We have the following two cases : 2a or 2b

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



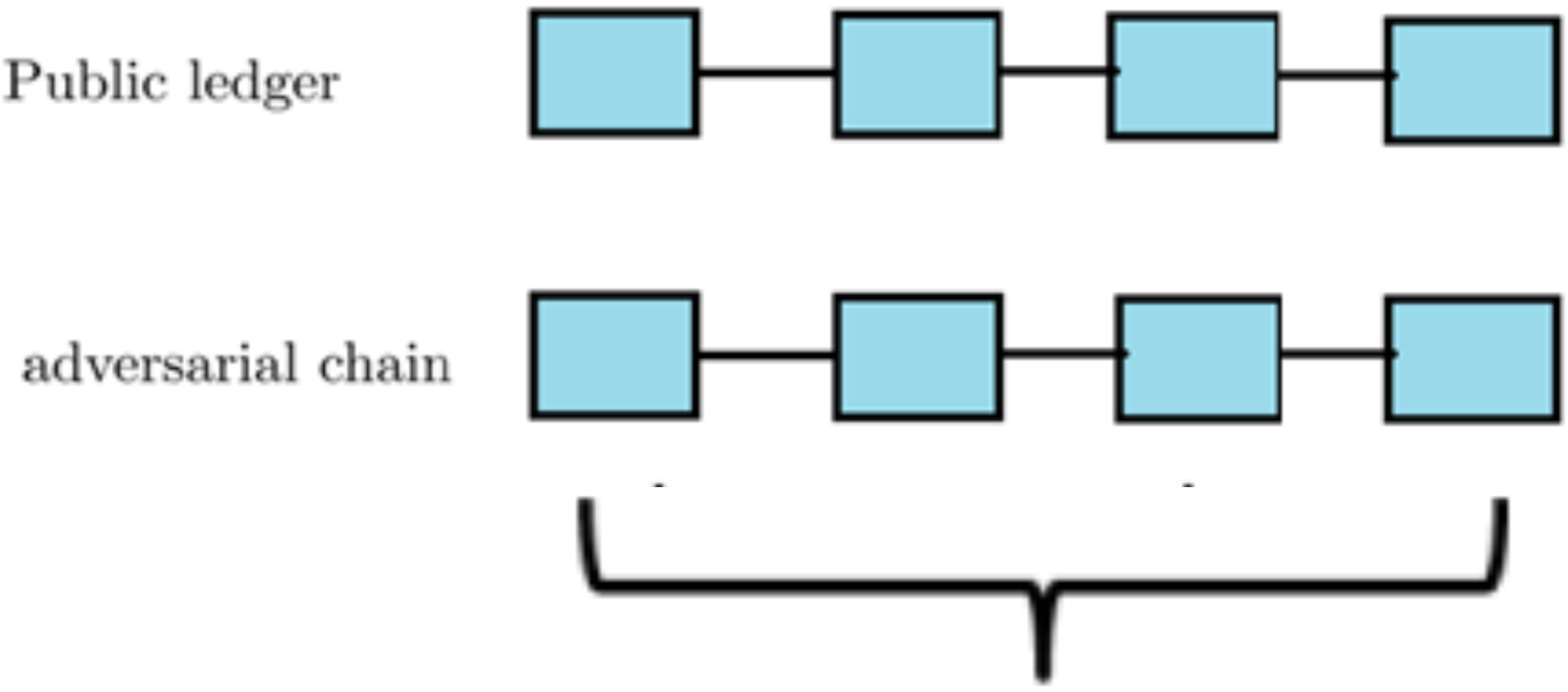
Selfish Mining, IV

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



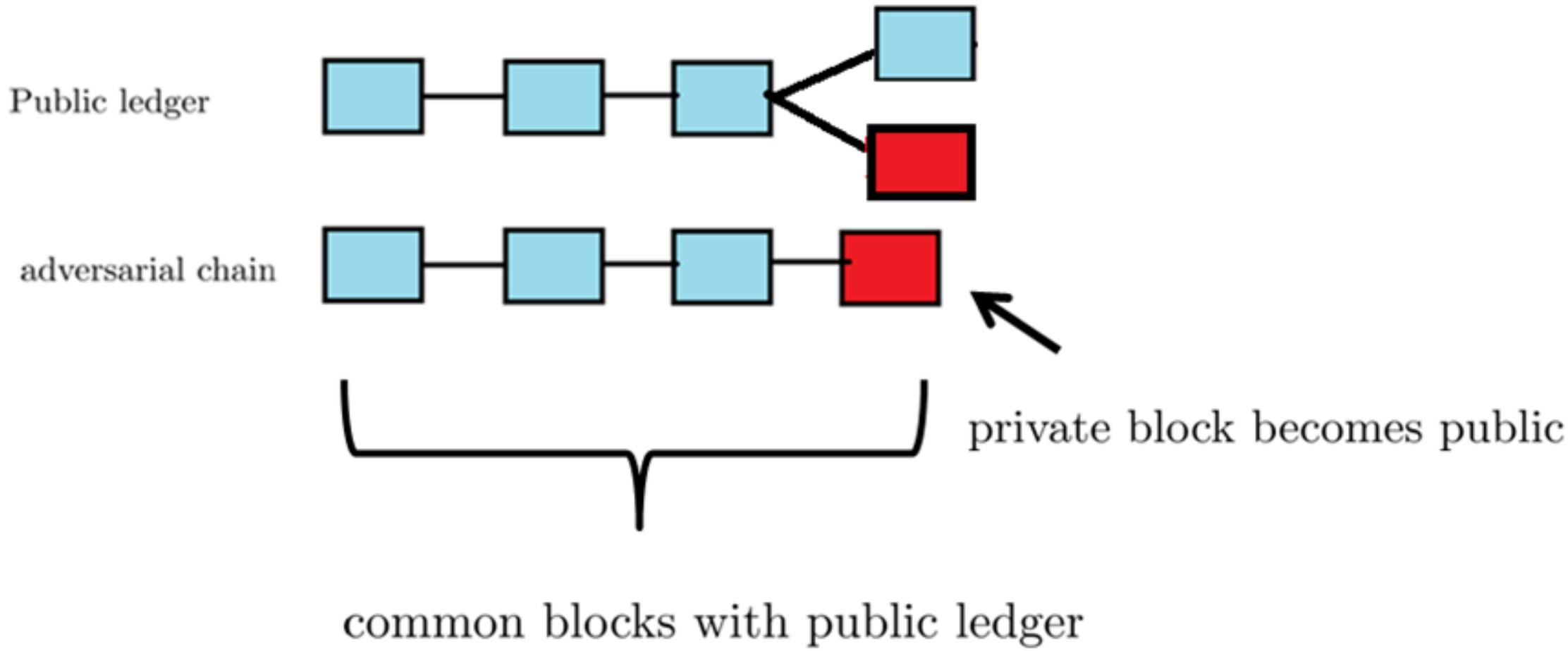
Case 2b

The attacker in this case adopts the public ledger.



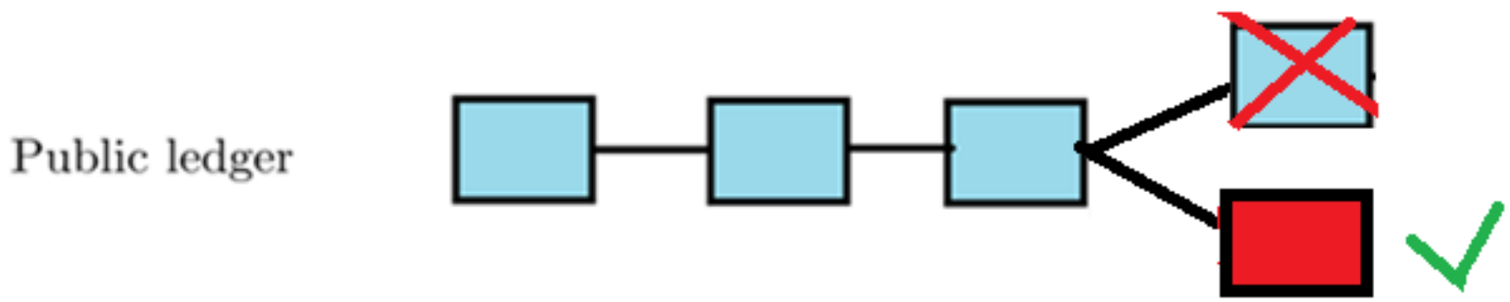
Case 2a

The attacker holds its block secret until another participant extends the public chain. When this happens it broadcasts its block.



Case 2a

If the other parties choose to extend the adversarial block then the adversary has dropped the last block of the public ledger.



When does this happen?

Case 2a

If the following two happen the attacker achieves to drop a block of the public ledger.

- The adversary manages to deliver first its block to the other parties.

- When an honest party receives two chains of the same length then it chooses the first that it received.

Selfish Mining, V

- Why the attacker can increase the expected value of its relative rewards supposing that it can deliver its blocks first and the honest parties adopt the first chain they receive in the case of a tie?

Selfish Mining, VI

- The computational power of the attacker contributes only to drop blocks and not to extend the public ledger.
- So when it implements the attack, the total number of blocks (expected value) in the public ledger is smaller compared to the total number of blocks in the case where it follows the protocol.

Selfish Mining, VII

- However if the attacker does not achieve to deliver its block first it loses the rewards from this block.

- This risk of the attack can be mitigated in the case when the flat reward becomes zero and the payoff of the miners is the transaction fees.

- In this case the attacker can choose to hold private only blocks of ``low risk" with low transaction fees and implement a more effective selfish mining attack.

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

Fairness, I

- Intuitively a blockchain protocol is fair if work done by the miners following the protocol is not rewarded due to actions of a malicious coalition.

- Selfish mining is a strategy that breaks fairness for bitcoin: the work performed by an honest miner to produce a block will not be rewarded because the block is knocked off the "main chain."

- Is it possible to design more fair protocols?

Fairness, II

- Recall GKL Consensus Protocol 2:
    - Players perform "2-for-1 POW" and issue "POW-inputs" that are attached to the chain.
    - The ratio of contributed POW-inputs of any subset of players is proportional to their hashing power, i.e., the blockchain contains a fair representation of contributed inputs from any subset of players.

Garay et al. "The Bitcoin Backbone Protocol: Analysis and Applications" (2015)

Fairness, III

- Taking this idea the next step, the "Fruitchain" blockchain protocol sets POW-inputs to be sets of transactions (nicknamed "fruits").

- Fruits can be accepted anytime for a certain window of opportunity (but old fruits are rejected).

- Fairness follows assuming a flat amount of effort per block of transactions.

Pass Shi, Fruitchain (2017)

Block Reward Zero: Attack

- When the flat reward becomes zero the following attack 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. (*PettyCompliant strategy*)

- Other miners can take advantage of this behavior 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 its block a transaction T0 that gives *bribe* money to miners who will adopt the fork and will extend the chain of the attacker.

- The input of T0 is also in the chain of the attacker and double spends money that exists in the public ledger.

- If the chain of the attacker does not manage to become longer than the public ledger then the attacker does not lose the *bribe* money as the transaction T0 has invalid input.

- However, 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).

Verification of Transactions

- Each miner should verify the transactions of the block that it tries to extend and check if a transaction spends money that has already been spent.

- It should also verify the validity of the transactions which it will include in the block it tries to produce.

- If the miner includes an invalid transaction in its block then the miners should reject this block.

Verification of Transactions: Incentives

- Miners have incentives to verify the validity of the transactions when the time they need to do the verification is very small compared to the time they need to produce a block.

Luu, Loi, et al. "Demystifying incentives in the consensus computer." (2015)

Transactions' Broadcast: Incentives

- If a transaction offers a very high transaction fee then the miners have no incentives to broadcast it.

- Some papers propose that the transaction offers a reward for propagating.

Babaioff, Moshe, et al. "On bitcoin and red balloons.". (2012)

Abraham, Ittai, et al. "Solidus: An incentive-compatible cryptocurrency based on permissionless Byzantine consensus." (2016)

Mining Pool

- Miners can participate in mining pools.
- The miners in a mining pool try to produce blocks of smaller difficulty ``shares" that give to the pool managers.
- When among them a block of the proper difficulty appears then the pool manager broadcasts it and rewards the miners according to their shares.
- The shares of the smaller difficulty are the partial proofs of work (PPoWs) and the shares of the proper difficulty are the full proofs of work (FPoWs).

Block Withholding Attack (BWH)

- The attacker submits only PPoWs to the mining pool not FPoWs and simultaneously does solo mining.

Rosenfeld, Meni. "Analysis of Bitcoin pooled mining reward systems."(2011)

Courtois, Nicolas T., and Lear Bahack. "On subversive miner strategies and block withholding attack in bitcoin digital currency." (2014)

Fork After Withholding (FAW) Attacks on Bitcoin

- The attacker uses its computational power to do solo mining (*innocent mining*) and to participate in the mining pool (*infiltration mining*).

- If it finds a block by innocent mining it broadcasts it.

- If it finds a FPoW in the mining pool, it does not give it to the pool manager and one of the following happens:

  - If another miner from the mining pool finds a FPoW, then the attacker discards its FPoW.

  - If another miner out of the mining pool finds a block then the attacker gives the FPoW to the pool manager and the manager broadcasts it making a fork.

  - If the attacker finds another FPoW by solo mining, it discards the old one and it broadcasts the new one.

Kwon, Yujin, et al. "Be Selfish and Avoid Dilemmas: Fork After Withholding (FAW) Attacks on Bitcoin."(2017)

End of Lecture 05