Blockchains and Distributed Ledgers
Lecture 05

Aggelos Kiayias

credits: Aikaterini-Panagiota Stouka
for slides preparation

THE UNIVERSITY of EDINBURGH

Lecture 05

- Bitcoin's Reward Mechanism.
- Incentive Compatibility of Bitcoin.
- Mining pools.
- Various protocol deviations.
- Fairness.

Incentives for the Ledger

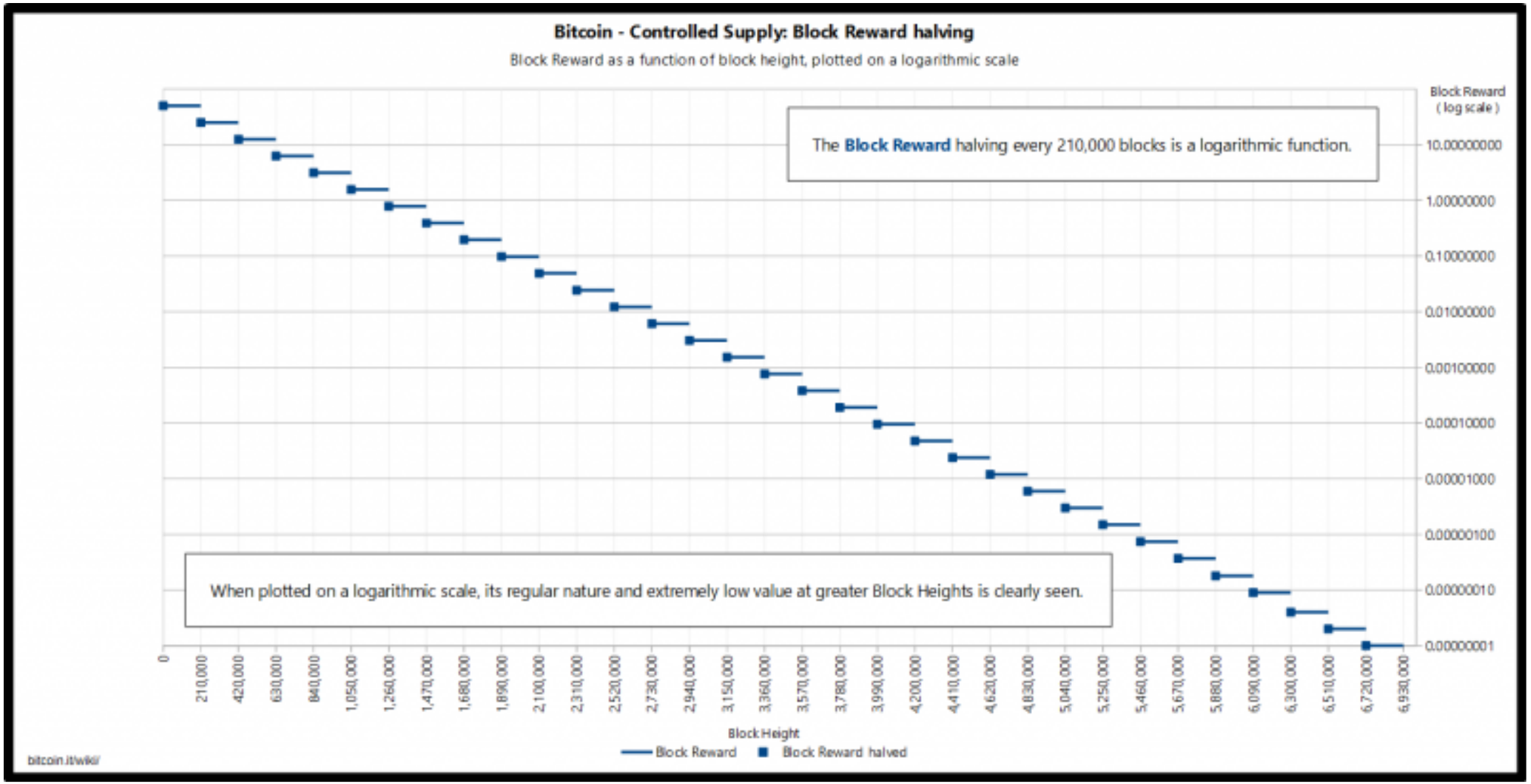
- So far : honest majority in hashing power implies robustness of transaction ledger.

- .. 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 reward is mined the time the block is produced.

- 50% reduction of the reward 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(s) from the value of the input(s).

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 rewards and the transaction fees to the address of the miner.

- Note: even if two blocks include the same transactions in the same order, the coinbase transaction is different.

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

A dominant strategy is not necessarily one that will result in the best possible outcome

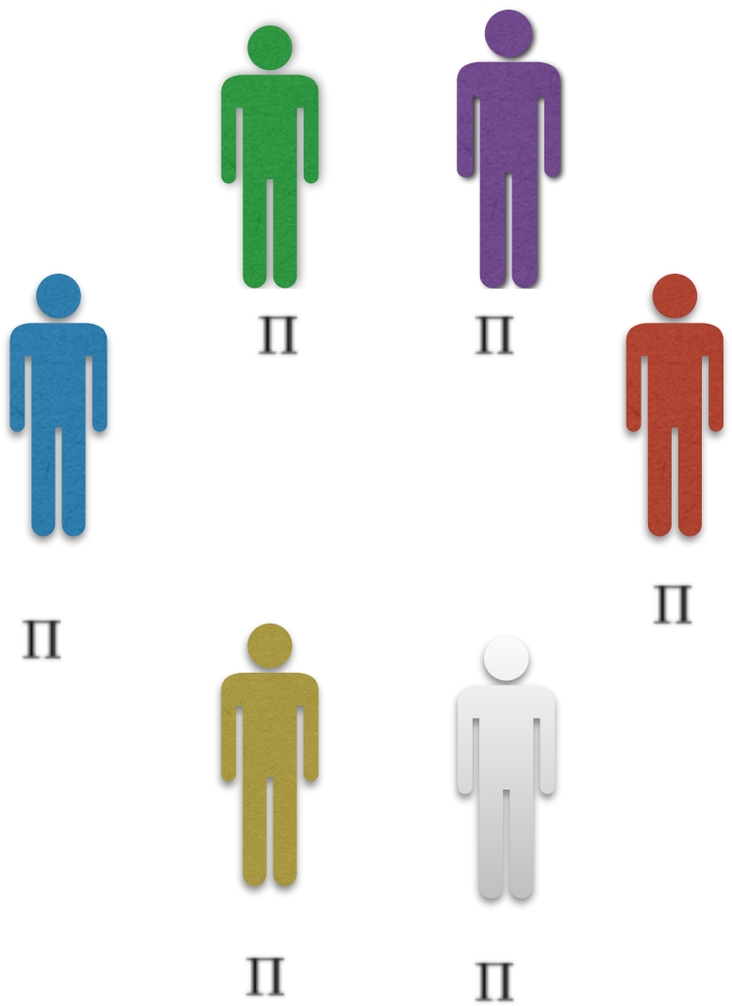
Definition: A Protocol is Nash Equilibrium I

- All the participants are rational and want to maximize the utility they obtain at the end of the execution.

- Utility of a participant is a function that takes as input the strategies of all the participants and has as output a real number that represents the gains of this participant at the end of the execution.

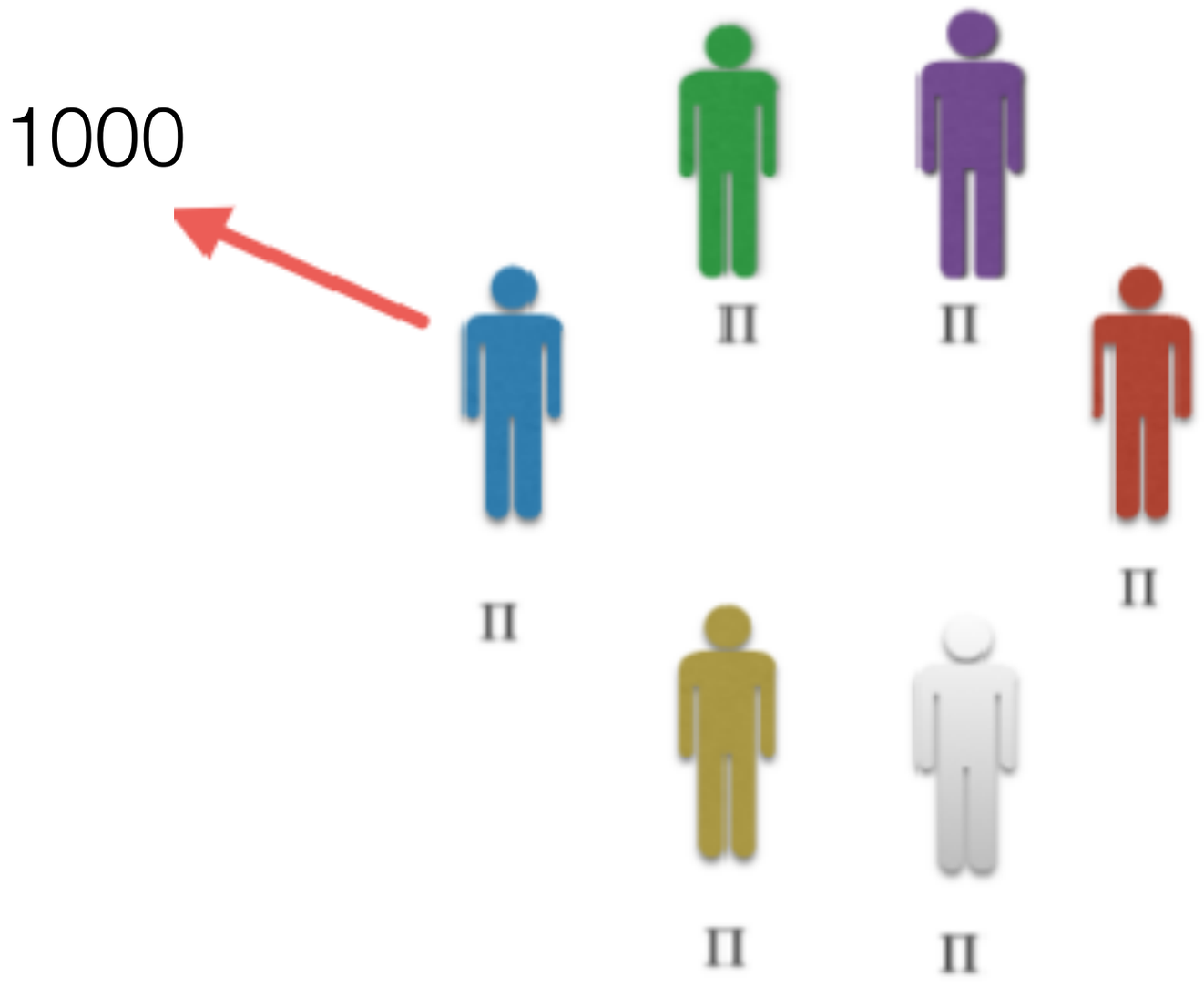
Definition: A Protocol is Nash Equilibrium II

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

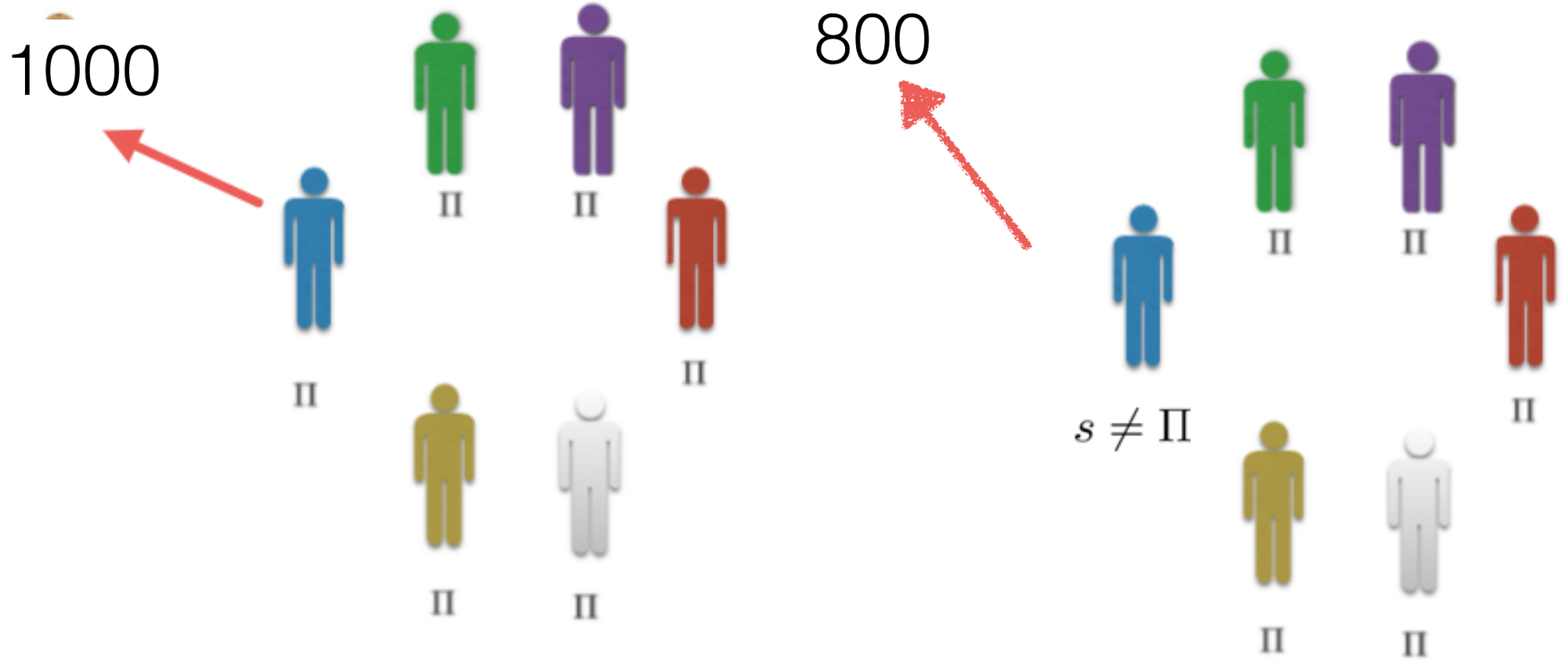


Protocol Π is Nash equilibrium

Definition: A Protocol is Nash Equilibrium III



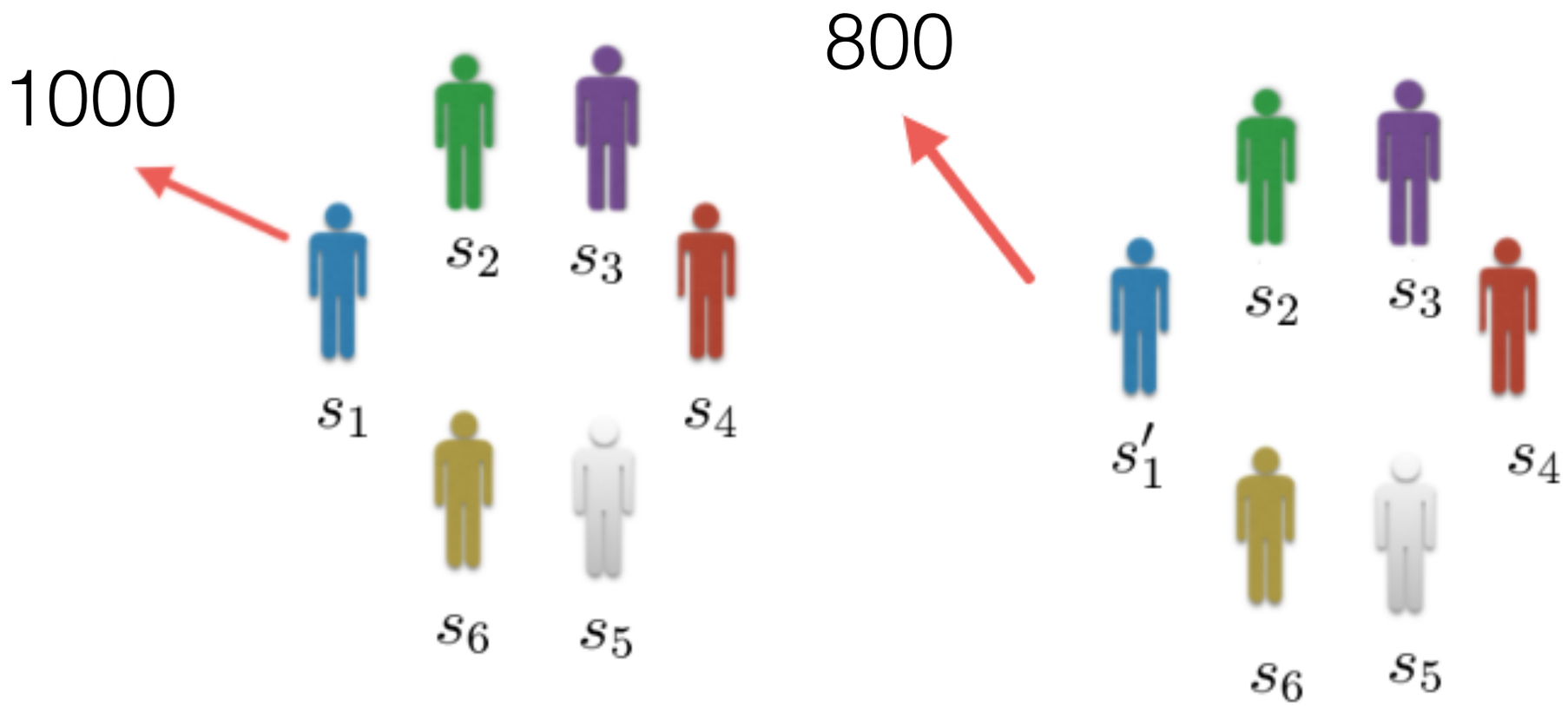
Definition: A Protocol is Nash Equilibrium IV



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 any benefits that they receive at the end of the protocol?

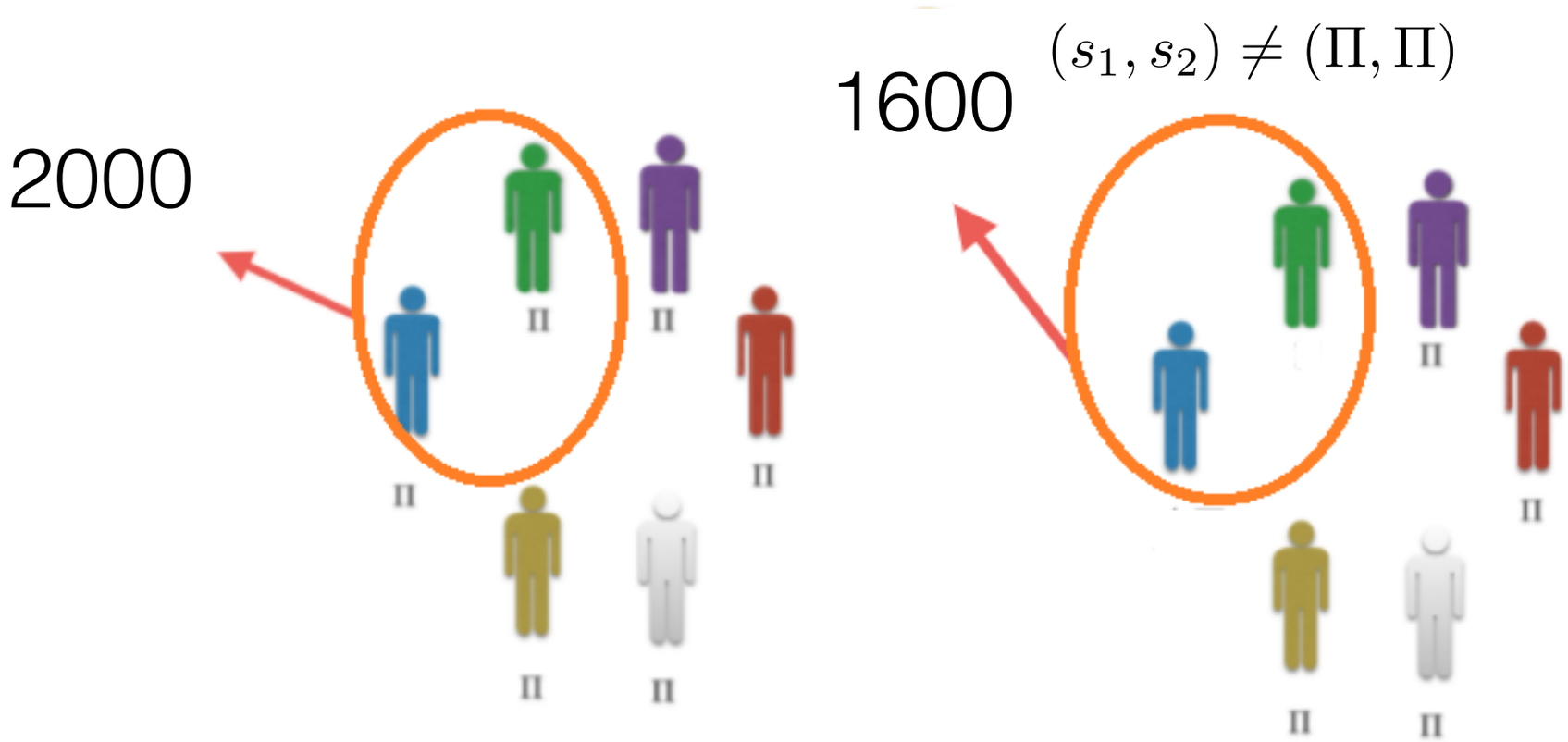
Utility Coalitions

- Utility of a coalition in the previous example was the sum of the rewards of the members.

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

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.

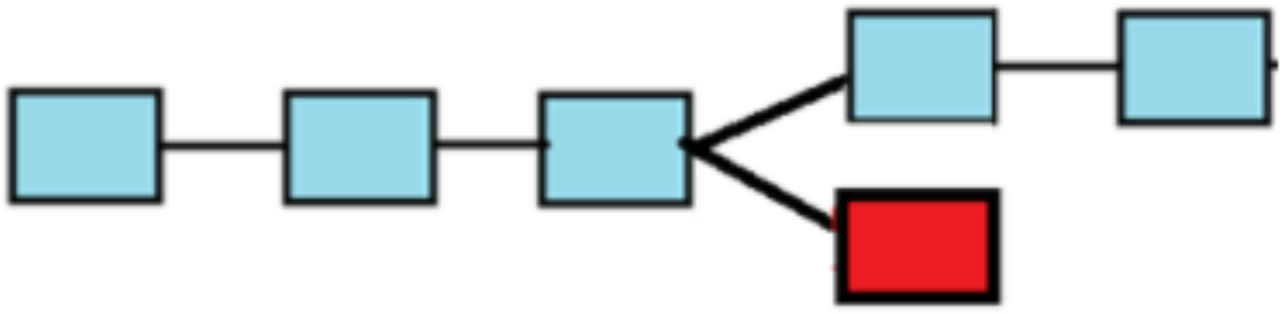
Types of Utility in Bitcoin

- What could be the utility in Bitcoin?

- How could utility be defined in a probabilistic protocol?

Absolute Rewards I

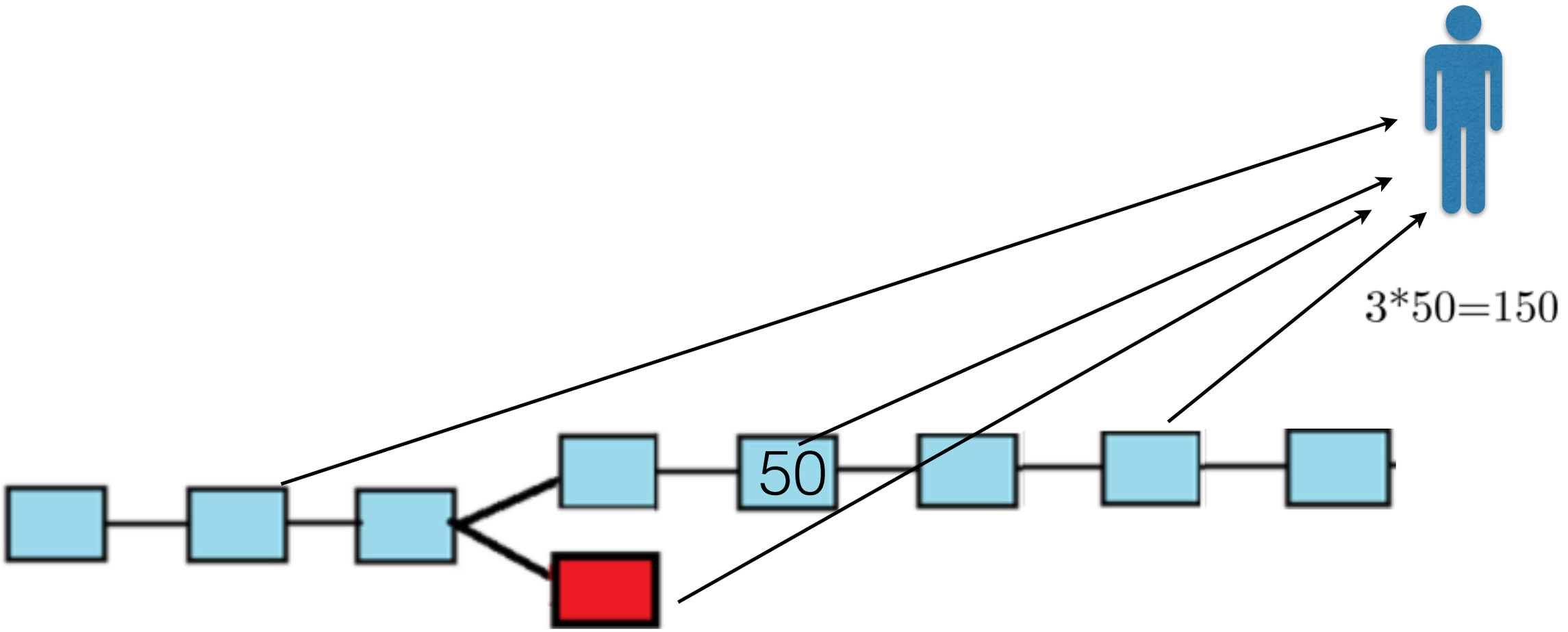
- If we specify the algorithms of all the participants and the outcome of all the randomness used by participants in a finite execution of the Bitcoin protocol then we have a unique outcome.



Absolute Rewards II

- Each block of the longest chain gives a reward to its producer. (e.g., initially this reward was 50 BTC).
- **Absolute rewards utility**. The utility of a coalition is equal to the number of BTC that it has obtained at the end of the execution.

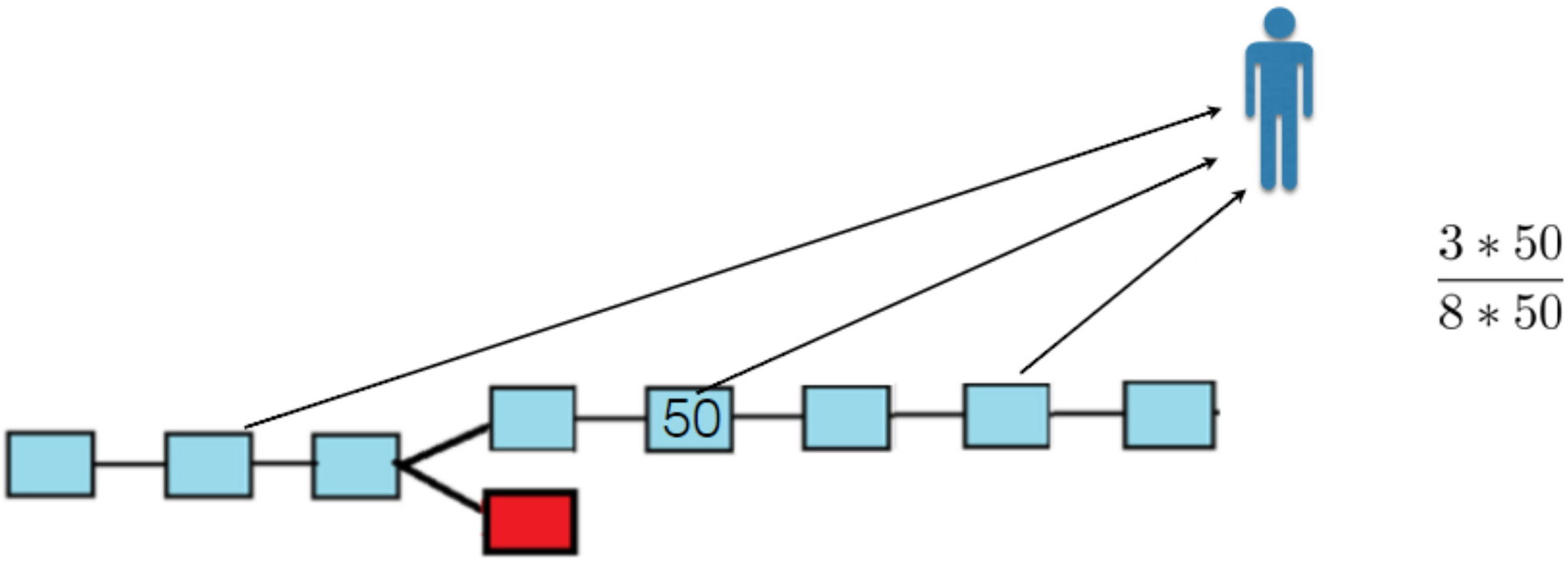
Absolute Rewards III



Relative Rewards I

- **Relative Rewards**. The utility of a coalition in Bitcoin is equal 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

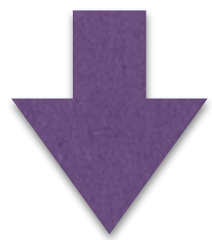
- Given the strategies of all the participants the outcome of the Bitcoin execution is a random variable. So the utility of a coalition is also a random variable. How to resolve this?

  - Via Expectation: will determine the expected value of utility.

  - Via events that happen with high probability.

Bitcoin Incentives, I

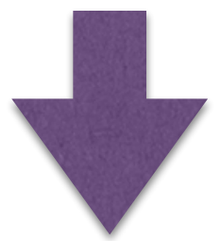
- Kroll et al. in ``The economics of Bitcoin mining, or Bitcoin in the presence of adversaries" (2013) show that a certain modeling of the Bitcoin protocol is a Nash equilibrium.



- Reasoning : If utility is equal to the expected value of absolute rewards and block difficulty is stable, then the expected number of blocks is proportional to mining power and this is delivered by Bitcoin.

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 equal to the expected value of relative rewards.

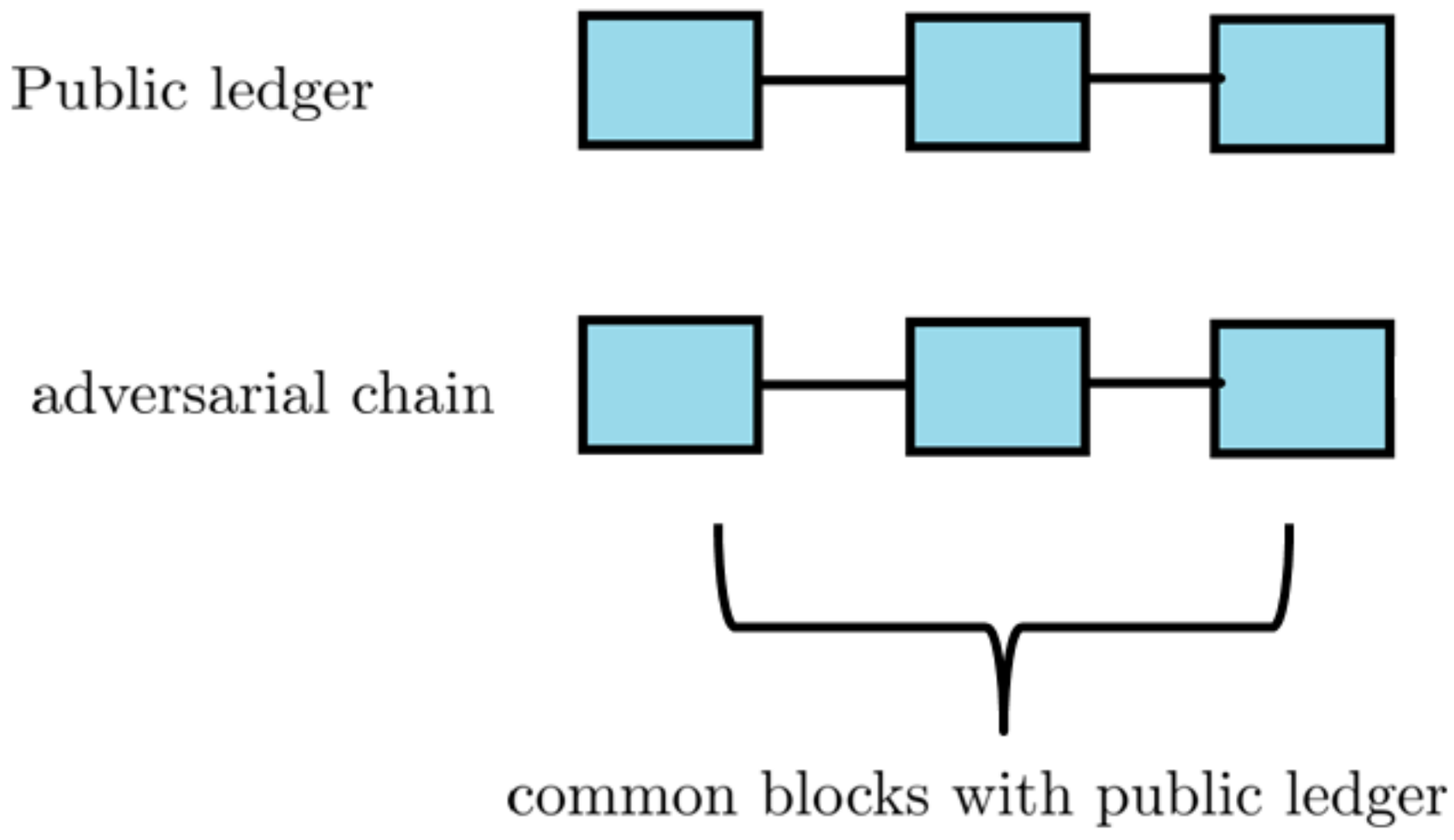
Selfish Mining, I

- 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

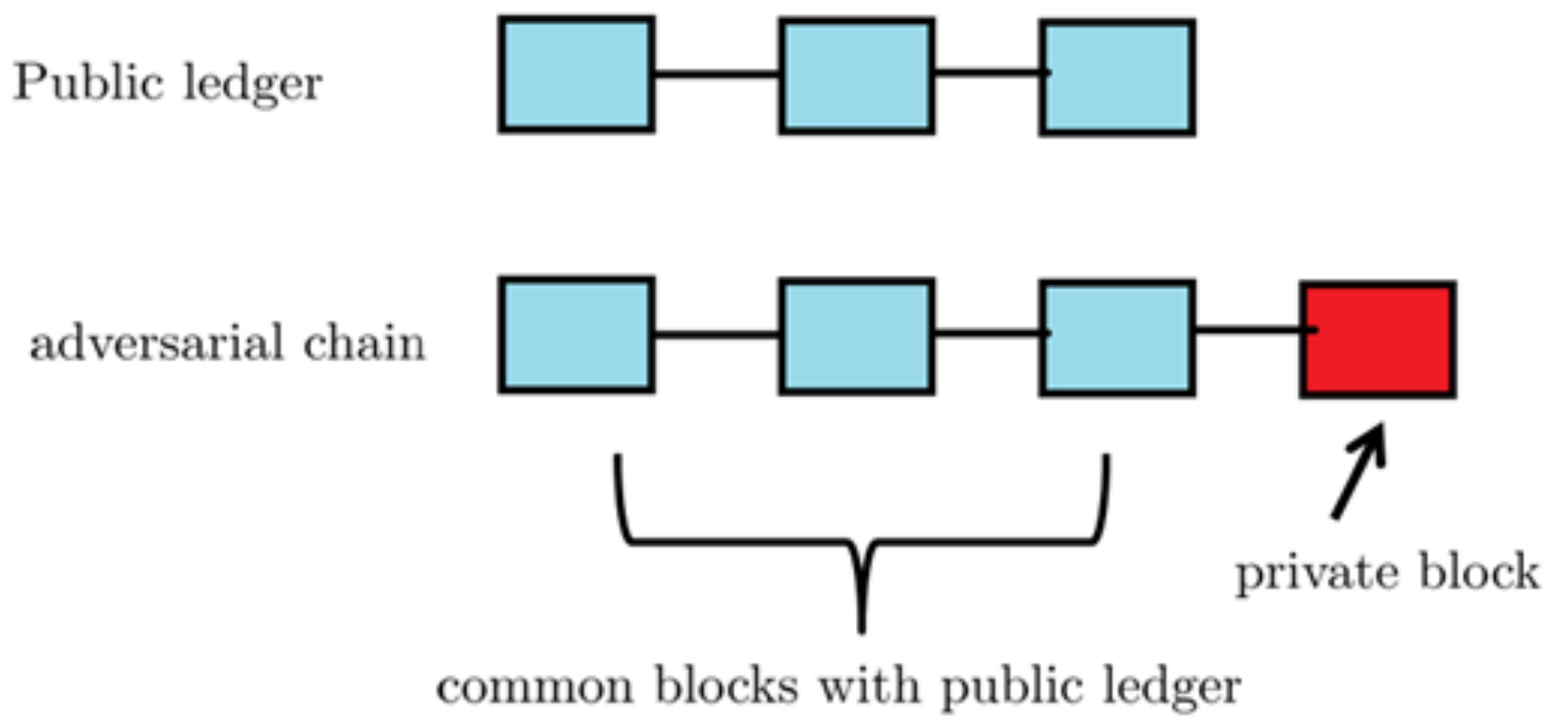
1)The attacker adopts the longest chain and tries to extend it.



Selfish Mining, III

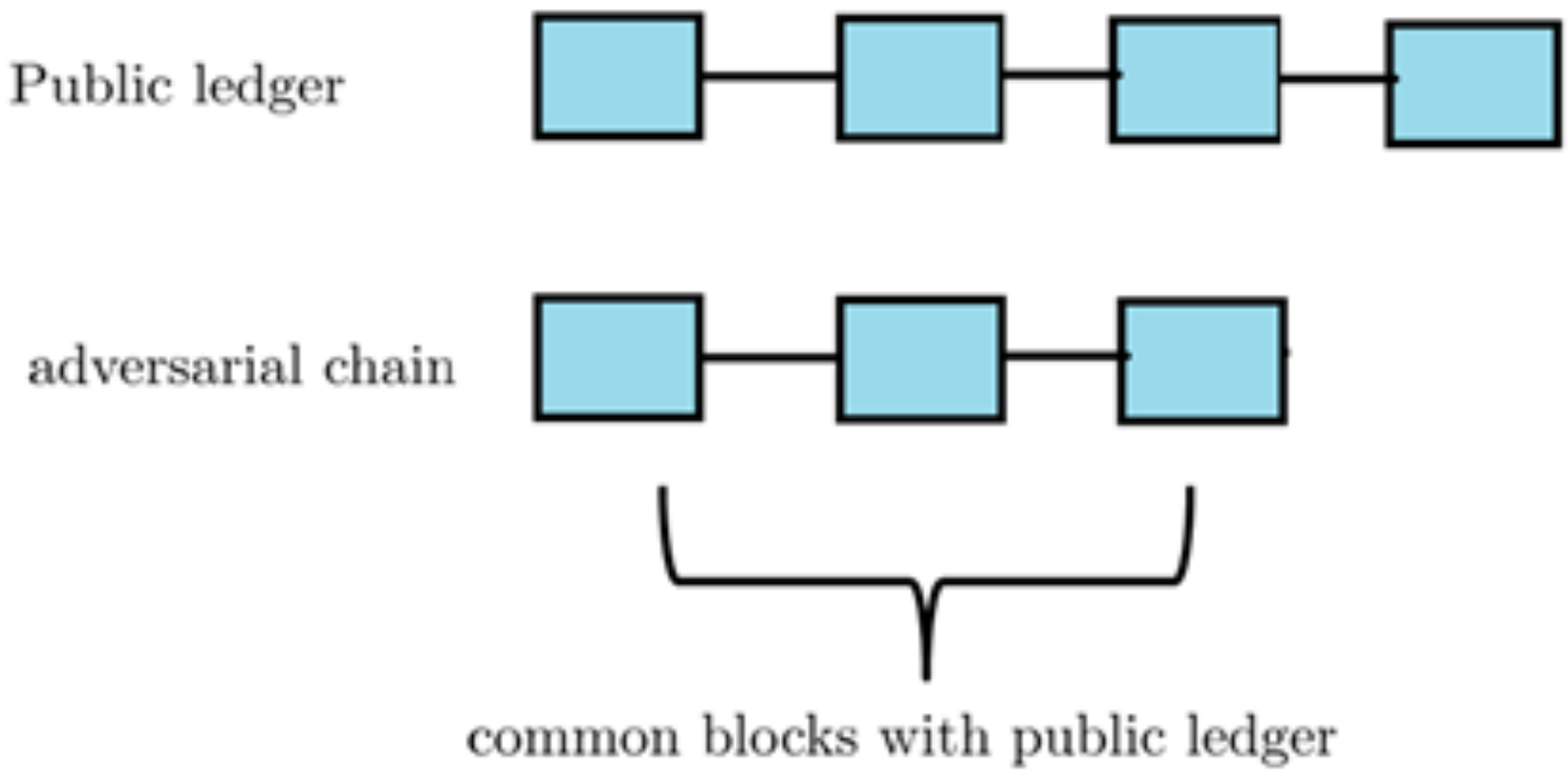
We have the following two cases : 2a or 2b

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



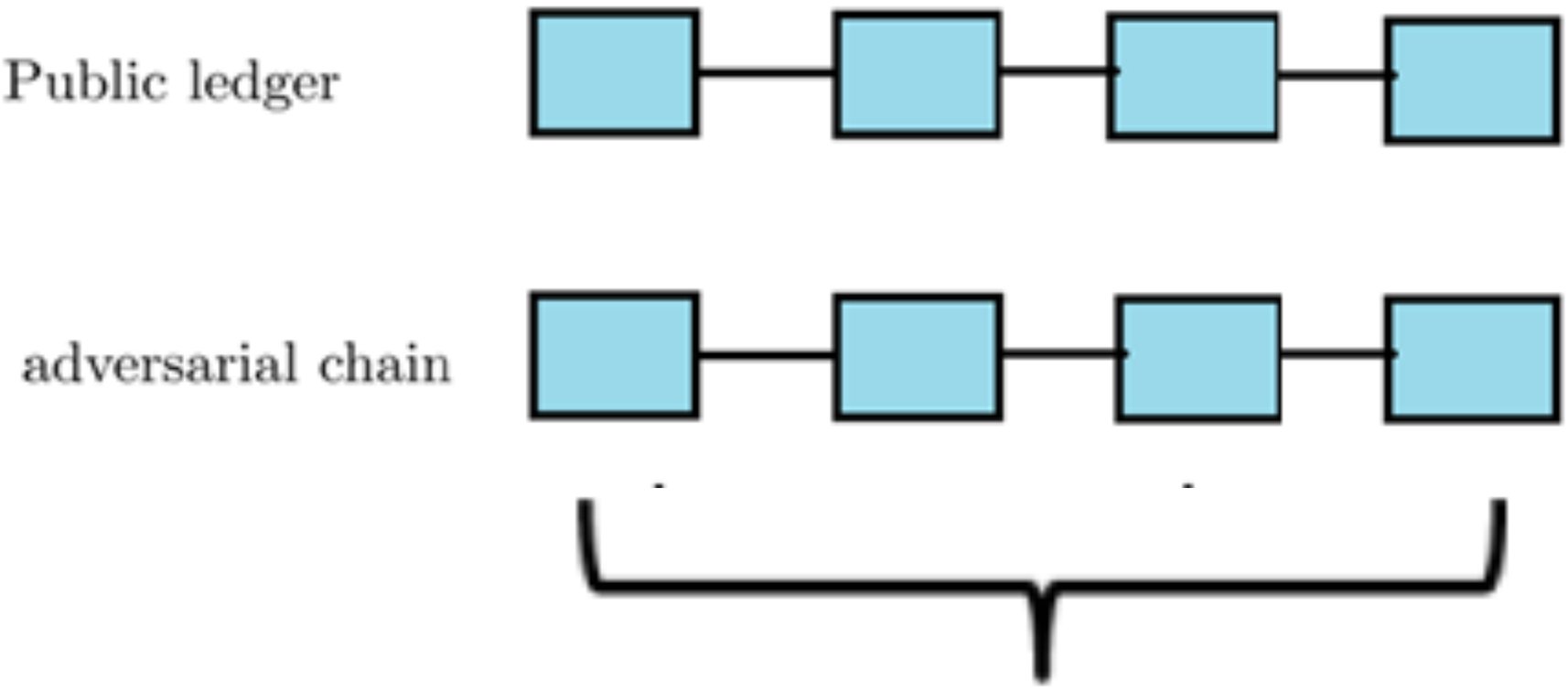
Selfish Mining, IV

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



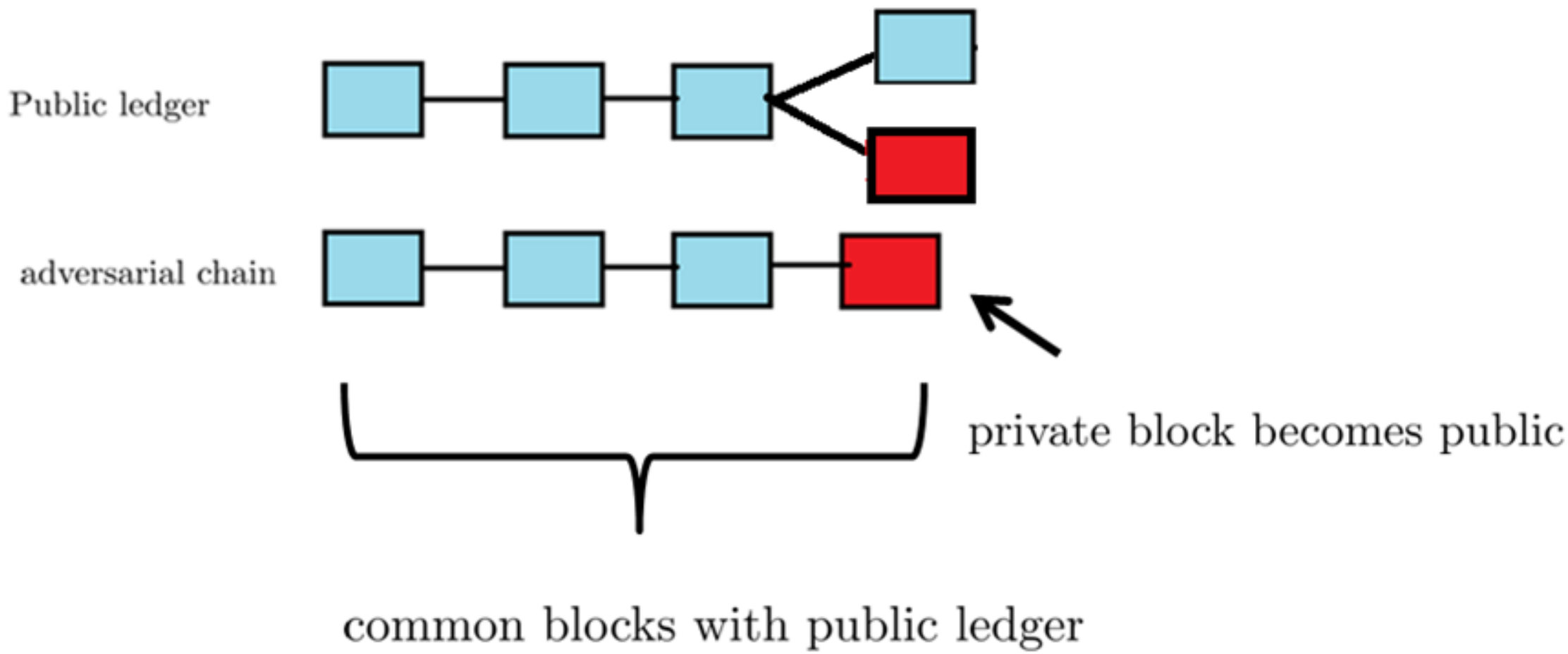
Case 2b

The attacker in this case adopts the public ledger.



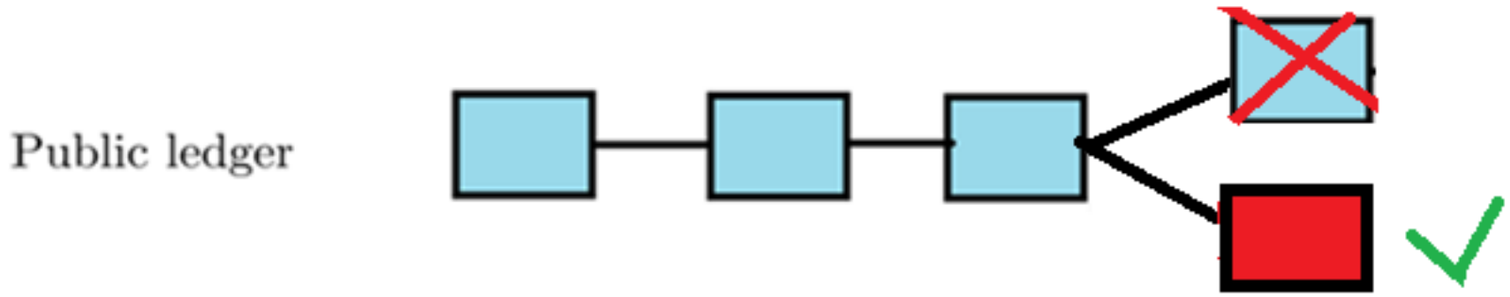
Case 2a

The attacker 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)



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.



When does this happen?

Case 2a

If the following two happen the attacker will be capable of censoring blocks

- The adversary gets two blocks ahead of the public chain.

- 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

- The computational power of the attacker contributes only towards censoring 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.

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

Selfish Mining, VI

- **Analysis:** Consider a process operating in “block rounds." Attacker has probability α to produce the next block. Assume attacker always wins the network race vs. public blocks.

  - Honest play: *n* rounds result to *n* blocks. Attacker owns αn blocks in expectation.
    Utility (Relative Rewards) = α, (Absolute Rewards) = α n.

  - Selfish play: *n* rounds result to (1-α)*n* blocks. Attacker owns αn of those blocks.
    Utility (Relative Rewards) = α/(1-α), (Absolute Rewards) = α n.

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 T0 that gives *bribe* money to miners who will adopt the fork and will extend this block.

- The input of T0 is also transmitted in the public ledger and double spends the bribe money.

- 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.

- 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.

- If the effort to verify is very large, then miners may choose not to verify in favour of devoting their computational power to block production.

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.
    - Instead they would keep it private so that they claim it.
    - Is it possible to reward transaction relaying?

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 Pools

- Miners can participate in mining pools.
- The miners in a mining pool try to produce blocks of smaller difficulty (some times referred to as “shares”) that present as proofs of their effort to the pool manager.
- When among them a block of the proper difficulty appears then the pool manager broadcasts it and rewards the miners according to shares submitted.
- 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

- The attacker submits only PPoWs to the mining pool, not FPoWs, and simultaneously does solo mining.
- A mining pool can successfully attack another pool in the relative rewards setting.

Rosenfeld, Meni. "Analysis of Bitcoin pooled mining reward systems."(2011)
Eyal, Ittay. "The Miner’s Dilemma” (2014)

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)

End of Lecture 05

- Next lecture:
  - Account management in distributed ledgers.
  - Secure Multiparty Computation
  - Fair swaps of assets.