ETH zürich

FruitChains: A Fair Blockchain

Presented by Manuel Grossmann

Structure

- Motivation of a Fair Blockchain
- Results of the Paper
- Selfish Mining
- Security of Blockchain Protocols
- Defining Fairness
- FruitChains Protocol
- Conclusion

Motivation of a Fair Blockchain

- Selfish mining undermines incentive compatibility
- Transaction fees exacerbate instability
- Mining pools harm decentralization

Results of the Paper

- Coalition of adversaries can't get more than their "fair" share (plus some δ)
  → selfish mining is not possible anymore

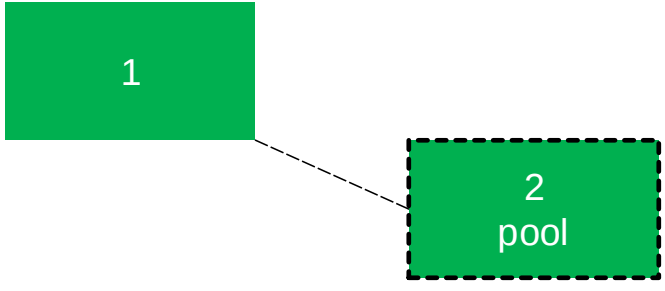
- Spreading out the transaction fees of a block over the miners of a sequence of blocks preceeding it
  → transaction fees no longer cause instability

- Mining difficulty can be made arbitrarily small, thus miners get paid more often
  → mining pools are no longer needed

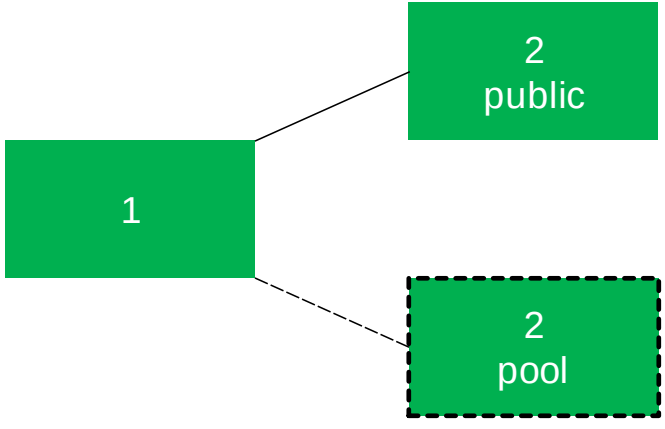
Selfish Mining

- Observed and discussed in Bitcoin forum
- First study in «Majority is not Enough: Bitcoin Mining is Vulnerable» by Eyal and Sirer
- A minority pool can obtain more revenue than its fair share

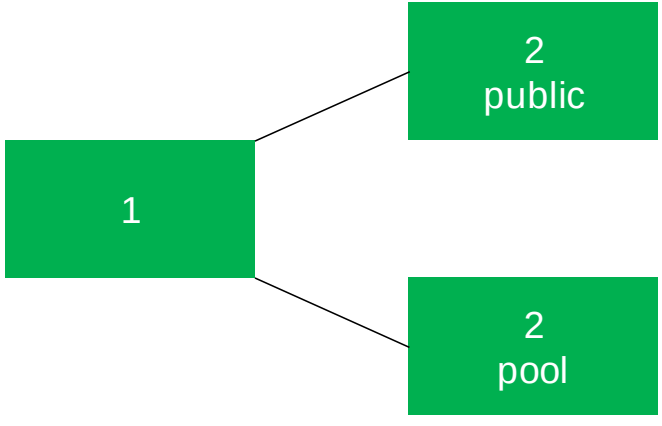
Selfish Mining cont.



Selfish Mining cont.

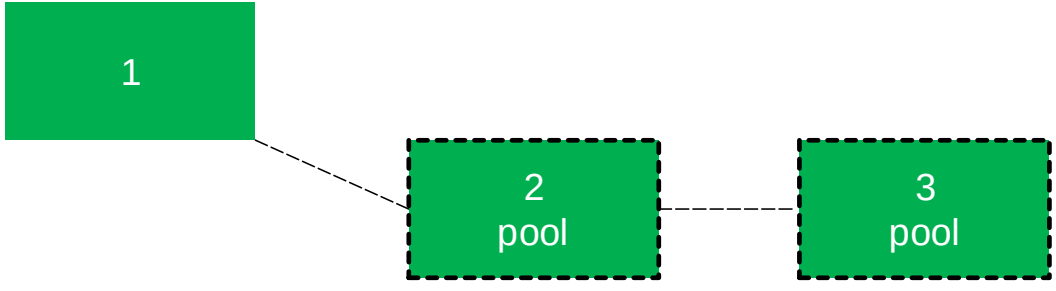


Selfish Mining cont.

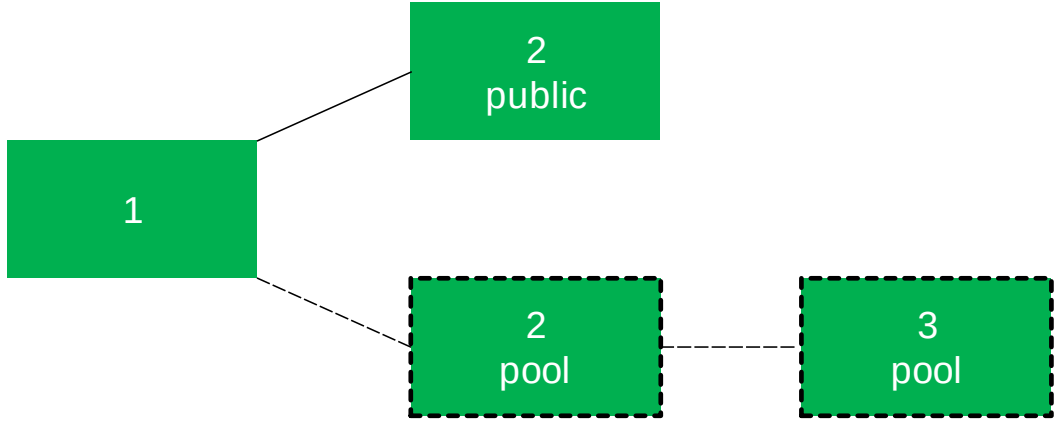


- The pool tries to mine on its previously private head, and the others split between the two heads
- Denote by γ the ratio of others that choose the non-pool block

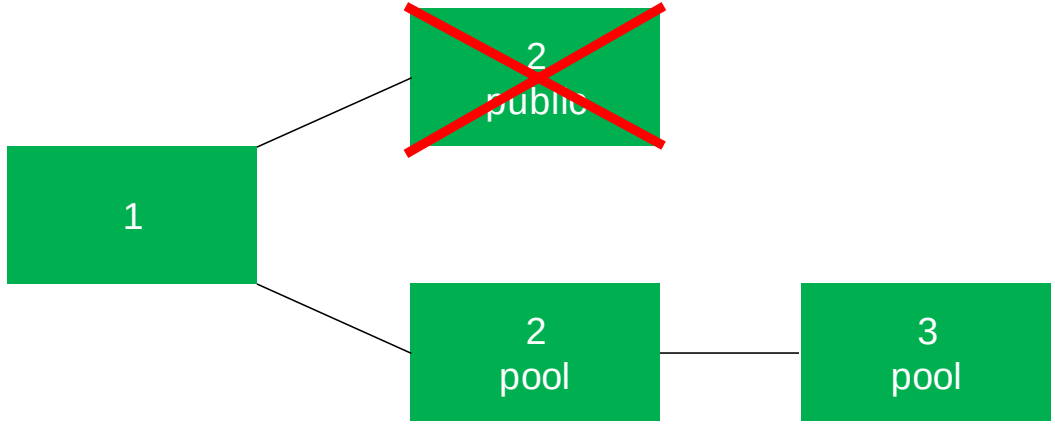
Selfish Mining cont.



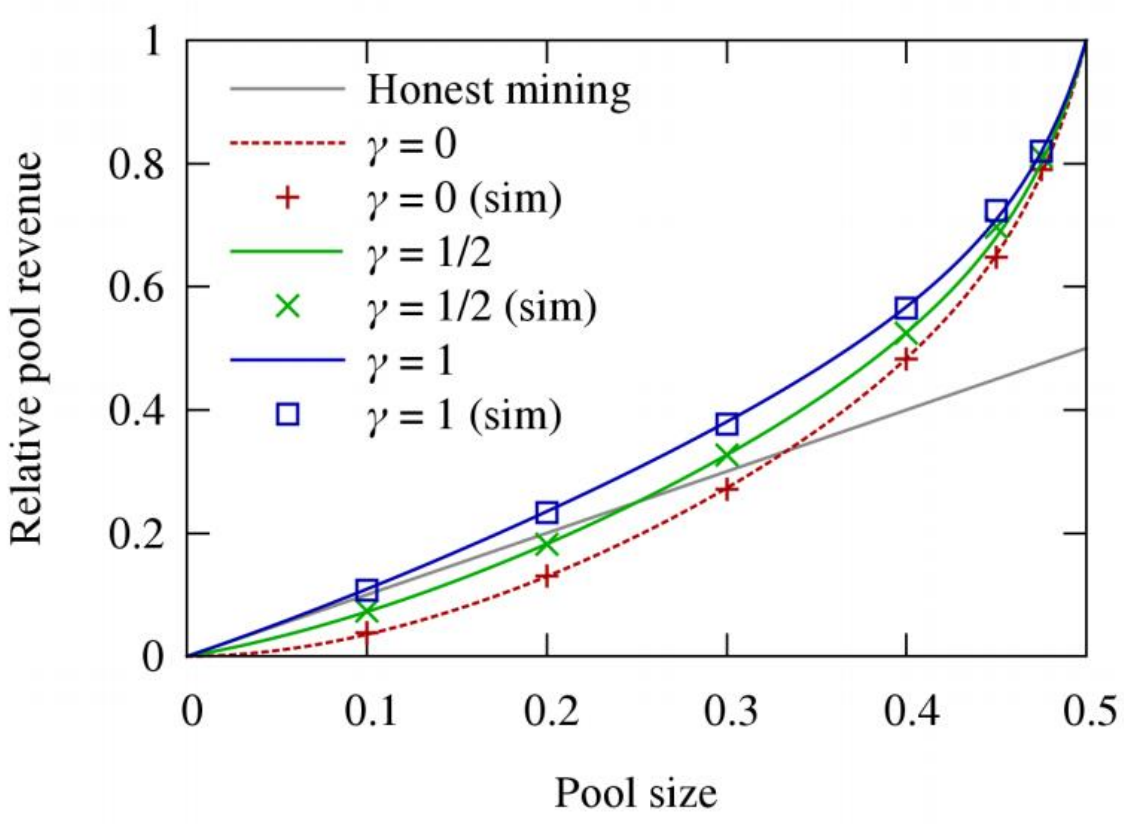
Selfish Mining cont.



Selfish Mining cont.

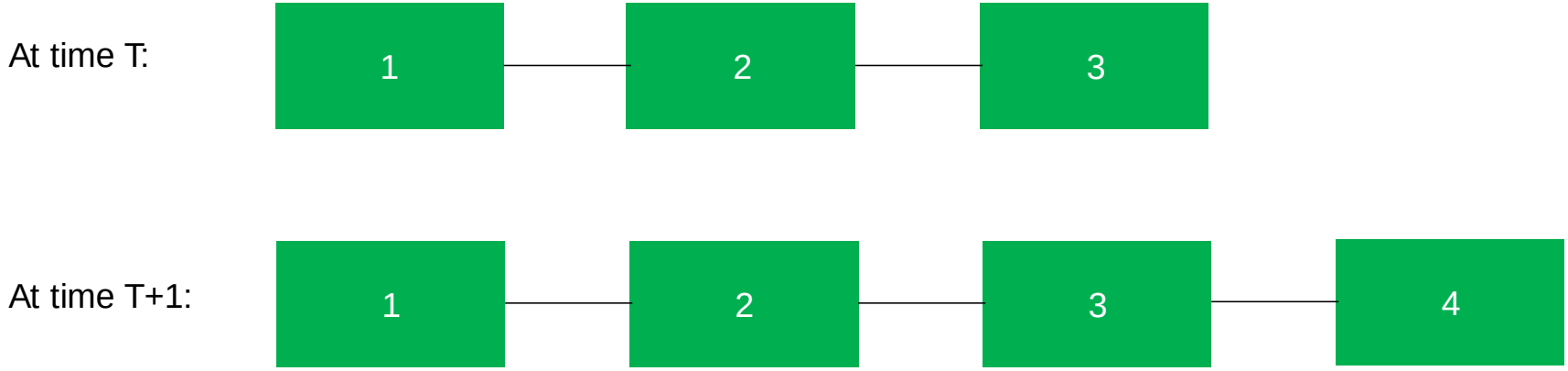


Selfish Mining cont.



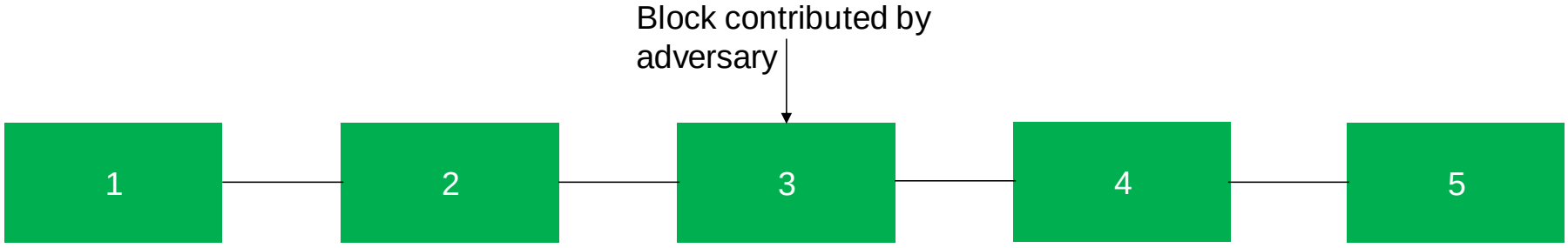
Security of Blockchain Protocols

- Chain growth: The chain grows proportionally with the number of time steps



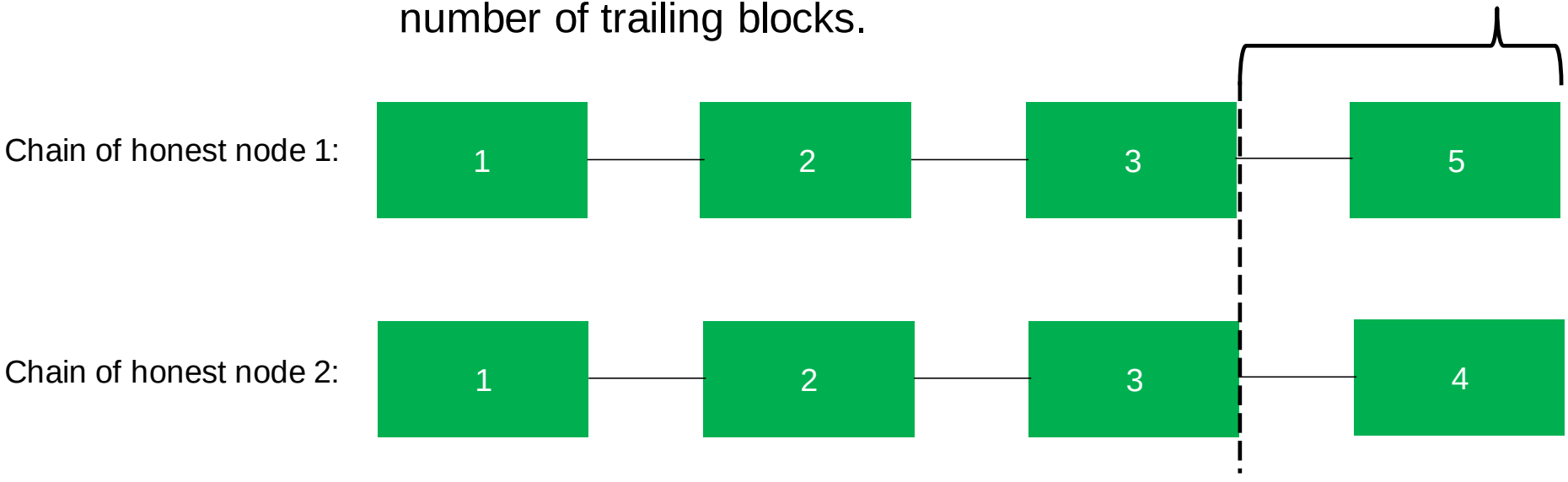
Security of Blockchain Protocols cont.

- Chain quality: The number of blocks contributed by the adversary is not too large



Security of Blockchain Protocols cont.

- Chain consistency: All honest nodes' chains are the same, exept for roughly O(κ) number of trailing blocks.



Defining Fairness for Static Corruption

- Adversary must declare corrupt nodes upfront
- (T, δ)-approximate fairness:

  The fraction of blocks contributed by nodes in S is at least $(1-\delta)\phi$ for any T consecutive blocks in the chain.
- Where S is any subset of honest nodes and ϕ the fraction that $|S|$ is w.r.t. the total number of nodes

Defining Fairness for Adaptive Corruption

- The subset of honest node can change anytime

Definition 3.1. A blockchain protocol Π has (approximate) *fairness* (T_0, δ) in Γ-environments, if for all Γ-compliant p.p.t. (A, Z), every positive constant $\phi \leq 1 - \rho$, every ϕ-fraction subset selection S, there exists some negligible function ϵ such that for every $\kappa \in \mathbb{N}$ and every $T \geq T_0$ the following holds:

$$\Pr\left[\text{view} \leftarrow \text{EXEC}^{\Pi}(A, Z, \kappa) : \text{quality}^{T,S}(\text{view}, (1-\delta)\phi)) = 1\right] \geq 1 - \epsilon(\kappa)$$

Defining Fairness cont.

- No ρ-size coalition can get more than a factor (1+δ) more than its fair share of blocks
- If a blockchain protocol satisfies (T_0, δ)-fairness, then it satisfies (T_0, μ)-chain quality where $\mu = (1 - \delta)(1 - \rho) \geq 1 - (1 + \delta)\rho$

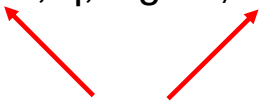
FruitChains Protocol

- Running instance of Nakamoto's Blockchain protocol
- Instead of records m inside blocks → records are stored inside "fruits"
- Fruits themselves requires to solve some proof of work with a different hardness parameter p_f
- Additionally a fruit "hangs" from a recent block

Valid Fruits / Valid Blocks

- Fruit f = (h_{-1}; h'; η; digest; m; h)

- Block b = ((h_{-1}; h' ; η; digest; m; h), F)



- h_{-1}: points to the previous block's reference
- h': points to a (recently stablized) block
- η: random nonce denoting the puzzle solution
- digest: denoting the hast of the fruit-set F
- m: the record to be contained in the fruit
- h: the hash of the previous fields
- F: fruit-set of valid fruits

FruitChains vs Selfish Mining

- Even if an adversary tries to “erase” a block mined by an honest player (which contains some honest fruits), the chain growth and chain quality properties guarantee that eventually an honest player will mine a new stable block which includes the “lost” fruits

- The time before such a block arrives is short enough for the fruits to still be recent

FruitChains vs Mining Pools

- Set the difficulty of mining a fruit as small as solving a partial proof-of-work
- If there is space for 1000 fruits per block → occupy 8% of a 1MB block
- This would get miners to get their rewards 1000x faster → days instead of years

Thank you for your attention!