ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Lecture 23-24: Nakamoto Consensus

CS 539 / ECE 526

Distributed Algorithms

Announcements

- PS 5 (last PS) graded, regrade due today
- Final exam
  - Wednesday May 3rd, Canvas, online
  - 90 min within 9 am to 11 am
  - Covers all materials, focus on second half
- Next Monday: final review, submit questions
- ICES evaluation

Bitcoin

- Whitepaper by Nakamoto in late 2008
- Deployed in January 2009

- An ingenious and unconventional solution to BFT replication
  - A permissionless model
  - Proof of work (PoW) and longest chain

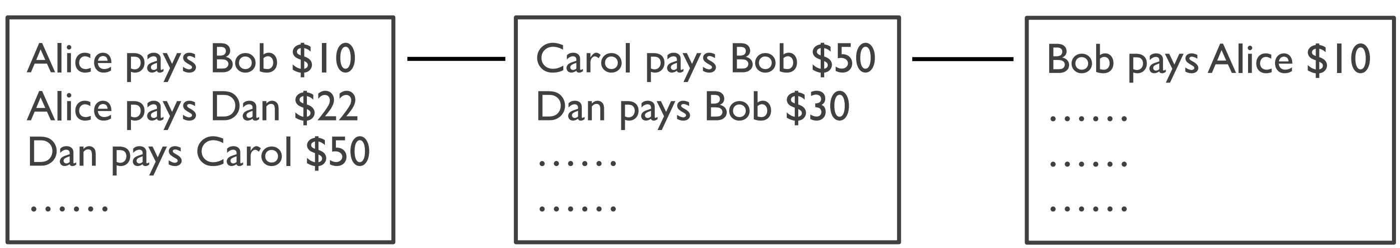
- An ingenious application of BFT replication

Bitcoin Transactions

- Values = blocks of transactions

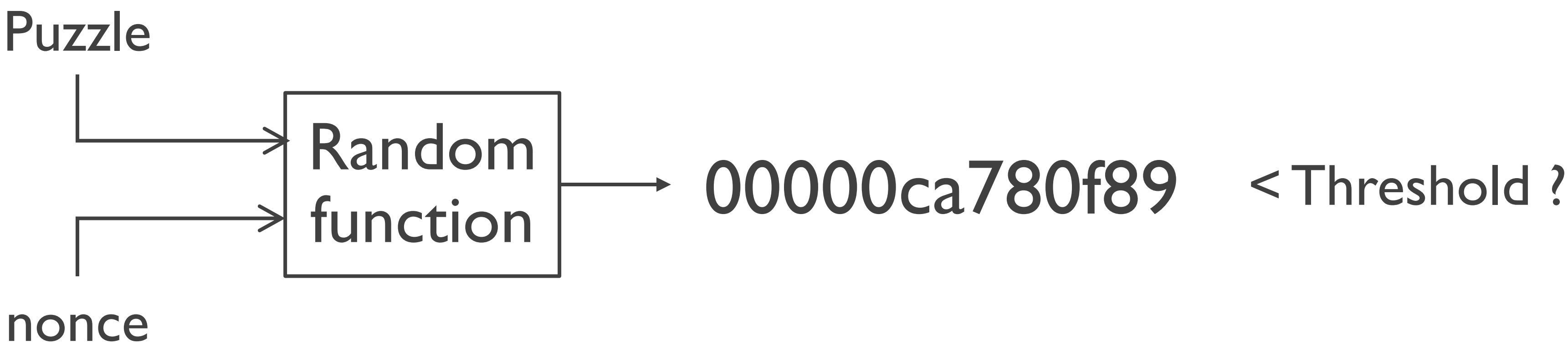
- “I, Alice, pay Bob \$10” signed by Alice
- “I, pk_A, pay pk_B \$10” signed with sk_A

- Agreement on transaction history == currency

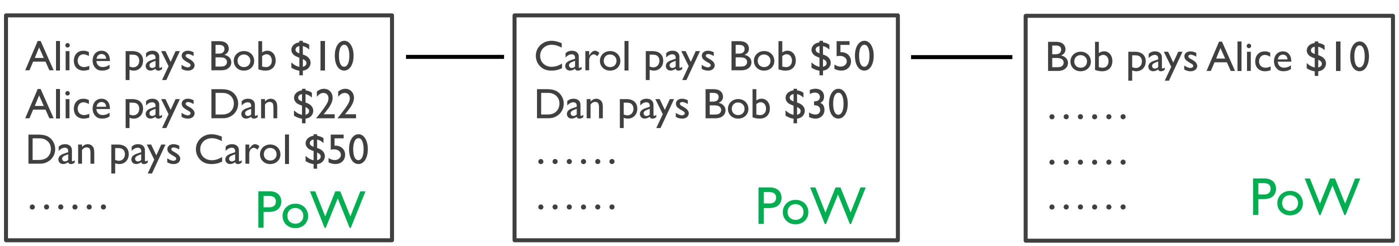


Bitcoin Mining

- Proof-of-work mining: solve hard puzzles
- What are the puzzles?

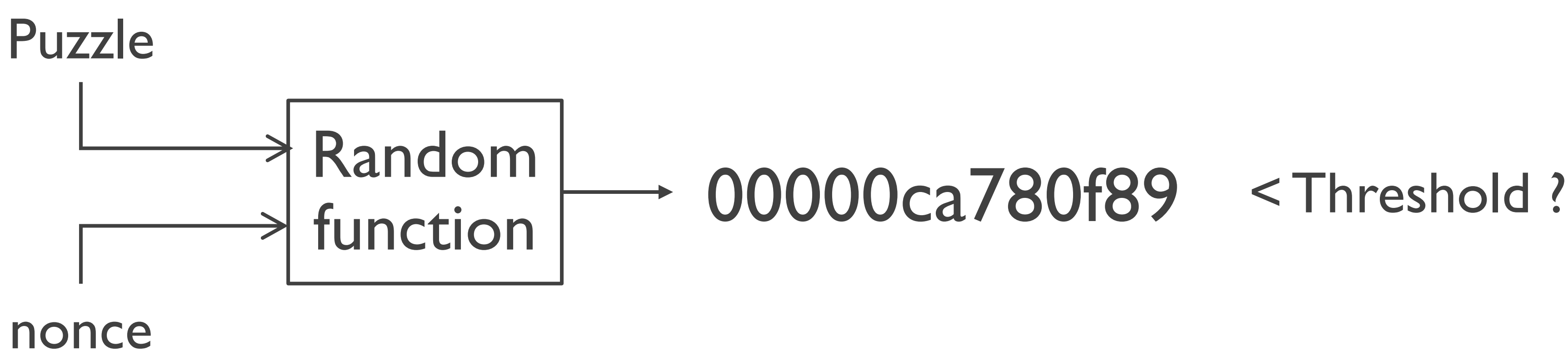


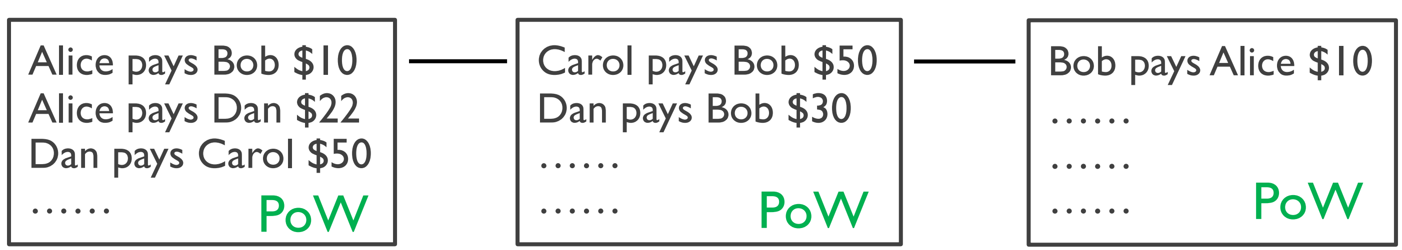
Proof-of-Work (PoW)



Bitcoin Mining

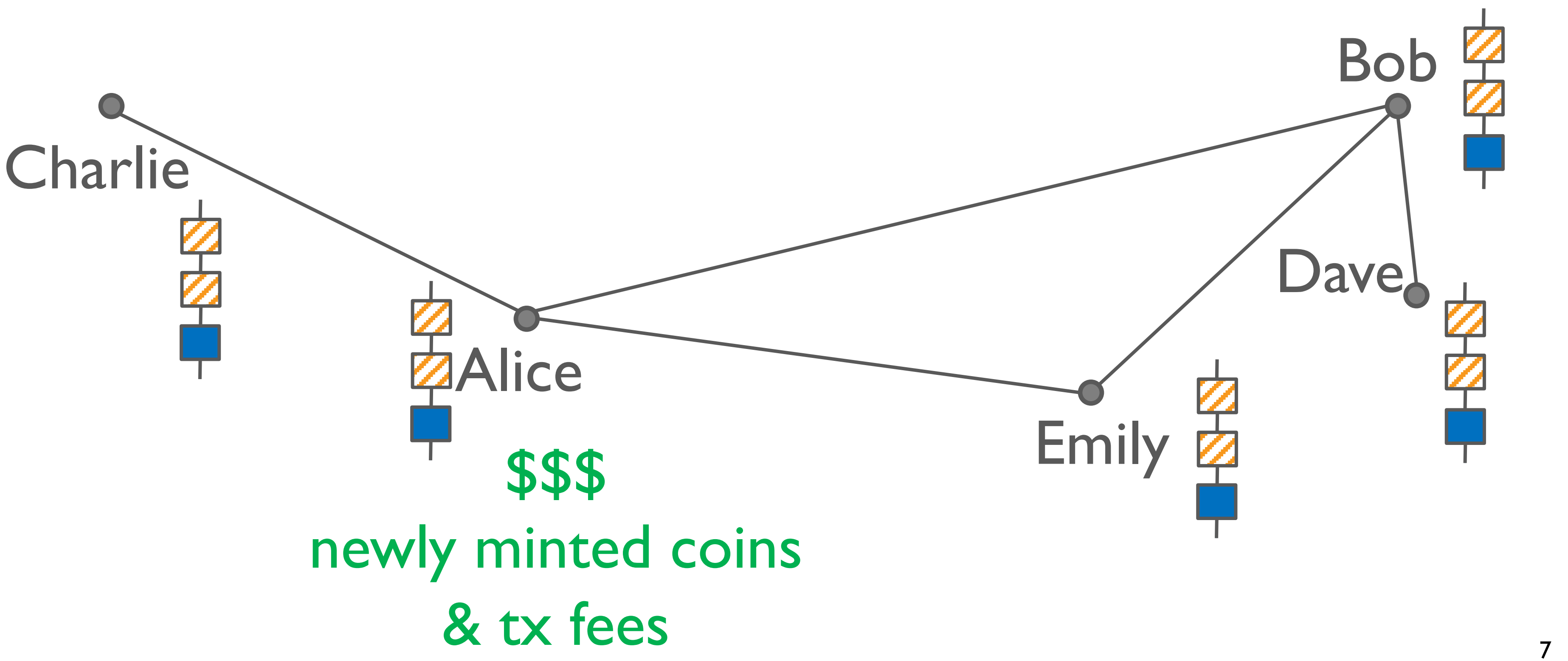
- A succinct representation of the ledger
- Puzzle = Hash(prev block) || Hash(new txs) || pk





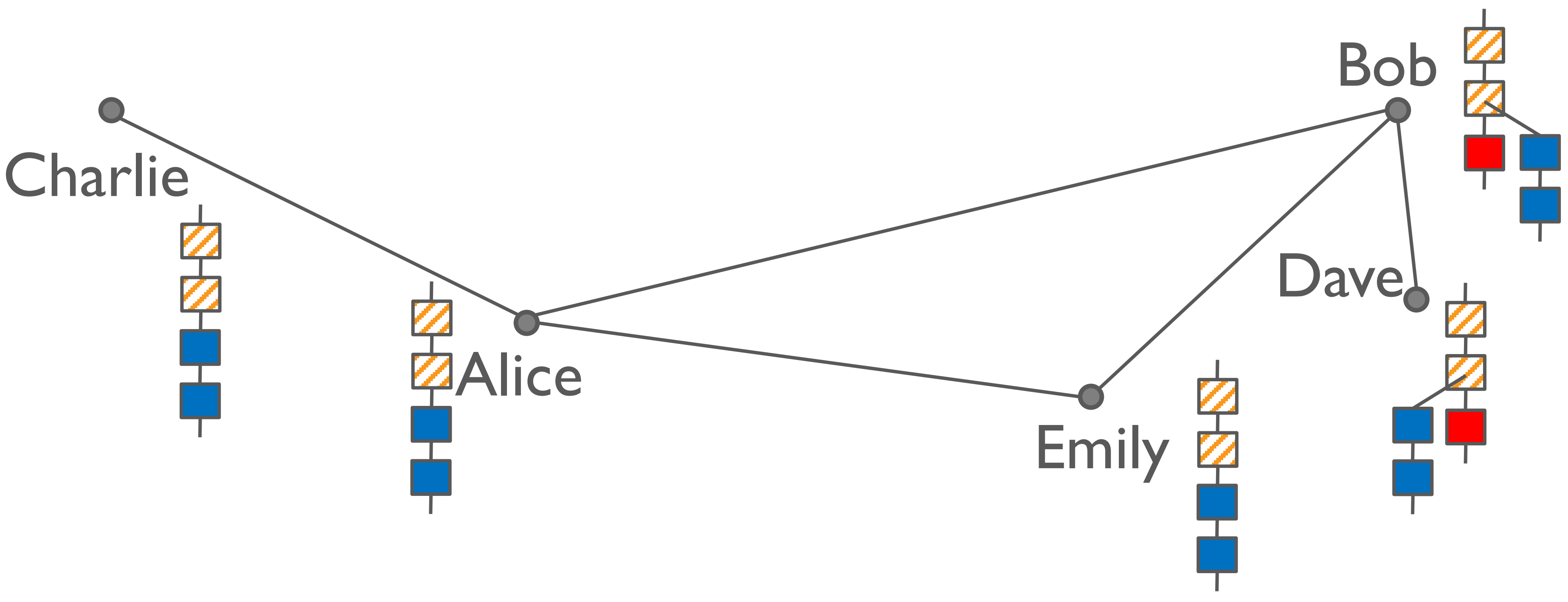
Bitcoin Protocol

- Mine on longest chain & send to all (via a peer-to-peer network)
- Upon mining or receiving a new longest chain, send to all



Bitcoin Protocol

- Mine on longest chain & send to all
- Upon having a new longest chain, send to all
- Commit blocks buried deep



Why does this work?

- Intuitively, a unique longest chain keeps growing faster than all other chains
- Hence, once a block is buried deep, it is unlikely to be "forked off"

- Rigorous proof (sketch) next

Model

- Honest and malicious. No incentives.
- Synchrony: known delay bound Δ
  - Bitcoin cannot handle unbounded delay (partial synchrony or asynchrony)
- Ideal memoryless mining with stable rates
- Attack a target transaction that shows up at a time without adversary's influence

Ideal Memoryless Mining

- Block production follows a Poisson process with rate λ
  - Stable in this work (fluctuate in practice)

- Let T be the time to solve a puzzle

  $\Pr[T > t] = e^{-\lambda t}$ (exponential distribution)

- ρ: ratio of honest mining rate
  - $\rho > 0.5$ (honest majority)

Attacker's Goal

- Attack a target transaction tx that appears at time τ
  - First, make some honest node finalize a block containing tx
  - Then, make another honest node finalize a different block at the same height
- The attacker cannot control τ (why?)

Intuition and Challenges

- Intuition: honest mining power > malicious → honest chain grows fastest, adv can't keep up

- Challenge 1: due to network delay, honest nodes may work against each other
- Challenge 2: need to consider all possible strategies by the adversary
  - Nakamoto only considered a specific strategy

Our Approach

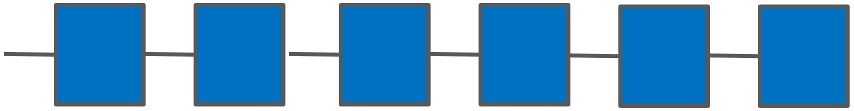
- Challenge 1: due to network delay, honest nodes may work against each other
  - Step 1: assume they don't (magic model)
  - Step 3: reduce to magic model
- Challenge 2: need to consider all possible strategies by the adversary
  - Step 2: find an optimal attack under magic model

Magic Model

- Every honest block is “aware” of all previous honest blocks
  - Essentially zero delay

Optimal Attack under Magic Model

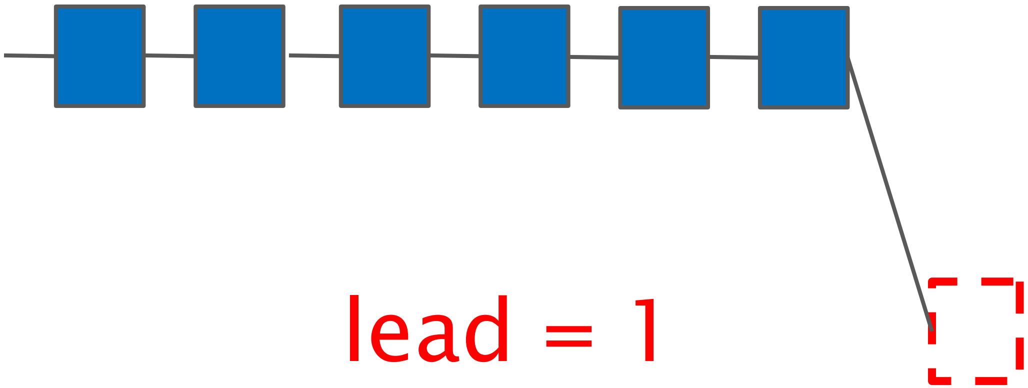
- Before time τ, attempt to build a lead
  - i.e., a longer private chain



lead = 0

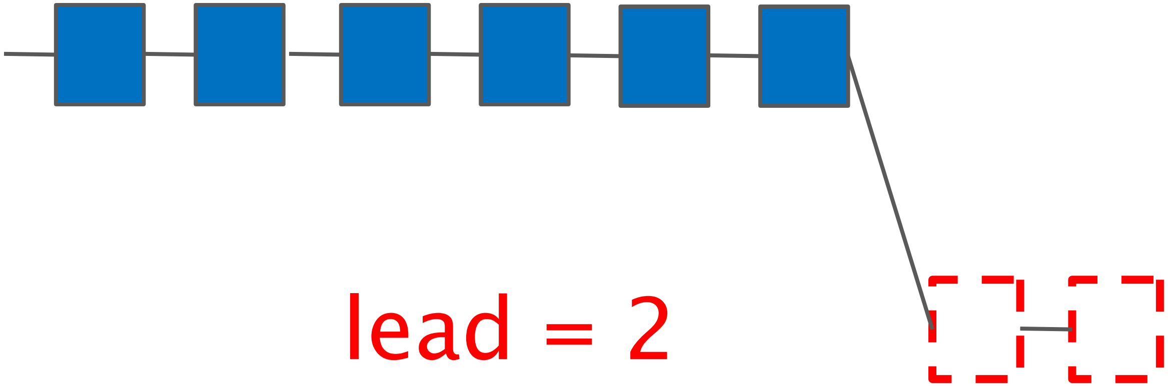
“Lead” Building

- Before time τ, attempt to build a lead
  - i.e., a longer private chain



“Lead” Building

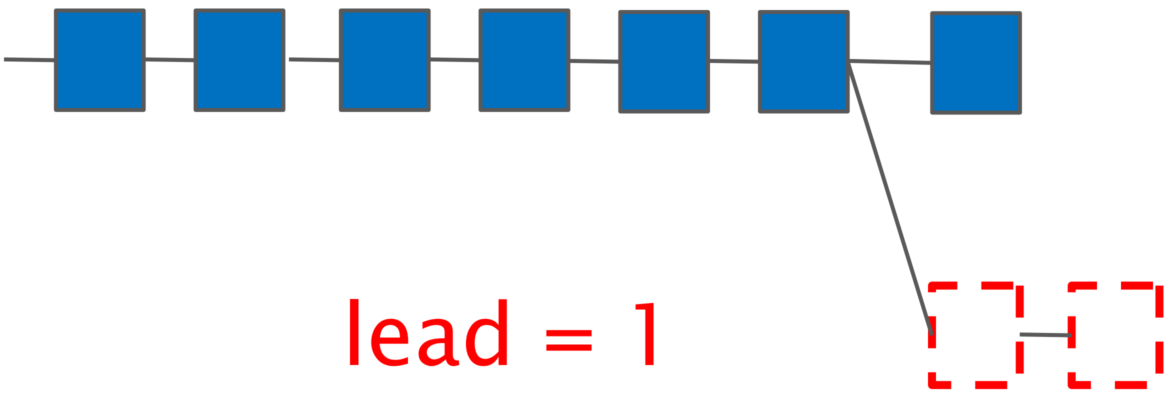
- Before time τ, attempt to build a lead
  - i.e., a longer private chain



lead = 2

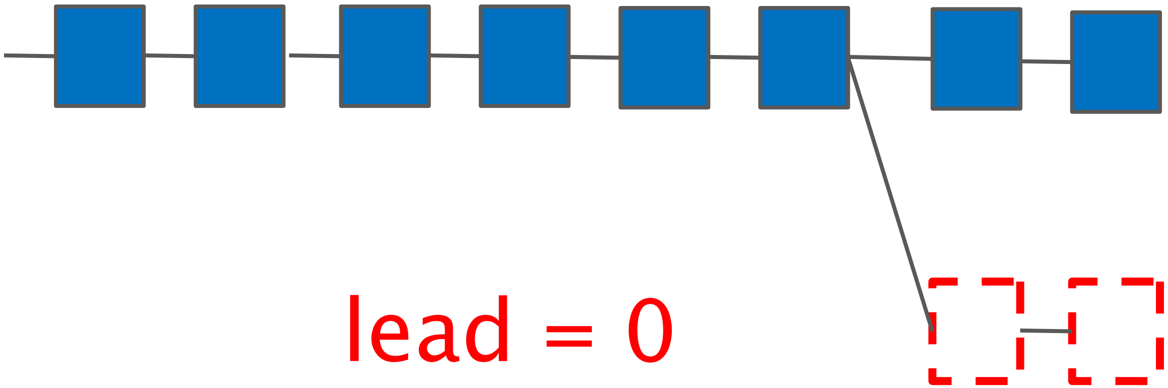
“Lead” Building

- Before time τ, attempt to build a lead
  - i.e., a longer private chain



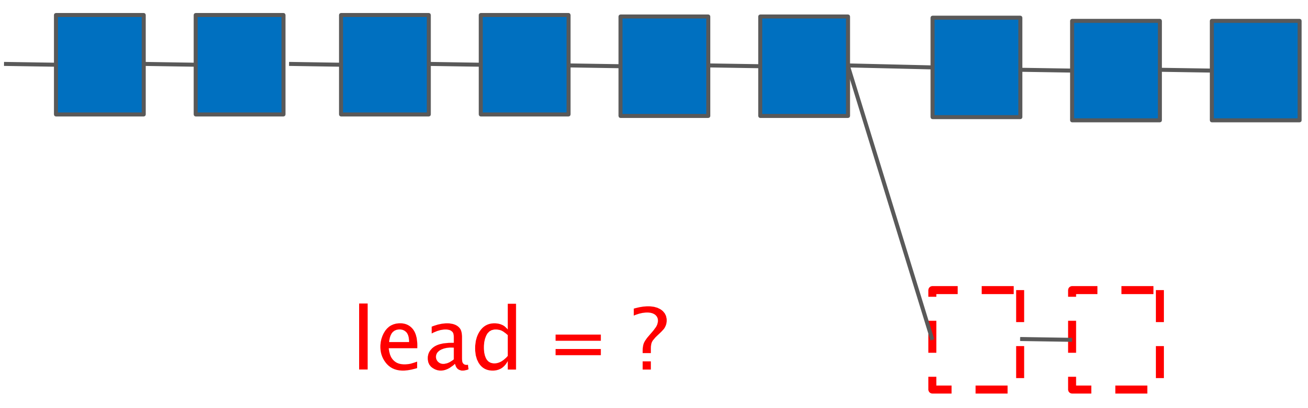
“Lead” Building

- Before time τ, attempt to build a lead
  - i.e., a longer private chain



“Lead” Building

- Before time τ, attempt to build a lead
  - If overtaken, reset (lead = 0)



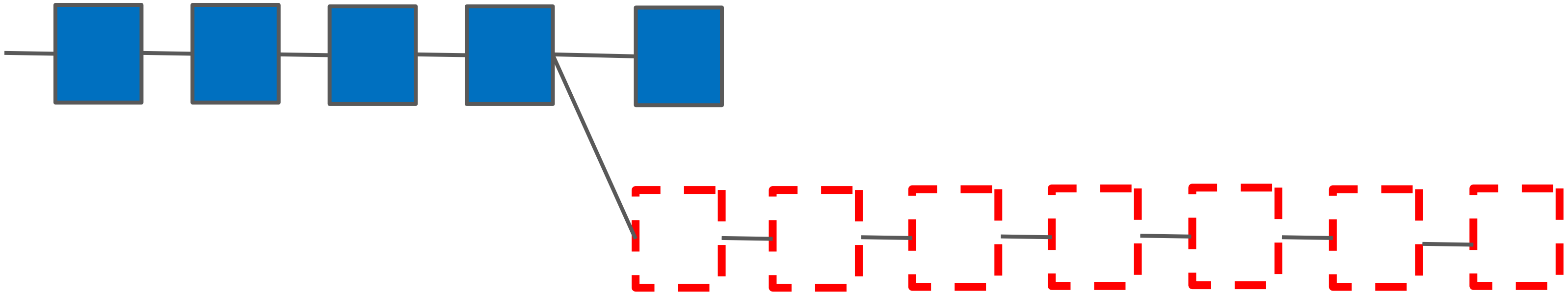
lead = ?

Optimal Attack under Magic Model

- Before time τ, attempt to build a lead
  - If overtaken, reset (lead = 0)
- After time τ, go all-in to mine a private chain without the target tx
- Attacker wins if private chain becomes longer after public chain finalizes tx
- Can prove this attack is optimal

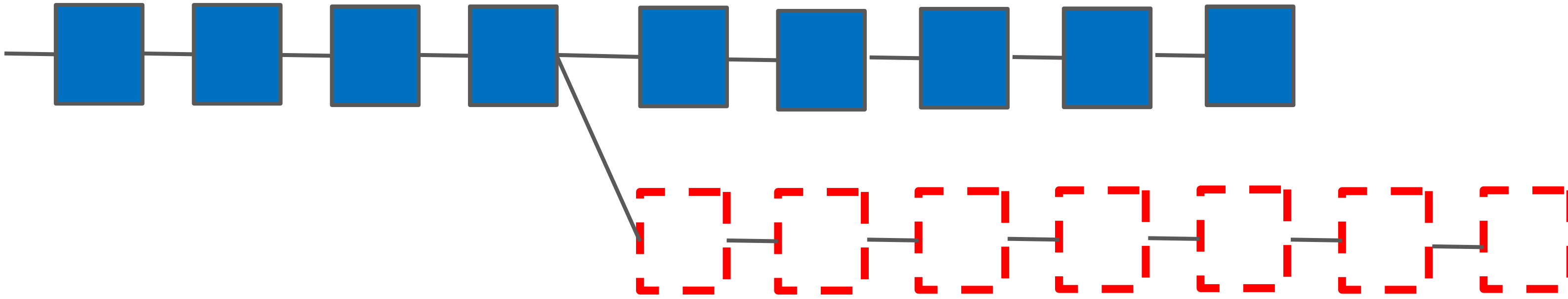
Compute Success Prob

- Attacker wins in three cases only:
  1. At time τ, lead $L > k$



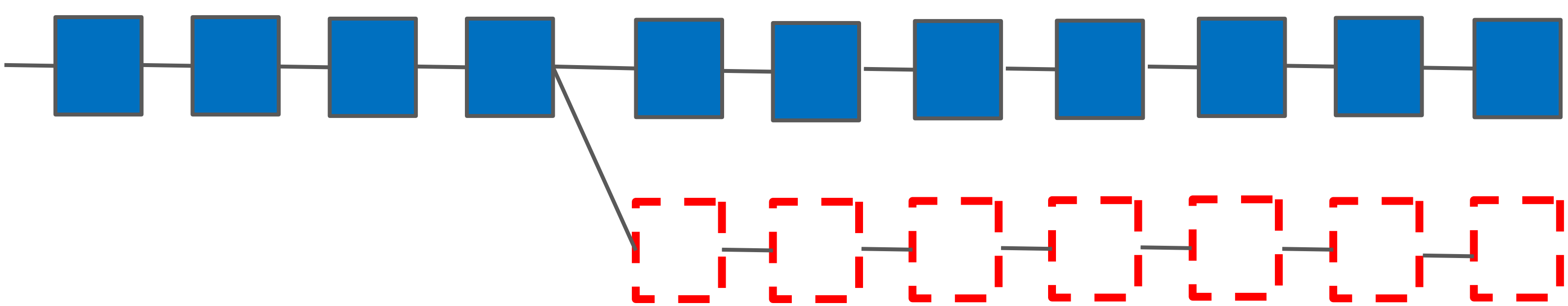
Compute Success Prob

- Attacker wins in three cases only:
  1. At time τ, lead L > k
  2. L <= k but private chain reaches k first



Compute Success Prob

- Attacker wins in three cases only:
  1. At time τ, lead L > k
  2. L <= k but private chain reaches k first
  3. L <= k and public chain reaches k first,
     but private chain eventually catches up



Compute Success Prob

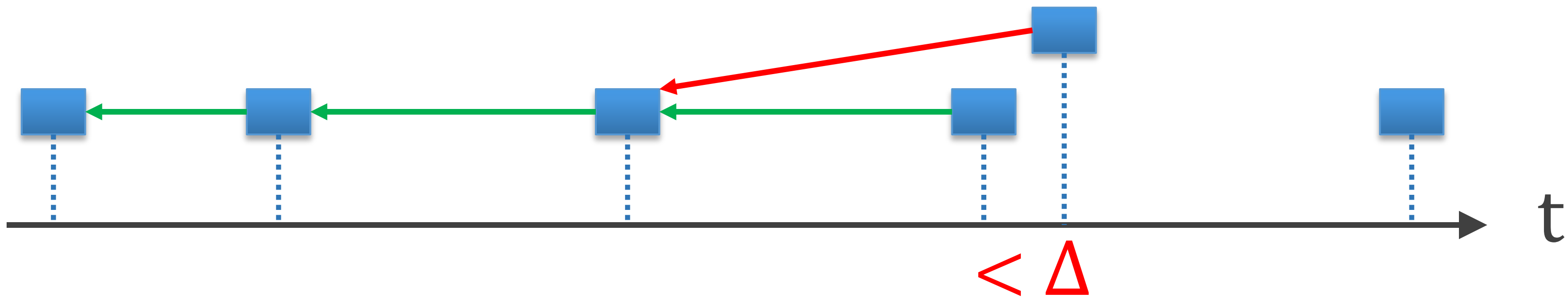
- Attacker wins in three cases only:
  1. At time τ, lead L > k
  2. L <= k but private chain reaches k first
  3. L <= k and public chain reaches k first, but private chain eventually catches up

$$\overline{F}_1(k;p) + \sum_{i=1}^{k} P_1(i;p) \cdot \left(\overline{F}_2(k-i; 2k+1-i, 1-p) + \sum_{j=0}^{k-i} P_2(j; 2k+1-i, 1-p) \cdot \overline{F}_1(2k+1-2i-2j; p) \right)$$

$$\left(2 + 2\sqrt{\frac{p}{1-p}}\right)\left(4p(1-p)\right)^k$$

Reduce to Magic Model

- How can we make sure every honest block is "aware of" all previous honest blocks?
- Give "offending" honest blocks to adversary!

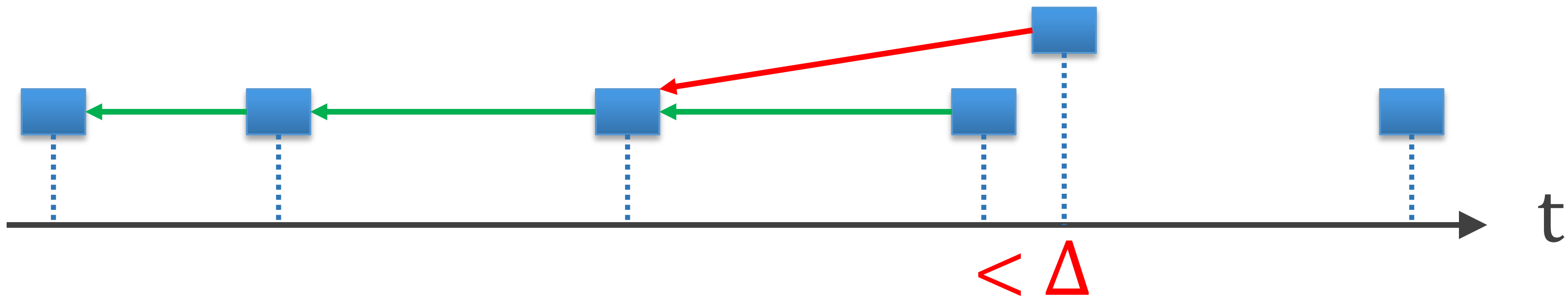


Reduce to Magic Model

- Pr[honest surviving] = Pr[honest] · Pr[lagging]

  $= \rho \cdot e^{-\lambda\Delta}$ (recall $\Pr[T > t] = e^{-\lambda t}$)

- A factor of $e^{-\lambda\Delta}$ loss due to delay
  - Small if Δ is small compared to block interval $1/\lambda$



Proof Summary

- With 0 delay, prove private mining attack is optimal and calculate its success rate
  - This step is precise
- With Δ delay, give some honest blocks to adversary to reduce to zero delay
  - Small precision loss if Δ is small

Gaps to Practice

- Ignored difficulty adjustment
  - An adversary's private chain grows just as fast
  - Use greatest work instead of longest chain
- Incentives
  - Arguably, no node is truly honest
  - Selfish & profitable strategies exist but not pursued

Guidance to Practice

- How shall we tune Bitcoin parameters?
- Nakamoto picked very conservative/ slow parameters with orphan rate < 1%
  - Is it really a bad choice?
- Shall we increase block size or block rate?
  - Same effect for throughput
  - Higher block rate gives better latency
- In fact, worth decreasing block size (Ethereum)
  - Cannot push this indefinitely, will eventually lead to too many “orphans” and break security

Many More Interesting Questions

- Timing model?
- Signatures?
- Latency?
- Communication?
  - Does peer-to-peer help or hurt?
- Roles of PoW?
- What does it mean to be permissionless?

Model of Bitcoin

- Timing model
  - Secure under synchrony
  - Insecure under partial synchrony or asynchrony
  - Some model in between? Major open problem

- Authentication
  - Did not use signatures in its consensus
  - Yet overcome the 1/3 fault bound

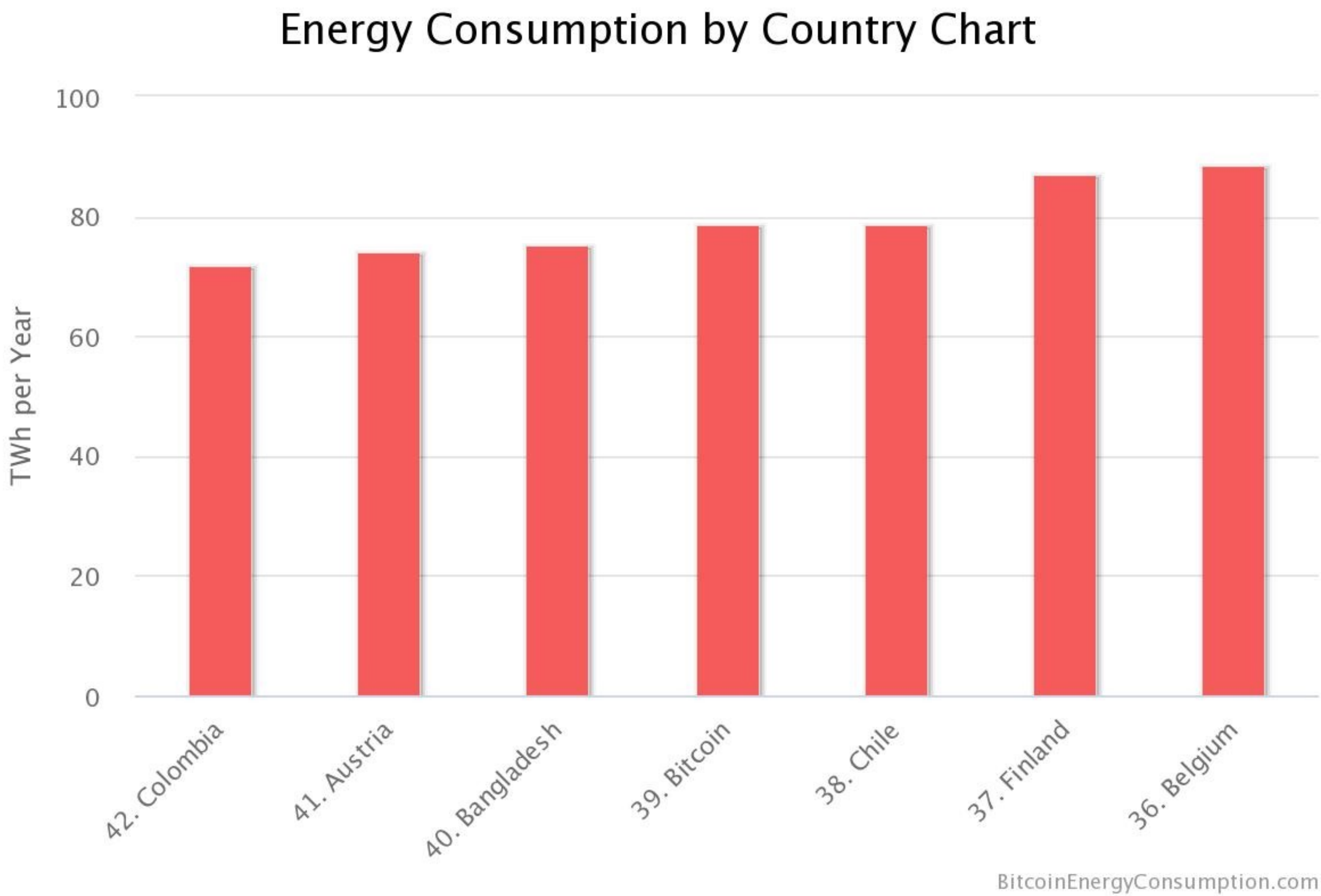
Efficiency of Bitcoin

- Very high latency
  - For exp(-k) error probability, wait for k blocks
  - Blocks arrive slowly

- Communication: O(ndB)
  - n: number of nodes
  - d: number of neighbors
  - B: block size (1MB)
  - Without peer-to-peer, it would have been $O(n^2)$

- Enormous energy efficiency

Bitcoin Energy Consumption



Roles of PoW

- Sybil resistance
- Leader election
  - Rate limiting
- Make equivocation hard
  - Somewhat resembles signatures

Summary

- Nakamoto consensus is a synchronous (!?) Byzantine fault tolerant SMR
  - With rigorous proof
- New application of BFT
- New model: permissionless (and more)
- New technique: PoW & longest chain