Can open decentralized ledgers be economically secure?

Jacob Leshno, Chicago Booth
Elaine Shi, CMU
Rafael Pass, Cornell Tech

CHICAGO BOOTH
The University of Chicago Booth School of Business

Open Distributed Ledger – Why?

Financial ledgers are an indispensable part of a modern economy

Typically provided by a trusted party

Visa, Swift, PayPal, FedNow, ACH, etc…

But, global 2021 revenue of payment providers was $2.1T USD

- global GDP was $96.5T USD

Limited competition

- Strong network effects
- Antitrust litigation and regulation

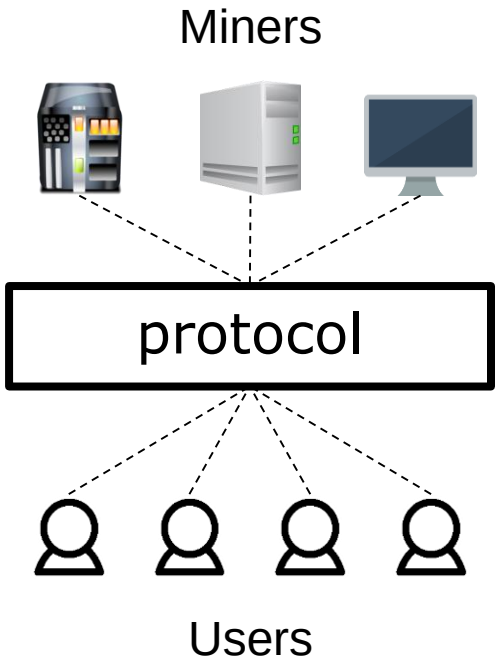




Open Distributed Ledger – Why?

Open decentralized ledgers (Nakamoto 2008) show an alternative

- Open source software
- Operated by anonymous miners
- Free entry of miners protects users from monopoly pricing (HLM ReSTUD 2021)



Open Distributed Ledger – Why?

Open decentralized ledgers (Nakamoto 2008) show an alternative

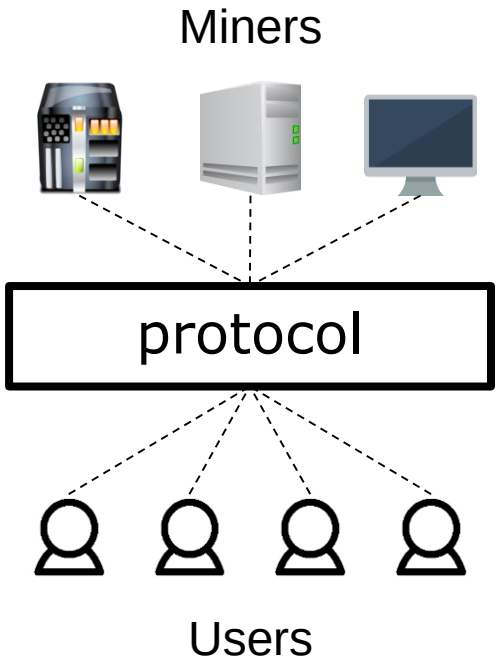
- Open source software
- Operated by anonymous miners
- Free entry of miners protects users from monopoly pricing (HLM ReSTUD 2021)

But concerns about security and energy consumption

- Many chains were attacked

Can a decentralized ledger be economically secure?

Is security at odds with “decentralization”?



Questions and Goals

Can an open decentralized ledger be economically secure?

Definition of economic security

Connecting distributed systems and economic theory

Evaluate security in Bitcoin

Level of mining rewards needed to secure the system?

Can another protocol be open and secure?

Is free entry of miners/nodes at odds with security?

How to design a secure open protocol?

Related Work

In economics:

Kroll et al. (2013), Budish (2018), Carlson et al. (2019), Garratt and van Oordt (2018), Biais et al. (2019), Chiu and Koeppl (2019), Fanti et al. (2019), Gandal and Gans (2019), Auer (2019), Moroz et al. (2020), Halaburda et al. (2022)

In computer science:

Gray (1978), Pease, Shostak, Lamport (1980,1982), Fischer, Lynch, Merritt (1986), Fischer, Lynch, Peterson (1985), Lamport (1998), Gilbert and Lynch (2002)

Barak et al. (2005), Nakamoto (2008), Garay, Kiayias, Leonardos (2015), Pass, Seeman, Shelat (2017)

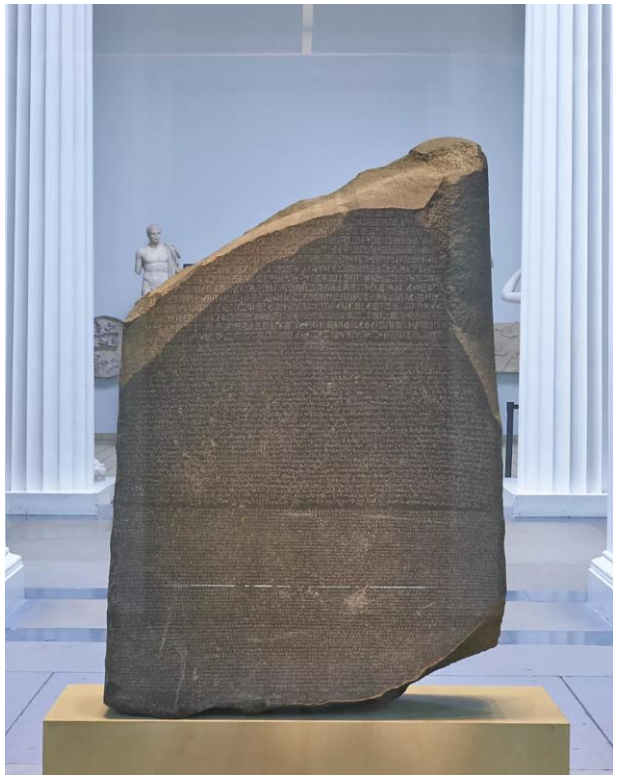
Side note: I taught class on economics of distributed systems, email me for materials

CHICAGO BOOTH
The University of Chicago Booth School of Business
CONSENSUS AND ECONOMICS (33916)
Leshno, Jacob

Building a digital ledger

If we have a big stone in the middle of the village (public write-only ledger), anyone can transact and verify validity

- To make a transfer, write to the ledger

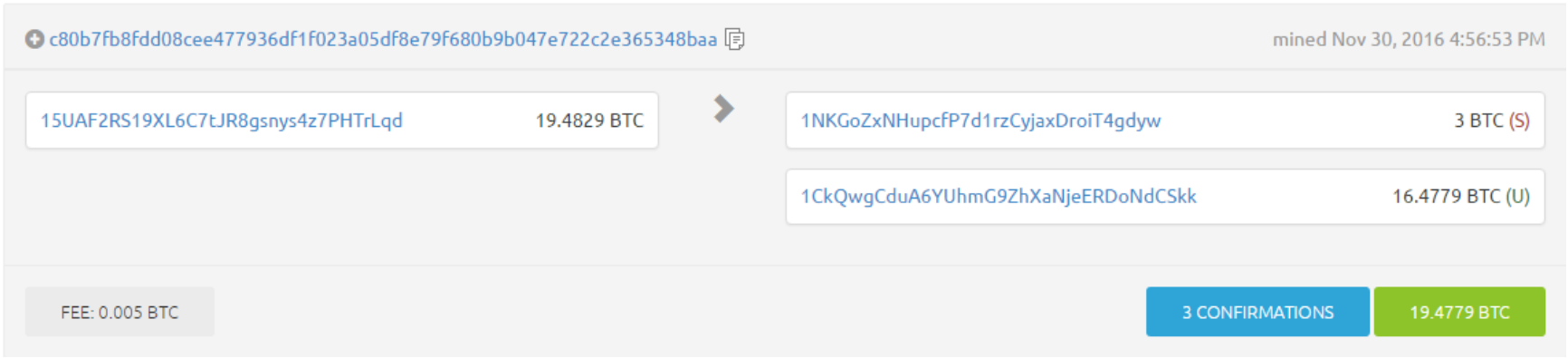


Cryptography can ensure a legal ledger

- Public-key signature proves ownership
- Open ledger proves accounting rules hold

A Bitcoin Transaction

		Y	3	btc
X	19.5 btc	Z	16.4	btc
		Fee	0.1	btc
cryptographic signature				



Building a Digital Ledger

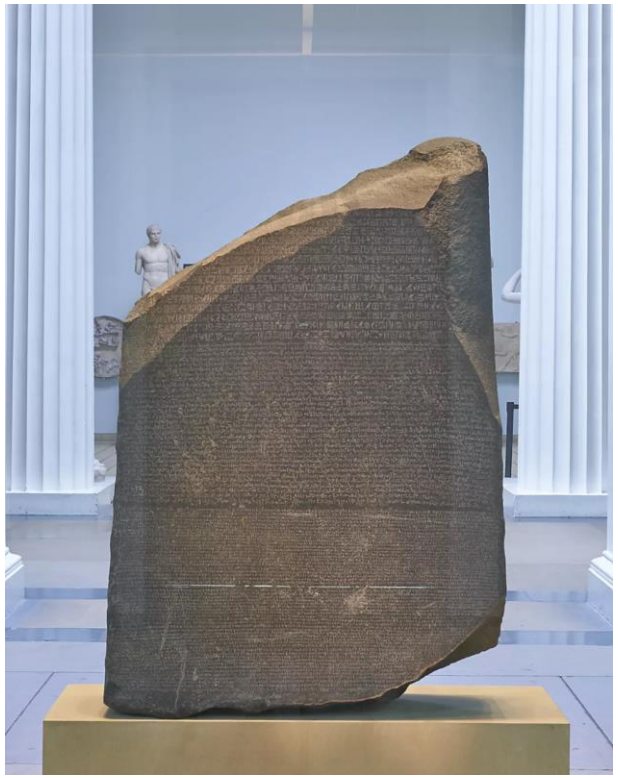
If we have a big stone in the middle of the village (public write-only ledger), anyone can transact and verify validity

- To make a transfer, write to the ledger

Cryptography can ensure a legal ledger

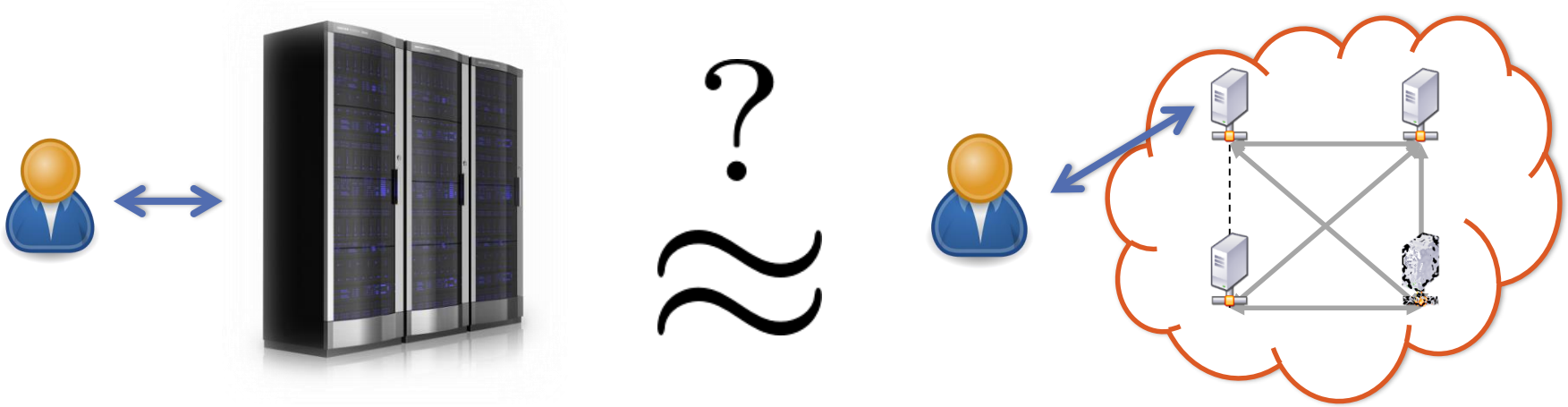
- Public-key signature proves ownership
- Open ledger proves accounting rules hold

But how do we make sure we all see the same digital ledger?



The Consensus Problem

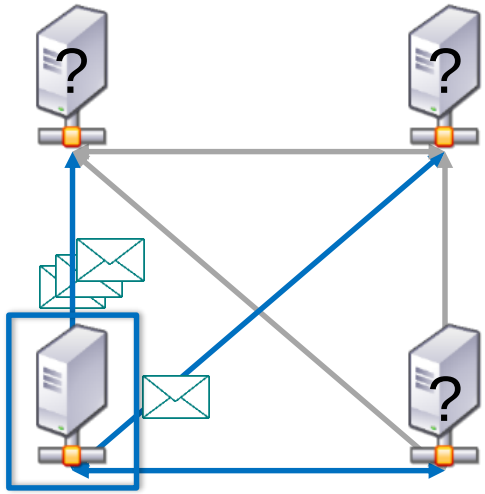
Can a collection of nodes act as one?



Distributed Systems Terminology

Protocol specifies a strategy

- Each node observes its local state
  - Received messages, past actions
- Can send messages, wait, finalize/decide
- Incomplete information about other nodes



Protocol should satisfy:

Consistency (safety) – if an honest node finalized a record, any other honest node agrees with the record

Liveness – new records become finalized

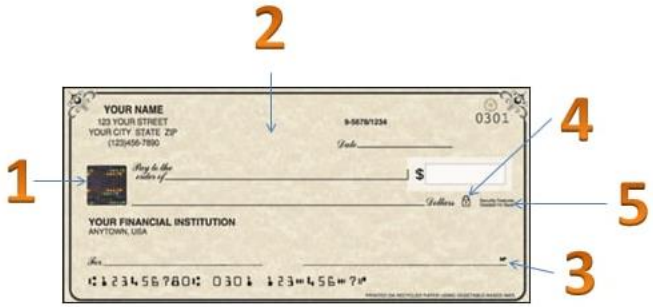
Economic Security

Merchant needs to verify whether a payment will recognized by others

- Merchant wants to use the balance to pay employees etc.
- E.g., Will the bank cash the check?



Secure if merchant can follow protocol to guarantee *others (bank)* recognize the transaction



Economic Security

Merchant needs to verify whether a payment will recognized by others

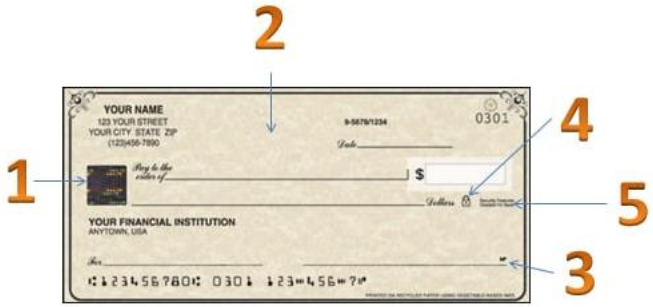
- Merchant wants to use the balance to pay employees etc.
- E.g., Will the bank cash the check?



Secure if merchant can follow protocol to guarantee *others* (bank) recognize the transaction

This is consistency!

- Merchant finalized transaction
  ⇒ any honest node finalized transaction



Definition for Economic Security

Merchant follows the protocol to check if record is finalized

Examine the transaction data; Use a node (terminal/miner); Communicate with other nodes; Impose delay before verifying the transaction

Rational attacker spends resources to deceive merchant

Under assumptions on attacker's capabilities and costs

Consider only costs of attacker (benefits harder to quantify)

Deceiving the merchant ⟺ breaking consistency

Consistency: my node finalized record iff all honest nodes finalize record

The economic security of the protocol is the minimal attacker's cost required to violate consistency.

Profitable Consistency Violations

Attacker can benefit from creating a consistency violation that misleads another party

- **Double spend:**

  Merchant sees a transaction was finalized

  Other nodes see a different ledger without the transaction

  Attacker gets good from merchant, keeps the payment

- **History rewrite:**

  Suppose an option on oil price is recorded on the ledger

  Attacker waits, rewrites the ledger to buy options retroactivly

Agreement on the Protocol

Consistency gives guarantees for honest nodes

That is, assuming other nodes follow the protocol

Necessary technological requirements

Necessary – paper must be unforgeable to be valuable

Not sufficient – unforgeable monopoly bills may have no value



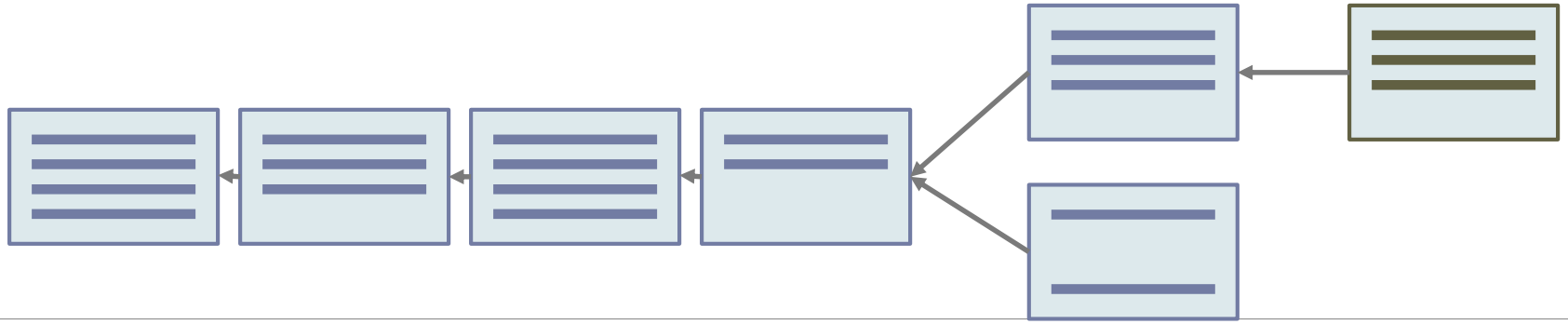
Bitcoin's Nakamoto Protocol

Blockchain as a write-only ledger

- A block is a batch of transactions with a pointer to a previous block
- A blockchain is the link-list of blocks

Random selection of a miner (node) to add a block

- Hash function is like a verifiable random number generator
- Finding a low hash grants permission to add a block
- Number of hashes (random draws) $\approx$ compute power
- Reward R for adding block to chain

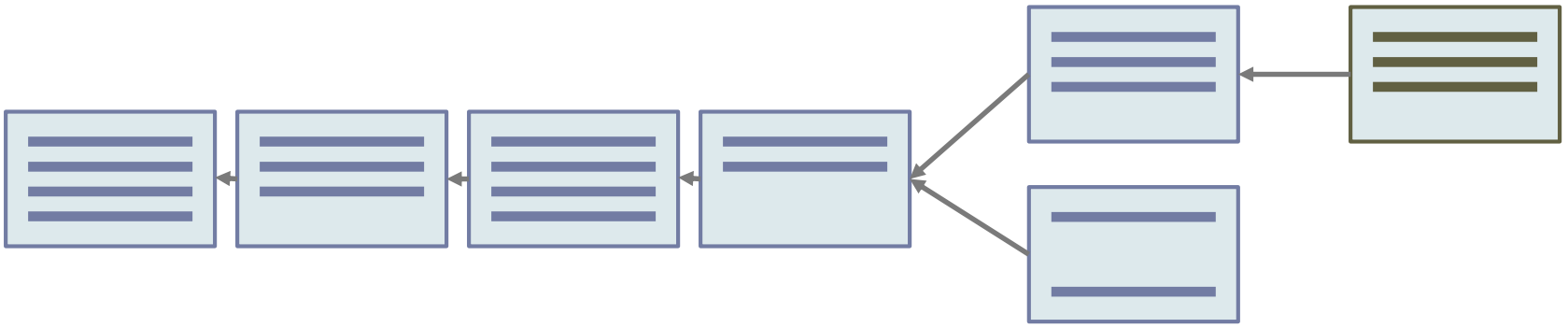


Bitcoin's Nakamoto Protocol

Longest Chain Rule

- Selected miner proposes a block
- Honest miner extend the longest chain they know
- Other miners verify blocks, disregard invalid blocks

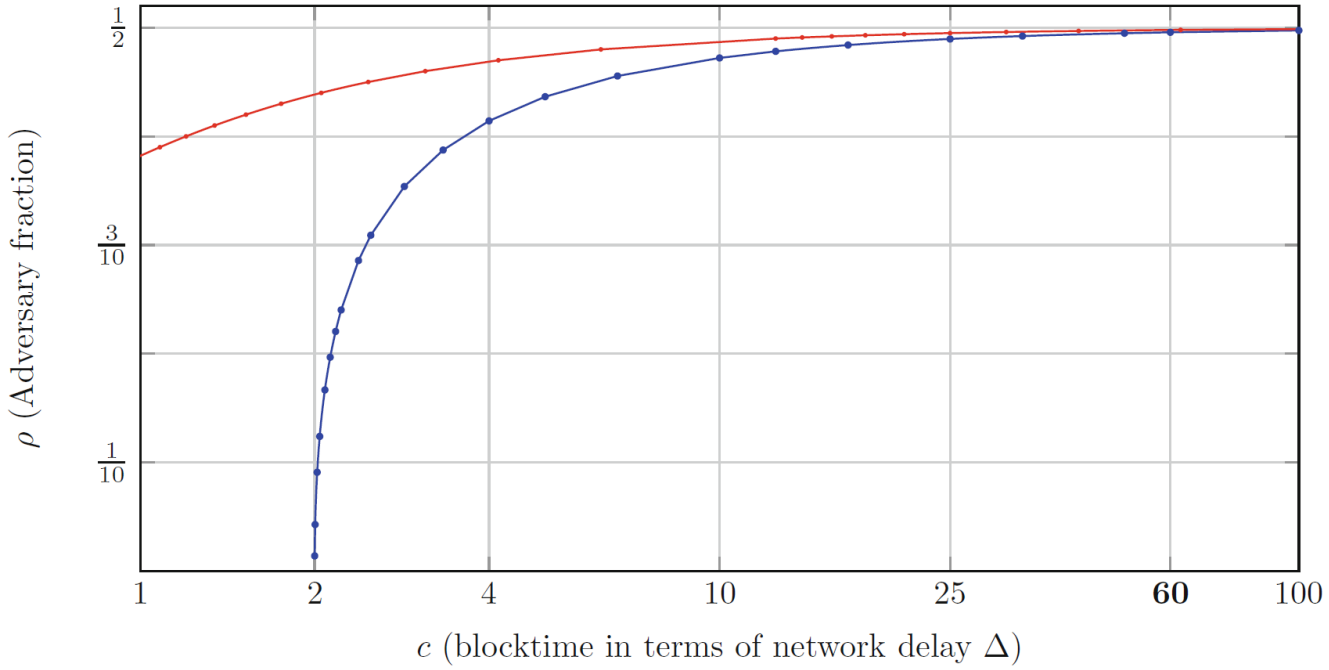
- Network lag can cause honest miners to produce conflicting blocks
- Commonly, blocks are finalized when are k-deep



Bitcoin's Nakamoto Protocol

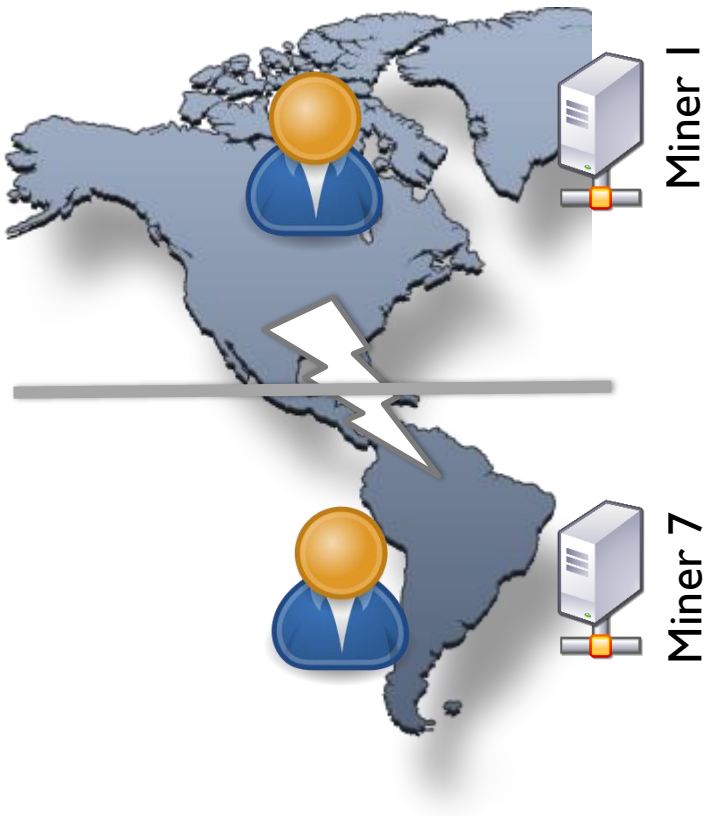
Consistency and liveness if:

- network lags are limited (synchronous communication)
- sufficient majority of honest nodes

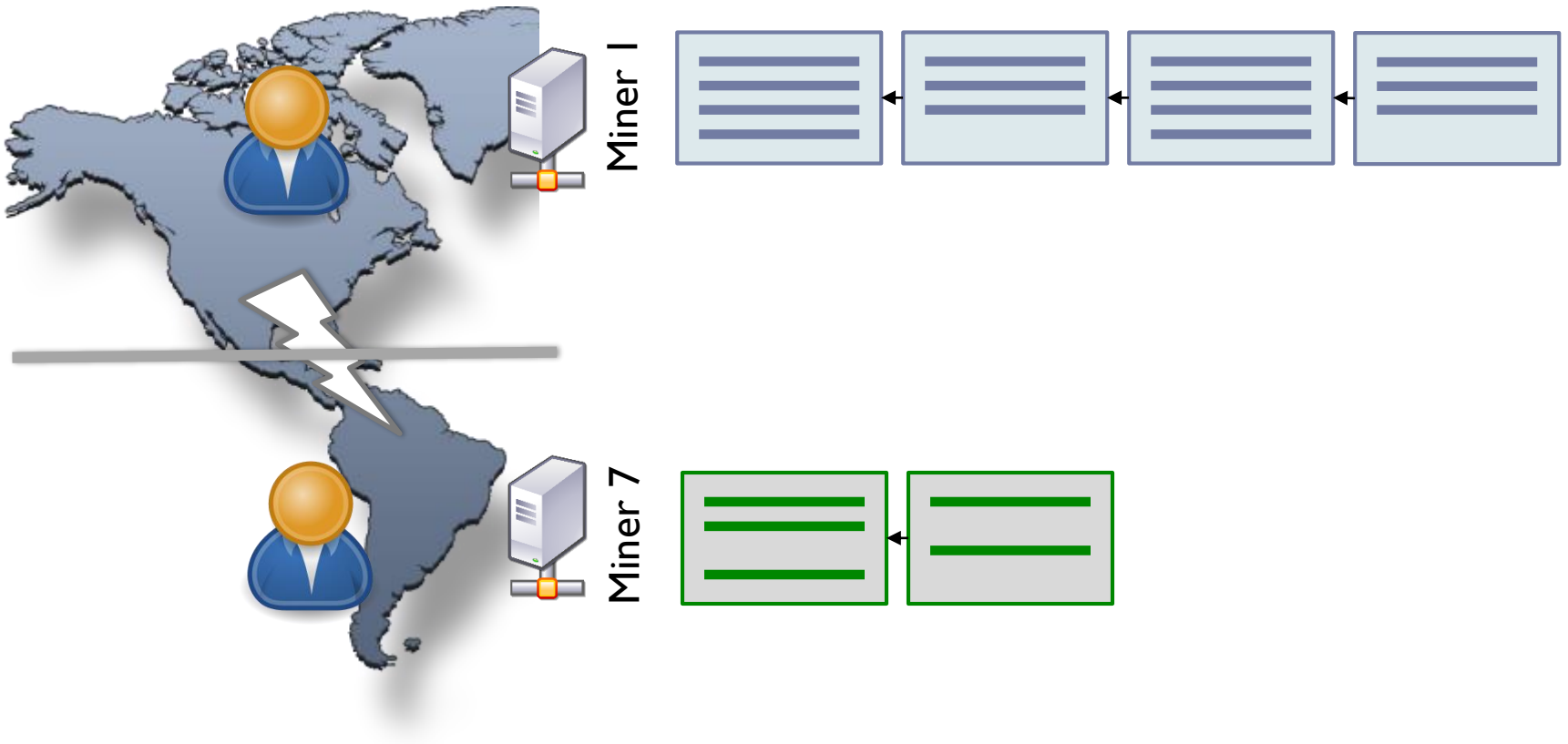


Pass, R., Seeman, L., & Shelat, A. (2017). "*Analysis of the blockchain protocol in asynchronous networks*." EUROCRYPT

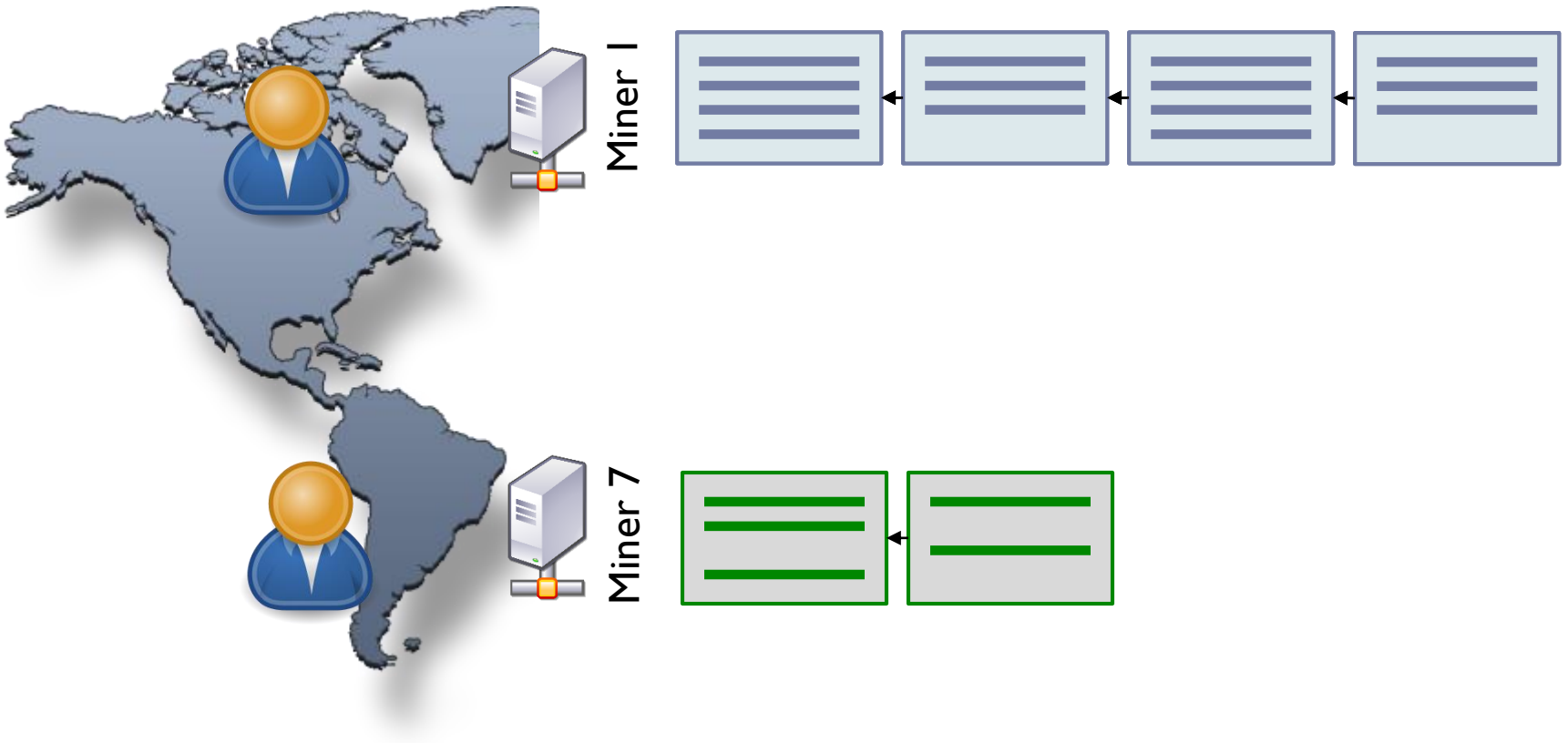
Longest Chain: Network Partition



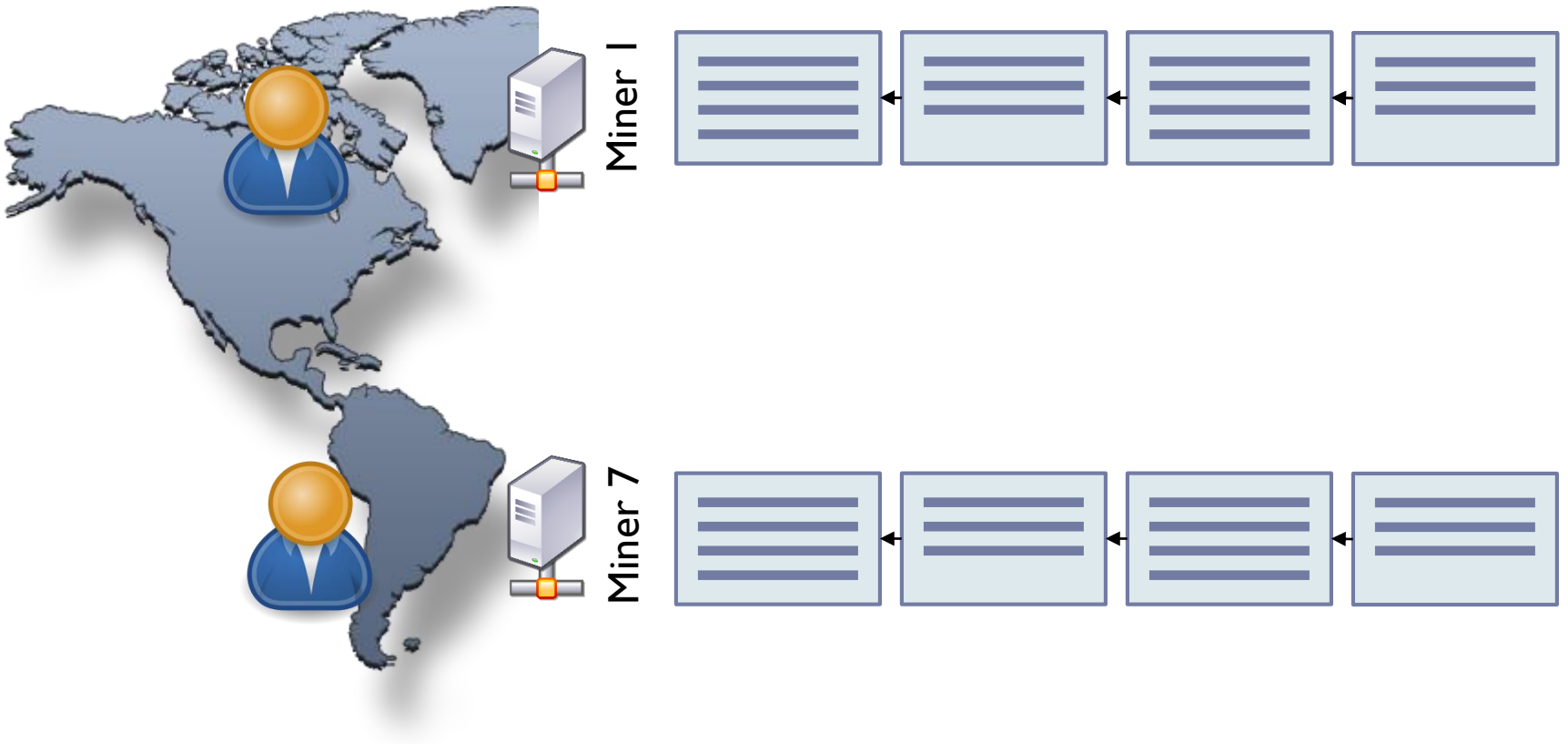
Longest Chain: Network Partition



Longest Chain: Network Partition



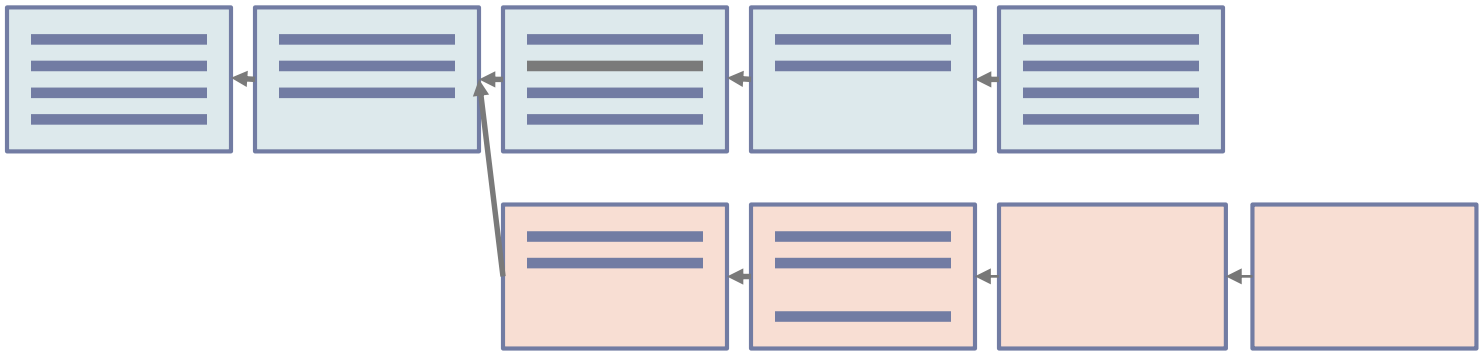
Longest Chain: Network Partition



Attacking Nakamoto

Attacker can violate consistency by creating an attack chain that overtakes the longest chain

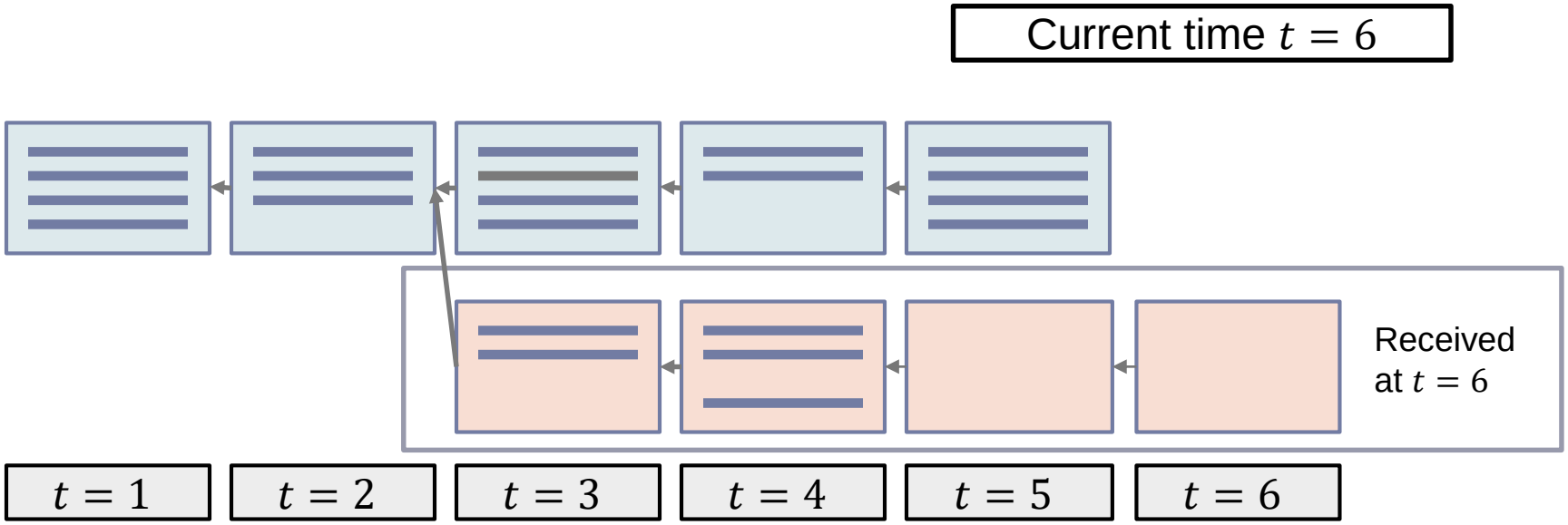
- Attacker “pretends” to have been disconnected
- Requires attacker to mine blocks faster than the honest chain



Attacking Nakamoto

Attacker can violate consistency by creating an attack chain that overtakes the longest chain

- Attacker “pretends” to have been disconnected
- Requires attacker to mine blocks faster than the honest chain



Security of Nakamoto – Rental Model

All miners rent compute power at cost of c per hash

In expectation, D hashes required to mine a block

Merchant receives payment via a transaction, needs to verify others acknowledge its inclusion in the ledger

- Can use a Bitcoin node (miner)
- Can wait till the transaction receives k confirmations (block is k deep)
  - Nakamoto protocol does explicitly specify finality, 6-confirmations is commonly used.
- Can use additional monitoring services

Security of Nakamoto – Rental Model

To rewrite history, attacker creates an alternative longest chain, which becomes consensus chain when public

- Need to create $k + l$ blocks, cost per block $= D \cdot c$
- If attack successful, reward per block $= R$

The attack is costless when successful

- Free entry implies $R = D \cdot c$
- Honest miners break even, attacker becomes an "honest miner"

Success probability can be arbitrarily close to 1

Security of Nakamoto – Rental Model

Protocol cannot penalize attacker without identifying attack

Incentives to prevent attack depend on the *difference* between attackers and honest miners

In line with Budish (2018) and Auer (2019)

Attack is costly only if attacker is distinct from honest miners

Security of Nakamoto – ASICs

Attacker cannot procure mining equipment

- E.g., Mining requires specialized scarce equipment (ASICs)
- Extreme fixed cost (Garratt & van Oordt 2019, Prat & Walter 2021)

If attack successful, miners suffer loss c_A

- Stake in the future of the system
- Lost rewards from orphaned blocks
- Attack makes ASICs less valuable

Miner's marginal cost of mining a block is c_m

Attacker can incentivize miners to create the attack chain

- Miners are rational profit maximizers

Security of Nakamoto – ASICs

Attacker bribes miners to construct an attack chain

Commits to pay $\tilde{R}$ for each block in attack chain

- Attacker collects block rewards from attack chain
- Can commit via a smart contract

Security of Nakamoto – ASICs

Miner's choice	Payoff if Attack successful	Payoff if Attack fails
Attack	$\tilde{R} - c_m - c_A$	$\tilde{R} - c_m$
Honest	$-c_m - c_A$	$R - c_m$

Attacker's cost is $k \cdot \left(\tilde{R} - R\right) \approx 0$

- Non pivotal miners join attack if $\tilde{R} \geq \mathrm{R}$

Tragedy of the commons

What Deters Attacks?

Attacks are detectable

- Agreement that ledger has corrupt records

Community hard fork can negate the attack

- Exchange rate collapse is a form of community response
- Ethereum (no attack) vs. Ethereum Classic (attacked)

Other deterrents:

- Miner concentration (e.g., c_A matters for Visa)
- Logistic/Marketing costs of organizing attack
- Attacker can be prosecuted

Incentives from mining rewards do little to deter attacks

Economic Security in Traditional Payments

In traditional systems, merchants verify transactions by communicating with the bank

Square

Important things to remember with offline credit card processing

When you take offline credit card transactions, you take on the risk of any expired, declined, or disputed payments. Basically, Square Point of Sale can't talk with the banks during offline payment processing, so when you use Offline Mode, you're responsible if any cards are declined. While offline payments are automatically processed when your device has connectivity again, it's important to note that transactions expire if they're not processed within 72 hours.

https://squareup.com/us/en/townsquare/offline-credit-card-processing

Secure Protocol under Synchrony

Nodes "ask" other nodes whether a block is finalized

- Prioritize consistency
- Baking the "community hard fork" into the protocol

Economic security that does not rely on mining rewards

- Does not require mining rewards

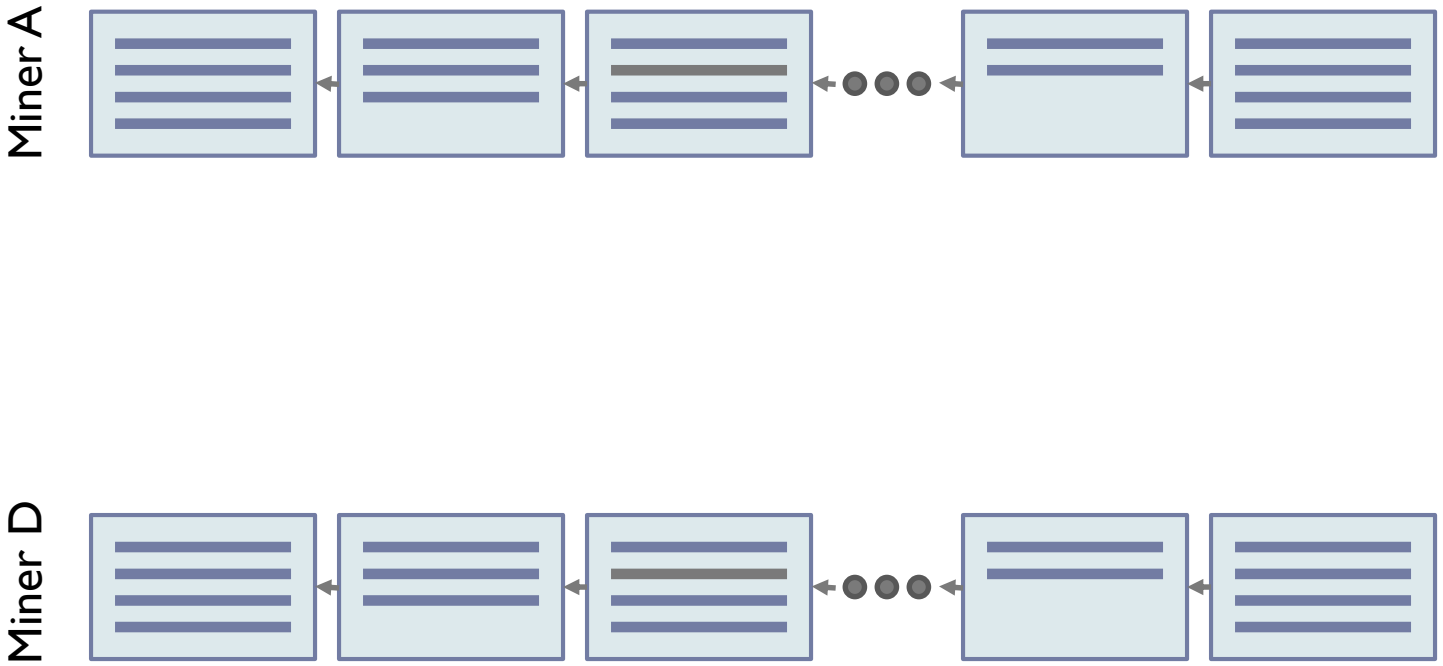
Maintains decentralized properties of Bitcoin's Nakamoto

- Allows free entry of anonymous miners

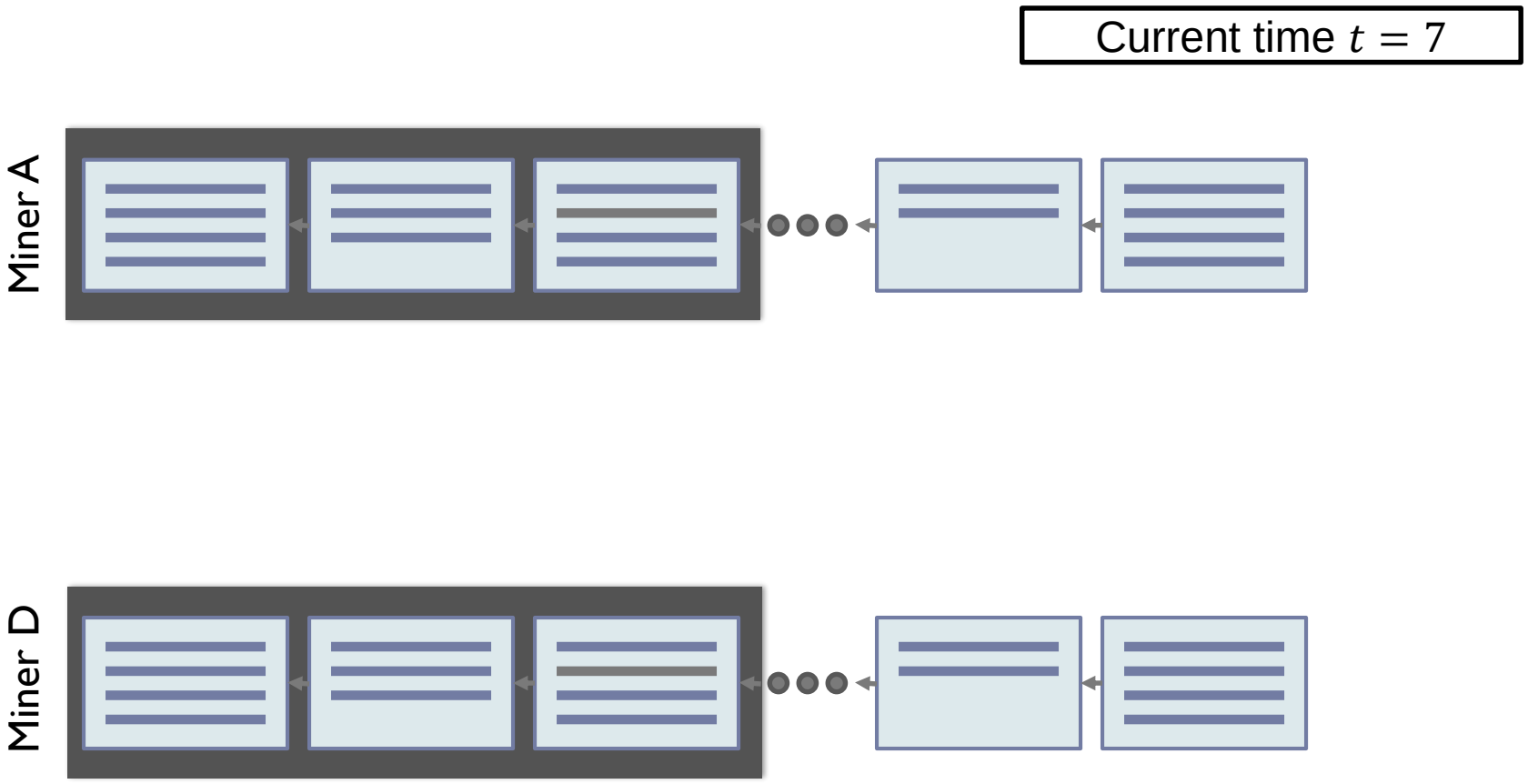
Stubborn Nakamoto is economically secure protocol

Stubborn Nakamoto

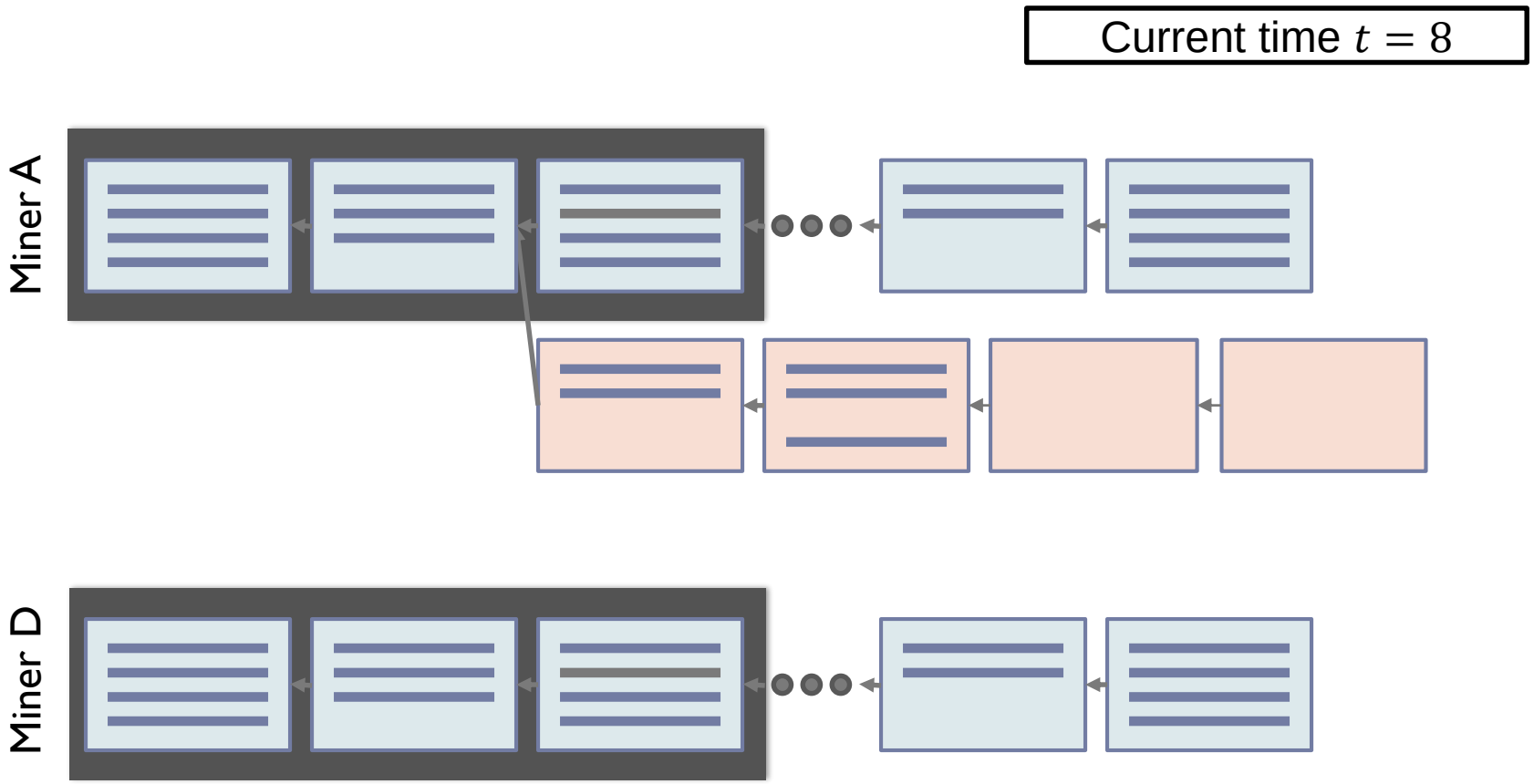
Current time $t = 6$



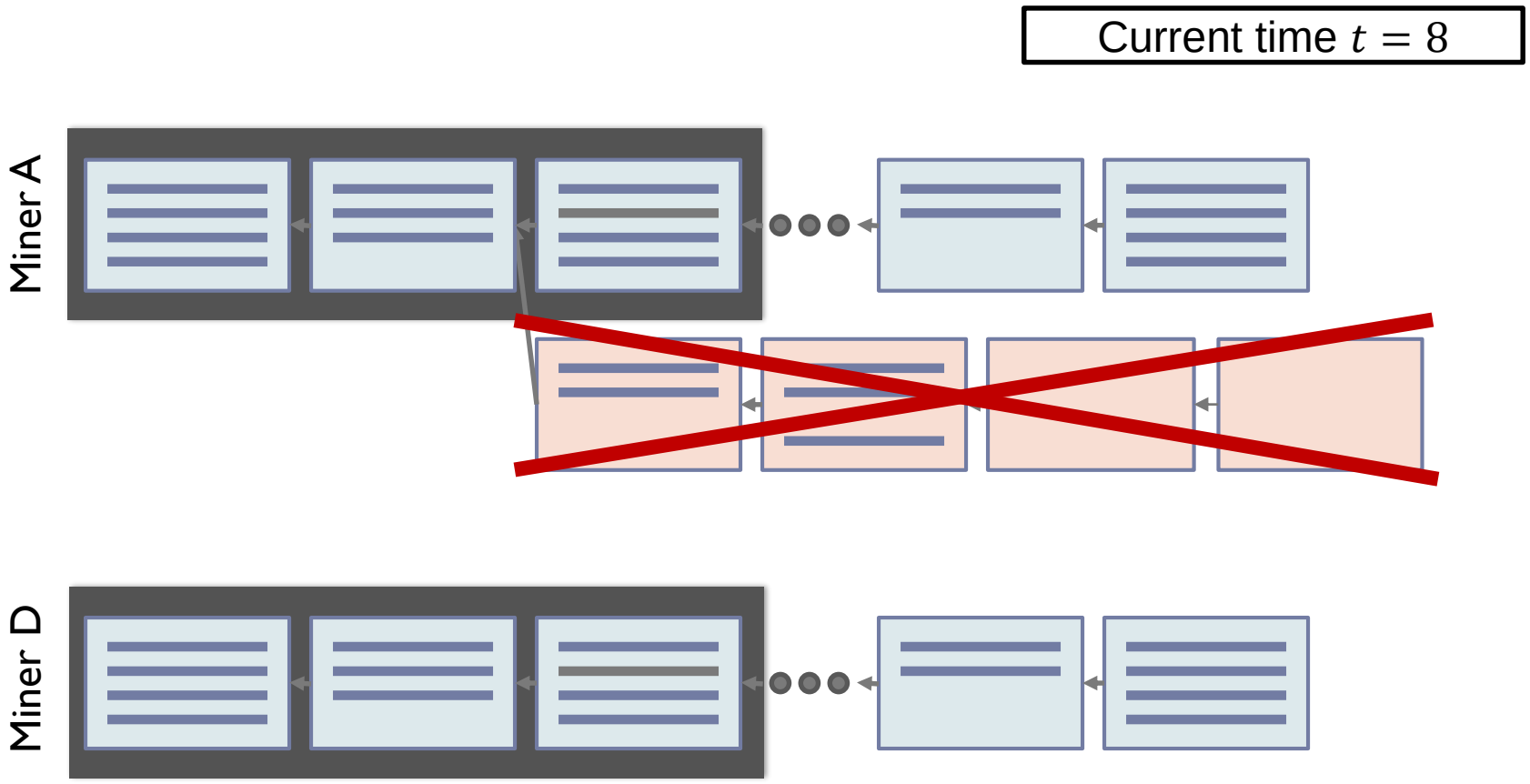
Stubborn Nakamoto



Stubborn Nakamoto



Stubborn Nakamoto



Security for Offline Merchants

Under stubborn Nakamoto, cost is constructing a ledger

- Attacker can produce a parallel ledger, pay the mining costs
- Offline merchant cannot distinguish between the true ledger and the attacker's copy
- Attacker's cost is at most the combined cost of honest miners, i.e. attacker's cost is at most R per block

Staking protocols (PoS) allow individual punishment, can be more efficient

- Nodes can be identified, locked funds can be slashed
- Rewards compensate forgone interest
- Punishment limit is R/r

Conclusions

Building financial rails using open decentralized protocols?

- Many questions remain
- Limited functionality may still be useful (e.g. TCP/IP)
- Long road ahead…

Design guideline: ***never lose consistency***

- Consistency gives economic security
- Violations can be profitable for attackers

Limits of incentives given anonymity

- Mining rewards neither sufficient nor necessary for security
- Tragedy of the commons, detecting attacks
- PoS can provide stronger incentives