Lecture 2

How Bitcoin Achieves Decentralization

Lecture 2.1:

Centralization vs. decentralization

Centralization vs. decentralization

Competing paradigms that underlie many digital technologies

Decentralization is not all-or-nothing

E-mail:

decentralized protocol, but dominated by centralized webmail services

Aspects of decentralization in Bitcoin

1. Who maintains the ledger?
2. Who has authority over which transactions are valid?
3. Who creates new bitcoins?
4. Who determines how the rules of the system change?
5. How do bitcoins acquire exchange value?

Beyond the protocol:
exchanges, wallet software, service providers...

Aspects of decentralization in Bitcoin

Peer-to-peer network:

open to anyone, low barrier to entry

Mining:

open to anyone, but inevitable concentration of power

often seen as undesirable

Updates to software:

core developers trusted by community, have great power

Lecture 2.2:

Distributed consensus

Bitcoin’s key challenge

Key technical challenge of decentralized e-cash: <u>distributed consensus</u>

or: how to decentralize ScroogeCoin

Why consensus protocols?

Traditional motivation: reliability in distributed systems

Distributed key-value store enables various applications:
DNS, public key directory, stock trades …

Good targets for Altcoins!

Defining distributed consensus

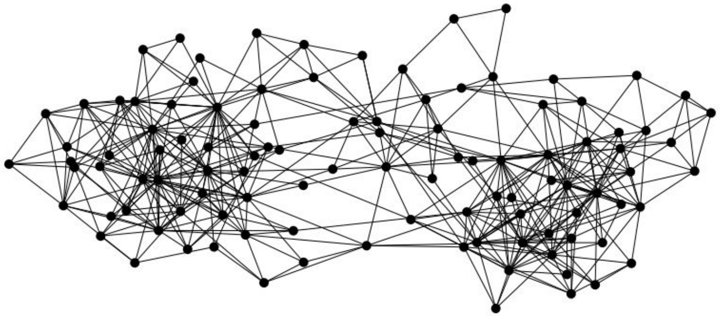
The protocol terminates and all correct nodes decide on the same value

This value must have been proposed by some correct node

Bitcoin is a peer-to-peer system

When Alice wants to pay Bob:
she <u>broadcasts the transaction</u> to all Bitcoin nodes

signed by Alice
Pay to pk_{Bob} : H( )



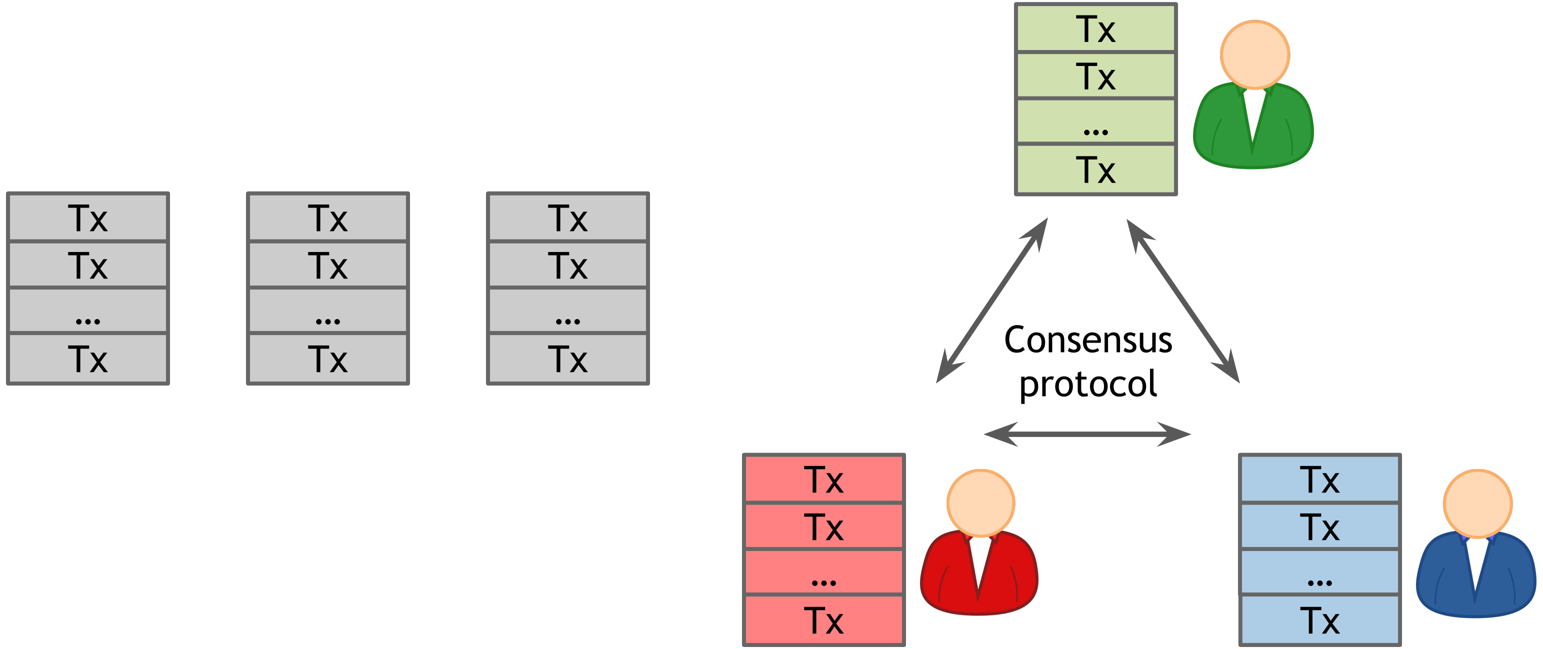
Note: Bob's computer is not in the picture

How consensus could work in Bitcoin

At any given time:

- All nodes have a sequence of blocks of transactions they've reached consensus on
- Each node has a set of outstanding transactions it's heard about

How consensus <u>could</u> work in Bitcoin



OK to select any valid block, even if proposed by only one node

Why consensus is hard

Nodes may crash
Nodes may be malicious

Network is imperfect

- Not all pairs of nodes connected
- Faults in network
- Latency

No notion of global time

Many impossibility results

- Byzantine generals problem
- Fischer-Lynch-Paterson (deterministic nodes): consensus impossible with a <u>single</u> faulty node

Some well-known protocols

Example: Paxos

Never produces inconsistent result, but can (rarely) get stuck

Understanding impossibility results

These results say more about the model than about the problem

The models were developed to study systems like distributed databases

Bitcoin consensus: theory & practice

Bitcoin consensus works better in practice than in theory

Theory is still catching up

<u>BUT</u> theory is important, can help predict unforeseen attacks

Some things Bitcoin does differently

Introduces incentives

- Possible only because it’s a currency!

Embraces randomness

- Does away with the notion of a specific end-point
- Consensus happens over long time scales — about 1 hour

Lecture 2.3:

Consensus without identity: the block chain

Why identity?

Pragmatic: some protocols need node IDs

Security: assume less than 50% malicious

Why don’t Bitcoin nodes have identities?

Identity is hard in a P2P system — <u>Sybil attack</u>

Pseudonymity is a goal of Bitcoin

Weaker assumption: select random node

Analogy: lottery or raffle

When tracking & verifying identities is hard, we give people tokens, tickets, etc.

Now we can pick a random ID & select that node

Key idea: implicit consensus

In each round, random node is picked

This node proposes the next block in the chain

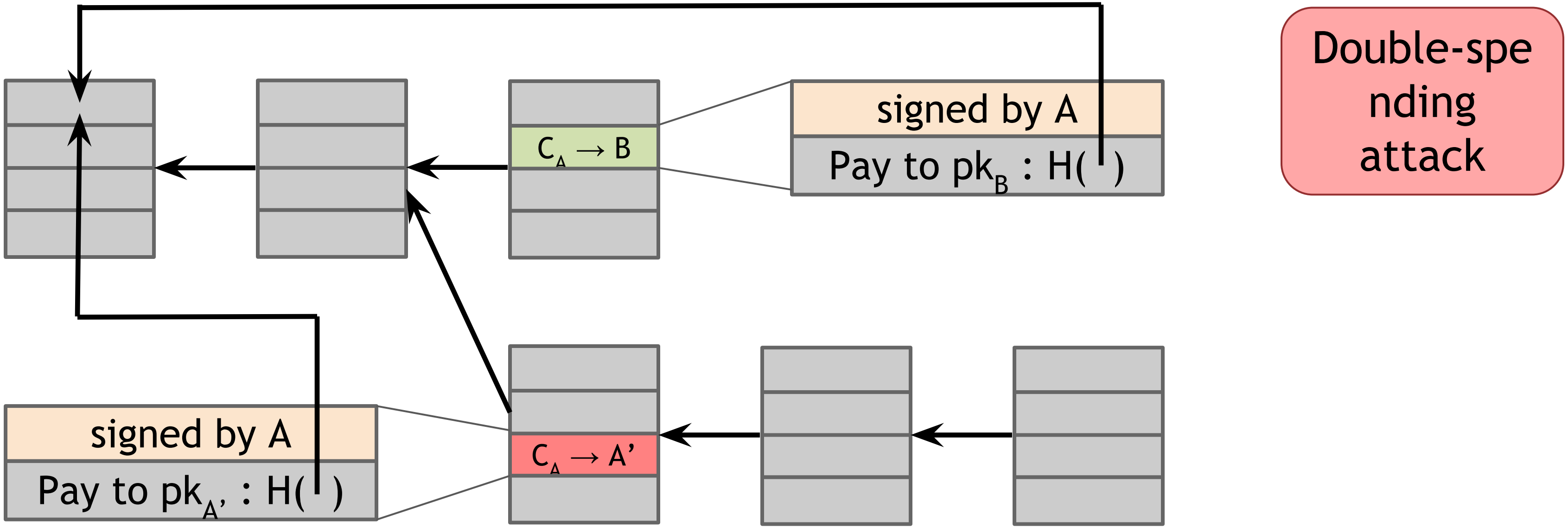
Other nodes implicitly accept/reject this block
- by either extending it
- or ignoring it and extending chain from earlier block

<u>Every block contains hash of the block it extends</u>

Consensus algorithm (simplified)

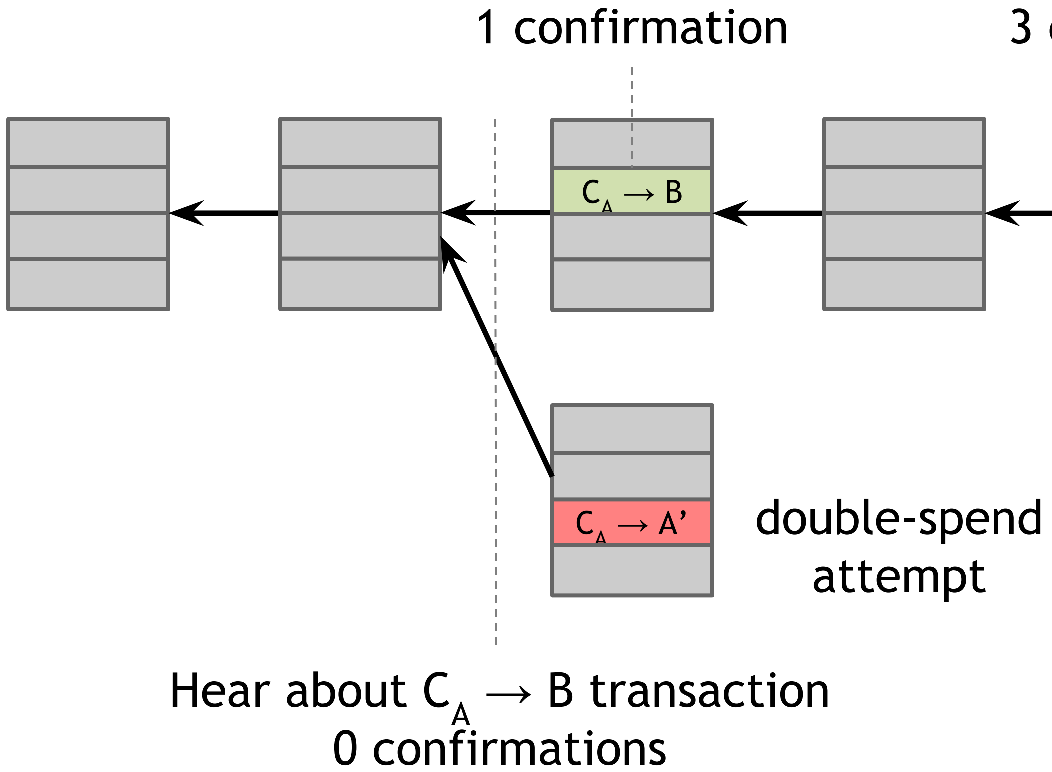
1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. In each round a <u>random</u> node gets to broadcast its block
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)
5. Nodes express their acceptance of the block by including its hash in the next block they create

What can a malicious node do?



Honest nodes will extend the <u>longest valid branch</u>

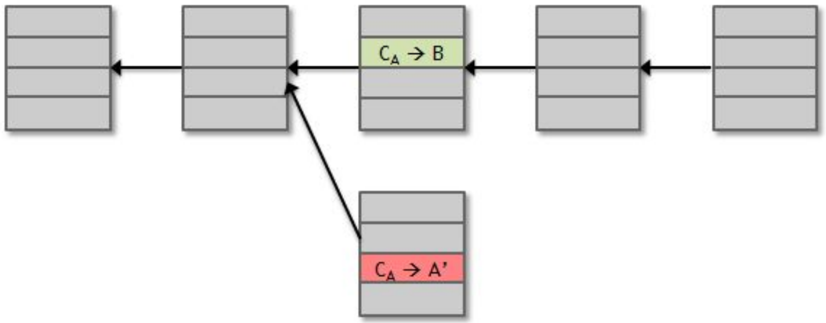
From Bob the merchant's point of view



Double-spend probability <u>decreases exponentially</u> with # of confirmations

Most common heuristic:
6 confirmations

Recap



Protection against invalid transactions is cryptographic, but enforced by consensus

Protection against double-spending is purely by consensus

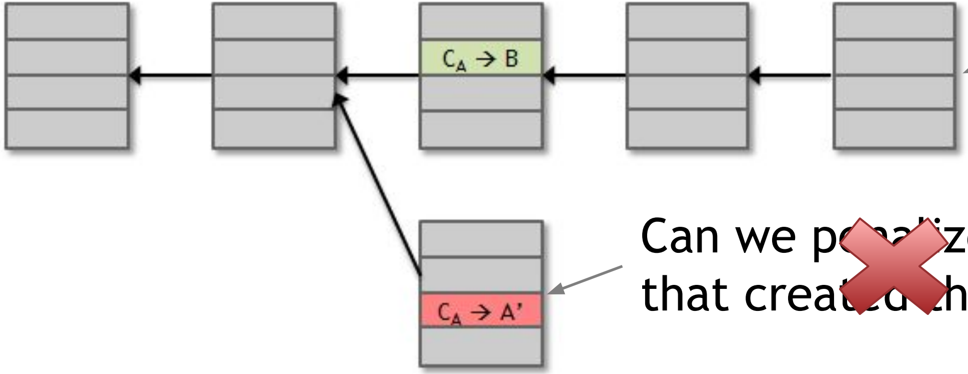
You’re never 100% sure a transaction is in consensus branch. Guarantee is probabilistic

Lecture 2.4:

Incentives and proof of work

Assumption of honesty is problematic

Can we give nodes <u>incentives</u> for behaving honestly?



Can we reward nodes
that created these blocks?

Can we penalize the node
that created this block?

Everything so far is just a distributed consensus protocol
But now we utilize the fact that the currency has value

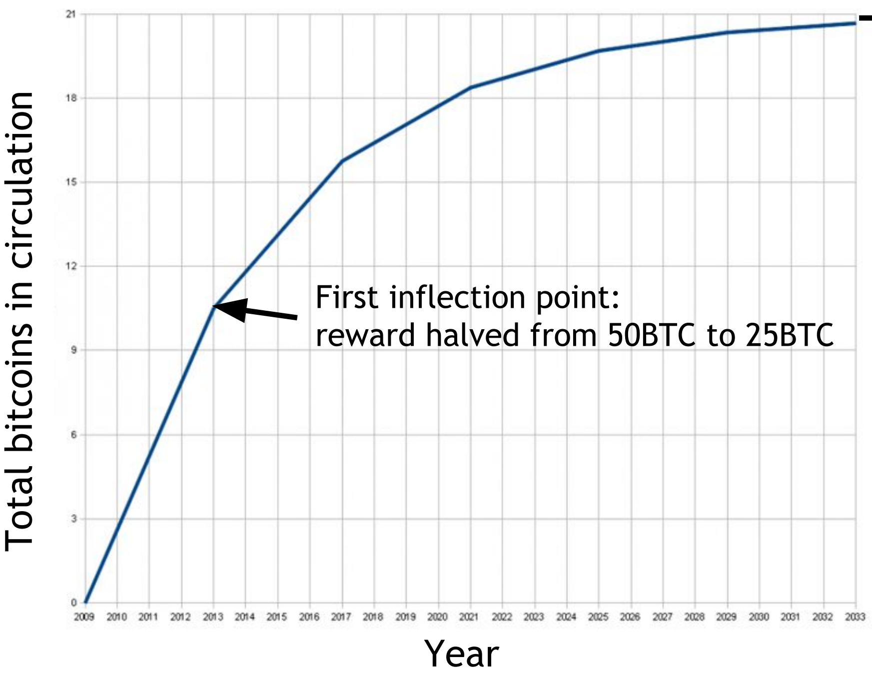
Incentive 1: block reward

Creator of block gets to
- include <u>special coin-creation transaction</u> in the block
- choose recipient address of this transaction

Value is fixed: currently 25 BTC, halves every 4 years

Block creator gets to “collect” the reward only if the block ends up on long-term consensus branch!

There's a finite supply of bitcoins



Total supply: 21 million

Block reward is how new bitcoins are created

Runs out in 2040. No new bitcoins unless rules change

Incentive 2: transaction fees

Creator of transaction can choose to make output value less than input value

Remainder is a transaction fee and goes to block creator

Purely voluntary, like a tip

Remaining problems

1. How to pick a random node?

2. How to avoid a free-for-all due to rewards?

3. How to prevent Sybil attacks?

Proof of work

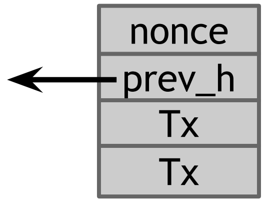
To approximate selecting a random node:
select nodes in proportion to a resource
that no one can monopolize (we hope)

- In proportion to computing power: proof-of-work
- In proportion to ownership: proof-of-stake

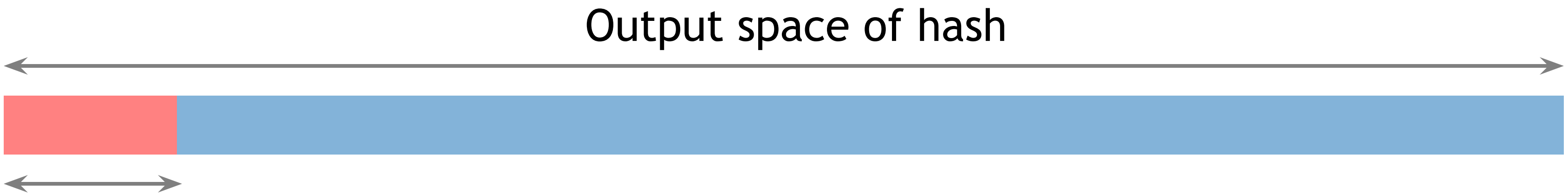
Equivalent views of proof of work

1. Select nodes in proportion to computing power
2. Let nodes compete for right to create block
3. Make it moderately hard to create new identities

Hash puzzles



To create block, find nonce s.t.
H(nonce ‖ prev_hash ‖ tx ‖ … ‖ tx) is very small



Target space

If hash function is secure:
only way to succeed is to try enough nonces until you get lucky

PoW property 1: difficult to compute

As of Aug 2014: about 10^{20} hashes/block

Only some nodes bother to compete –
miners

PoW property 2: parameterizable cost

Nodes automatically re-calculate the target every two weeks

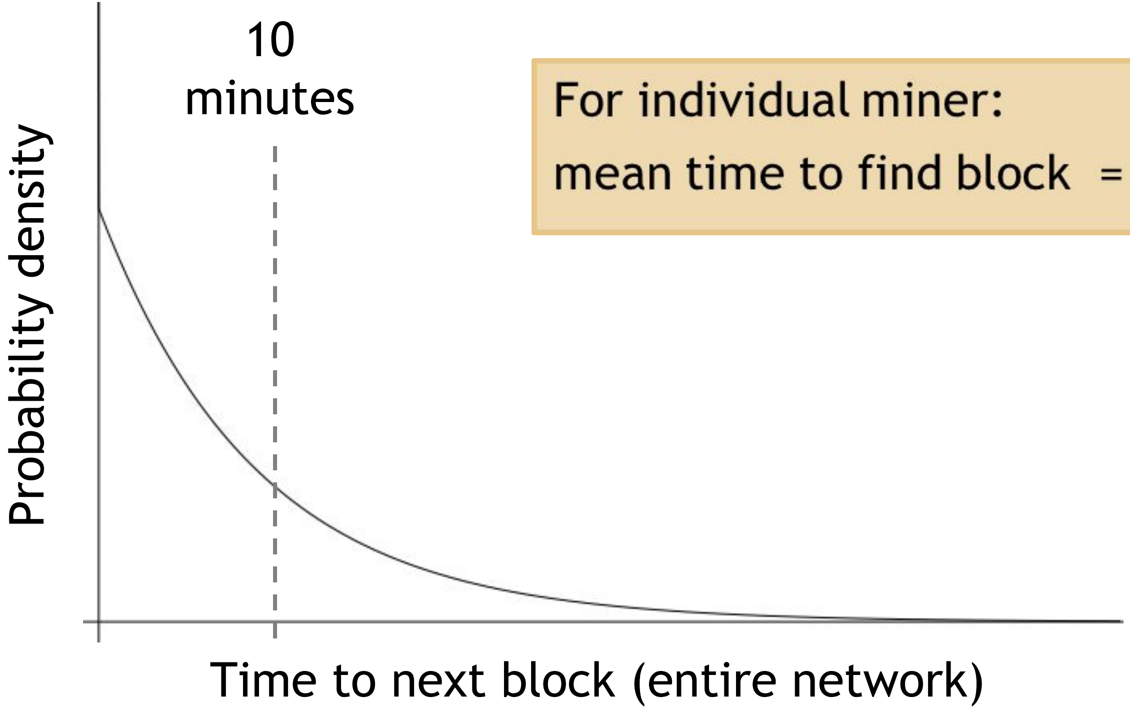
Goal: <u>average</u> time between blocks = 10 minutes

Prob (Alice wins next block) =
fraction of global hash power she controls

Key security assumption

Attacks infeasible if majority of miners <u>weighted by hash power</u> follow the protocol

Solving hash puzzles is probabilistic



For individual miner:

$$\text{mean time to find block} = \frac{\text{10 minutes}}{\text{fraction of hash power}}$$

PoW property 3: trivial to verify

Nonce must be published as part of block

Other miners simply verify that
$H(\text{nonce} \parallel \text{prev\_hash} \parallel \text{tx} \parallel \ldots \parallel \text{tx}) < \text{target}$

Mining economics

If mining reward (block reward + Tx fees)	>	hardware + electricity cost	→	Profit

Complications:

- fixed vs. variable costs
- reward depends on global hash rate

Lecture 2.5:

Putting it all together

Recap

Identities

Transactions

P2P network

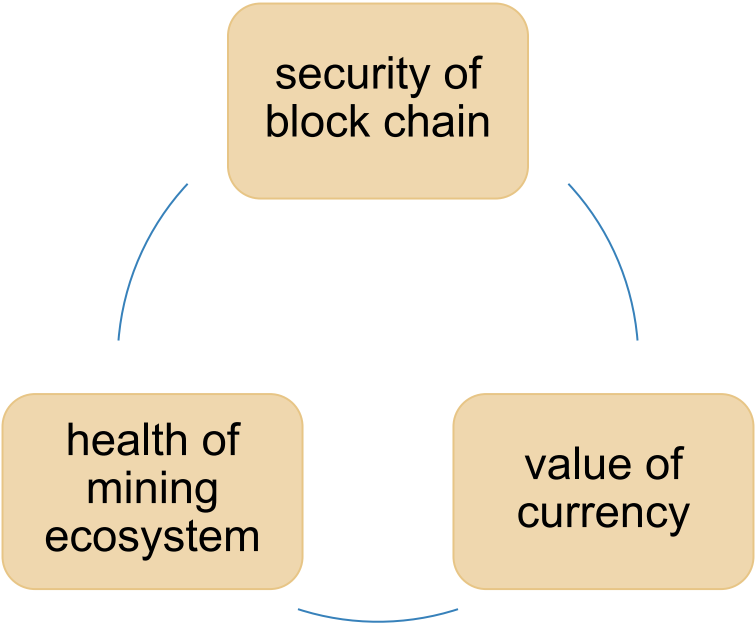
Block chain & consensus

Hash puzzles & mining

Bitcoin has three types of consensus

- Value
- State
- Rules

Bitcoin is bootstrapped



What can a “51% attacker” do?

Steal coins from existing address? ✗

Suppress some transactions?
- From the block chain ✓
- From the P2P network ✗

Change the block reward? ✗

Destroy confidence in Bitcoin? ✓✓

Remaining questions

How do we get from consensus to currency?

What else can we do with consensus?