Nakamoto Consensus

Marco Canini

Slides reproduced with permission from Bitcoin and Cryptocurrency Technologies
by Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder

Recap digital currency

GoofyCoin

Goofy can create new coins

signed by pk_{Goofy}
CreateCoin [uniqueCoinID]

A coin's owner can spend it.

signed by pk_{Goofy}

Pay to pk_{Alice} : H( )

signed by pk_{Goofy}

CreateCoin [uniqueCoinID]

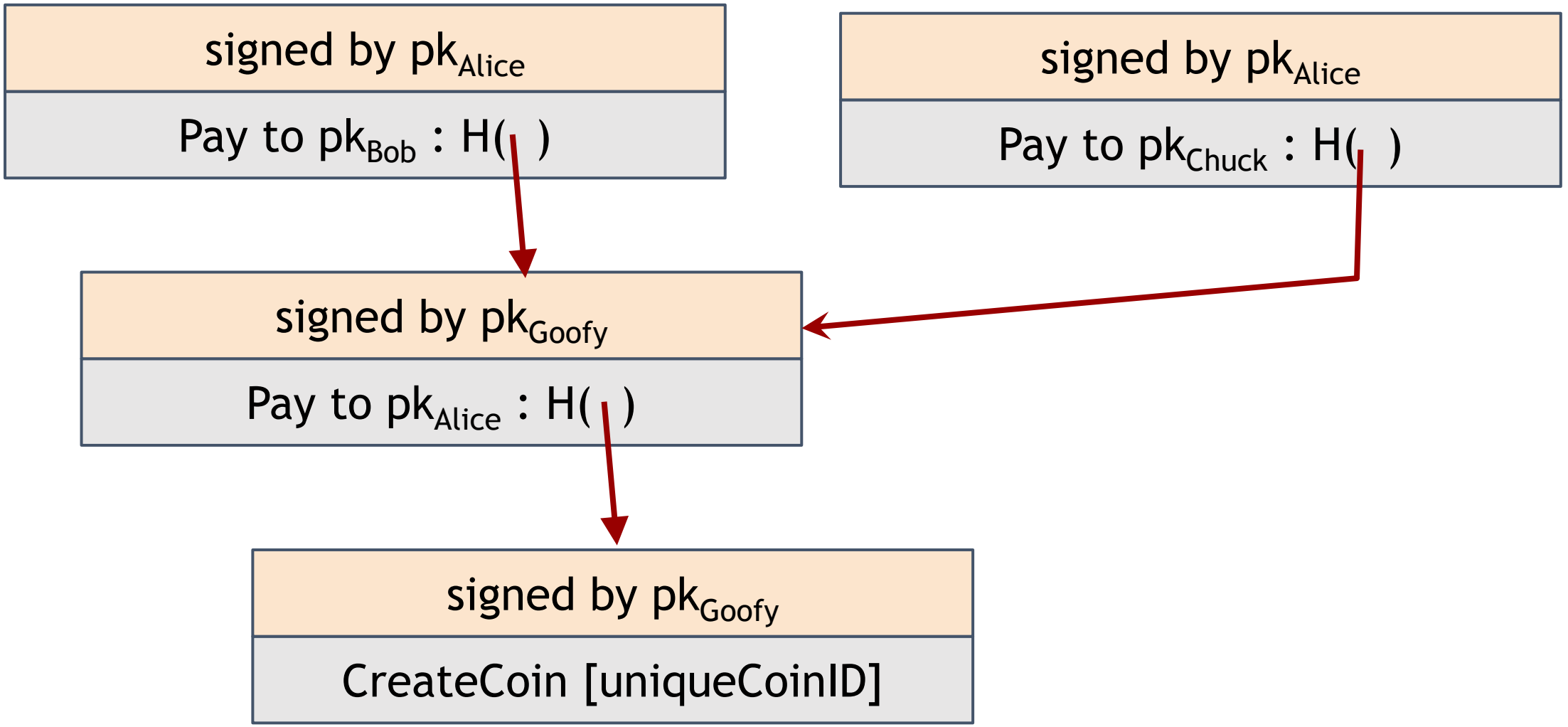
The recipient can pass on the coin again.

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

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

signed by pk_{Goofy}
CreateCoin [uniqueCoinID]

double-spending attack



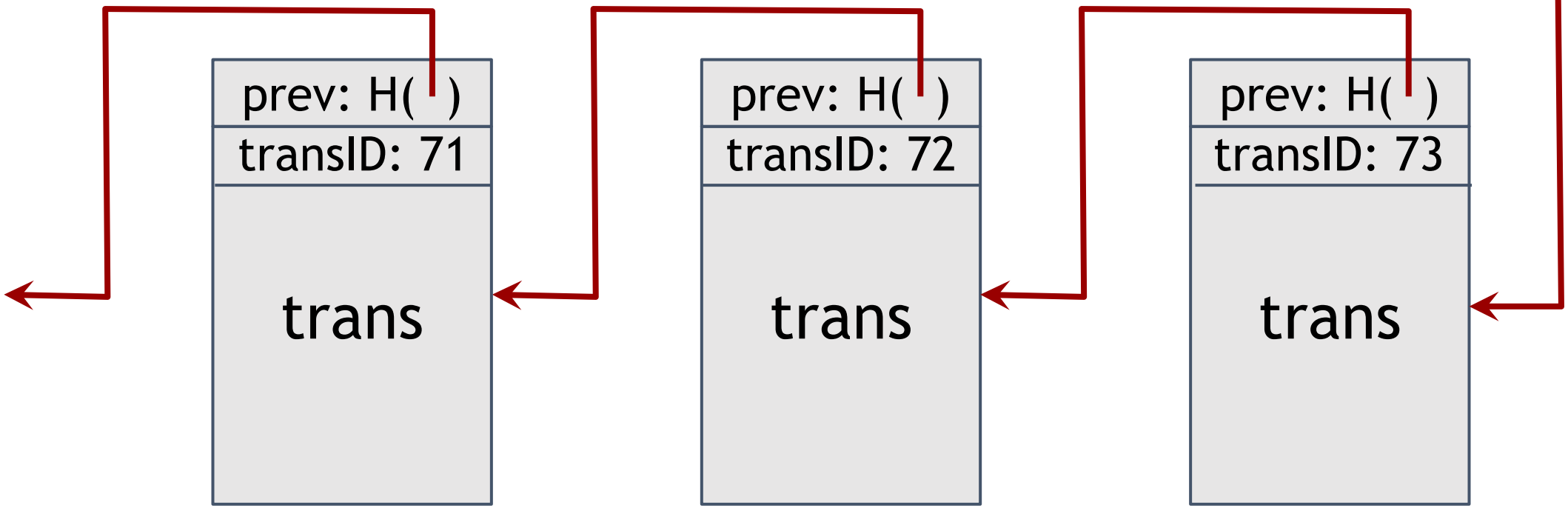
double-spending attack

the main design challenge in digital currency

ScroogeCoin

Scrooge publishes a history of all transactions
(a block chain, signed by Scrooge)

H( )



optimization: put multiple transactions in the same block

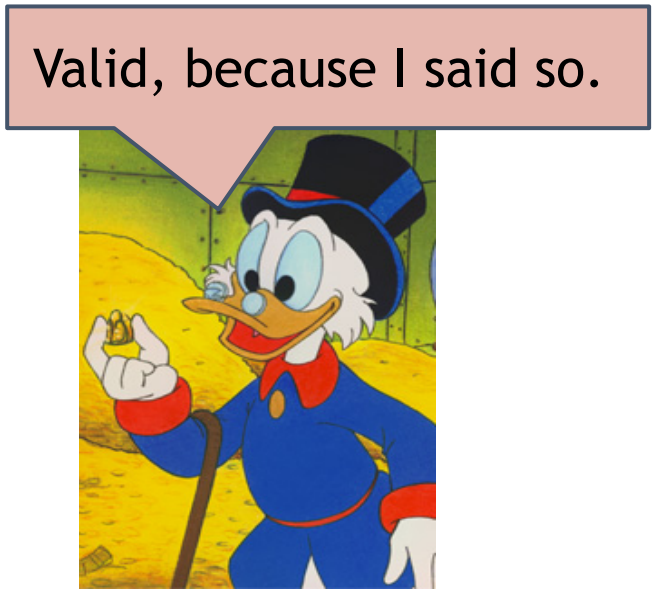
CreateCoins transaction creates new coins

transID: 73	type:CreateCoins	
coins created		
num	*value*	*recipient*
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...

coinID 73(0)

coinID 73(1)

coinID 73(2)



PayCoins transaction consumes (and destroys) some coins, and creates new coins of the same total value

transID: 73 type:PayCoins		
consumed coinIDs: 68(1), 42(0), 72(3)		
coins created		
num	*value*	*recipient*
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...
signatures		

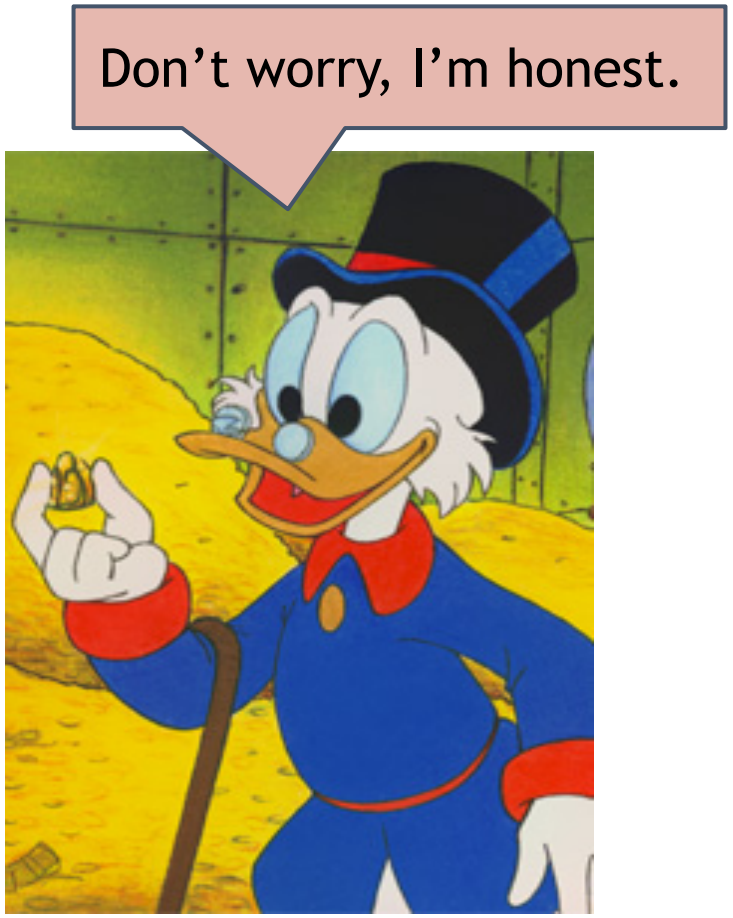
Valid if:
-- consumed coins valid,
-- not already consumed,
-- total value out = total value in, and
-- signed by owners of all consumed coins

Immutable coins

Coins can’t be transferred, subdivided, or combined.

But: you can get the same effect by using transactions

- to subdivide: create new trans
  - consume your coin
  - pay out two new coins to yourself



Crucial question:

Can we descroogify the currency, and operate without any central, trusted party?

Nakamoto consensus

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

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

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

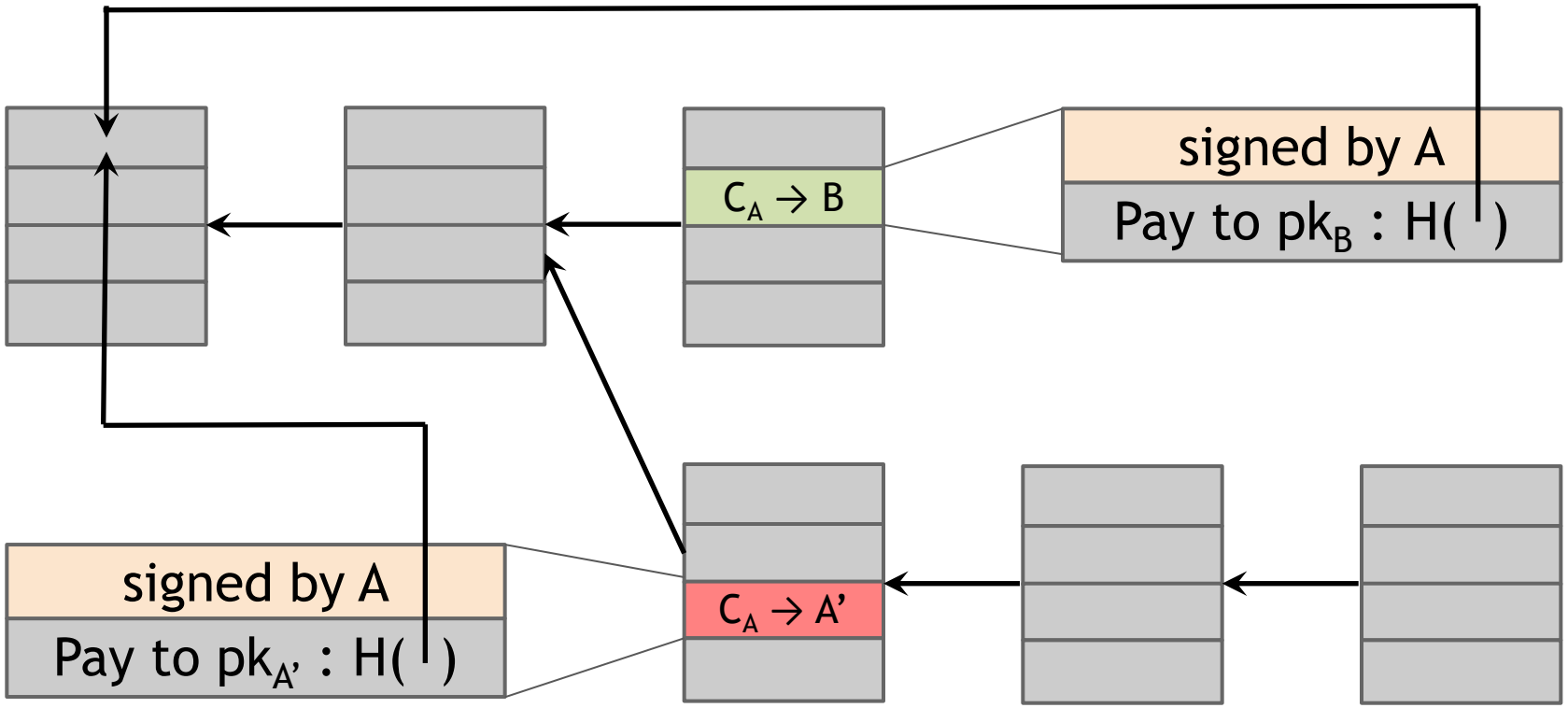
- by either extending it
- or ignoring it and extending chain from earlier block

Every block contains hash of the block it extends

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 random 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

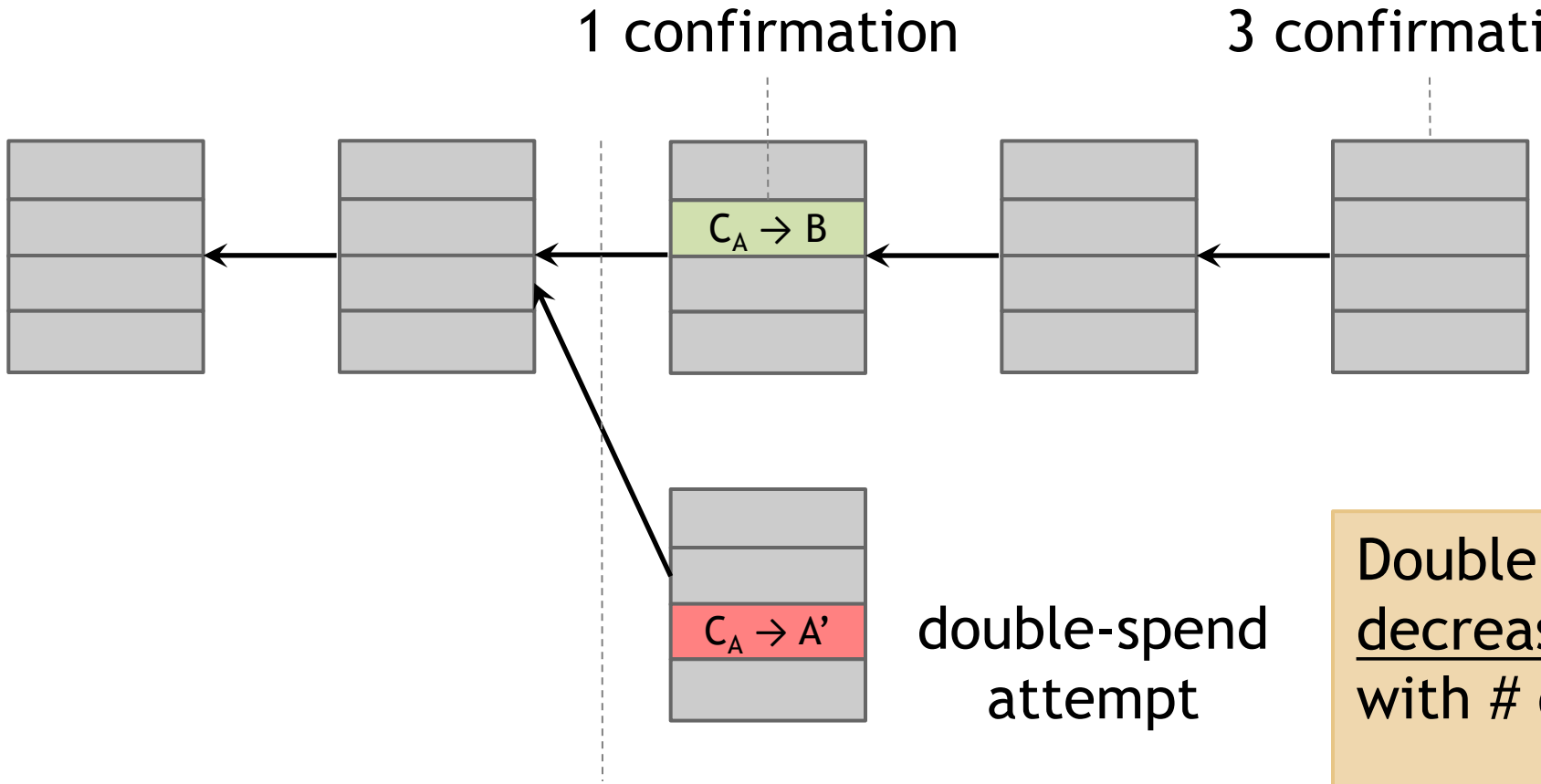
What can a malicious node do?



Double-spending attack

Honest nodes will extend the longest valid branch

From Bob the merchant's point of view

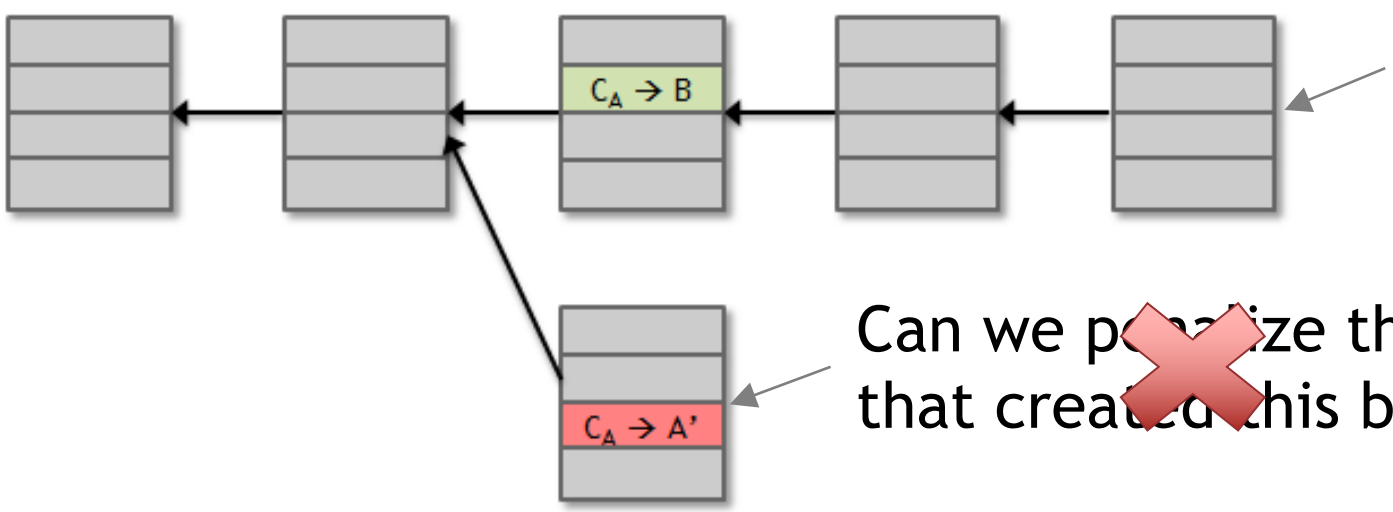


Double-spend probability <u>decreases exponentially</u> with # of confirmations

Most common heuristic:
6 confirmations

Assumption of honesty is problematic

Can we give nodes incentives 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

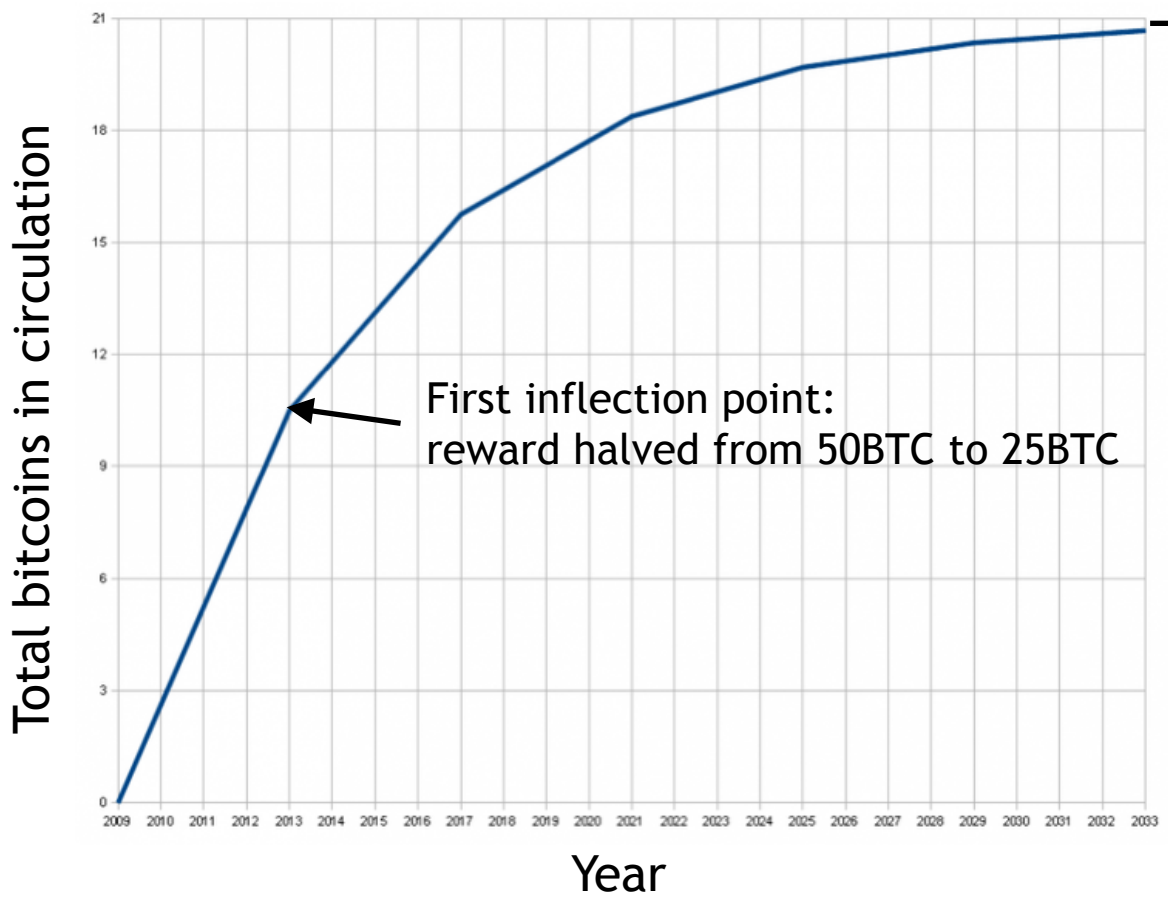
Creator of block gets to

- include special coin-creation transaction 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?

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

1. 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

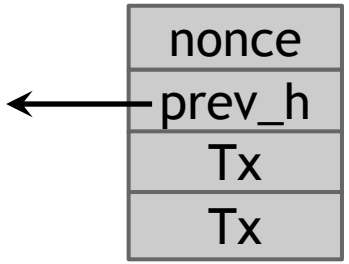
1. Select nodes in proportion to computing power

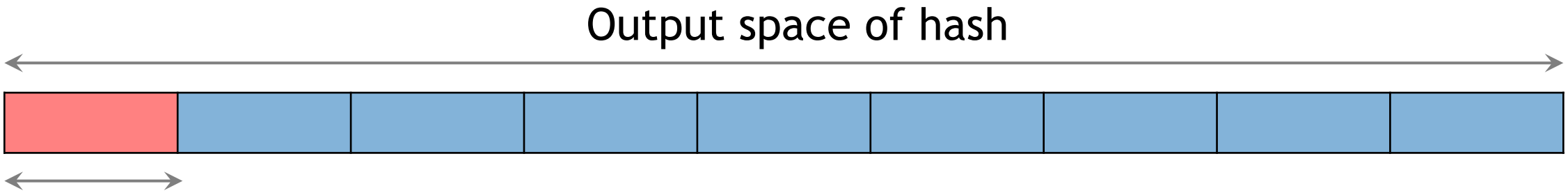
1. Let nodes compete for right to create block

1. 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





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

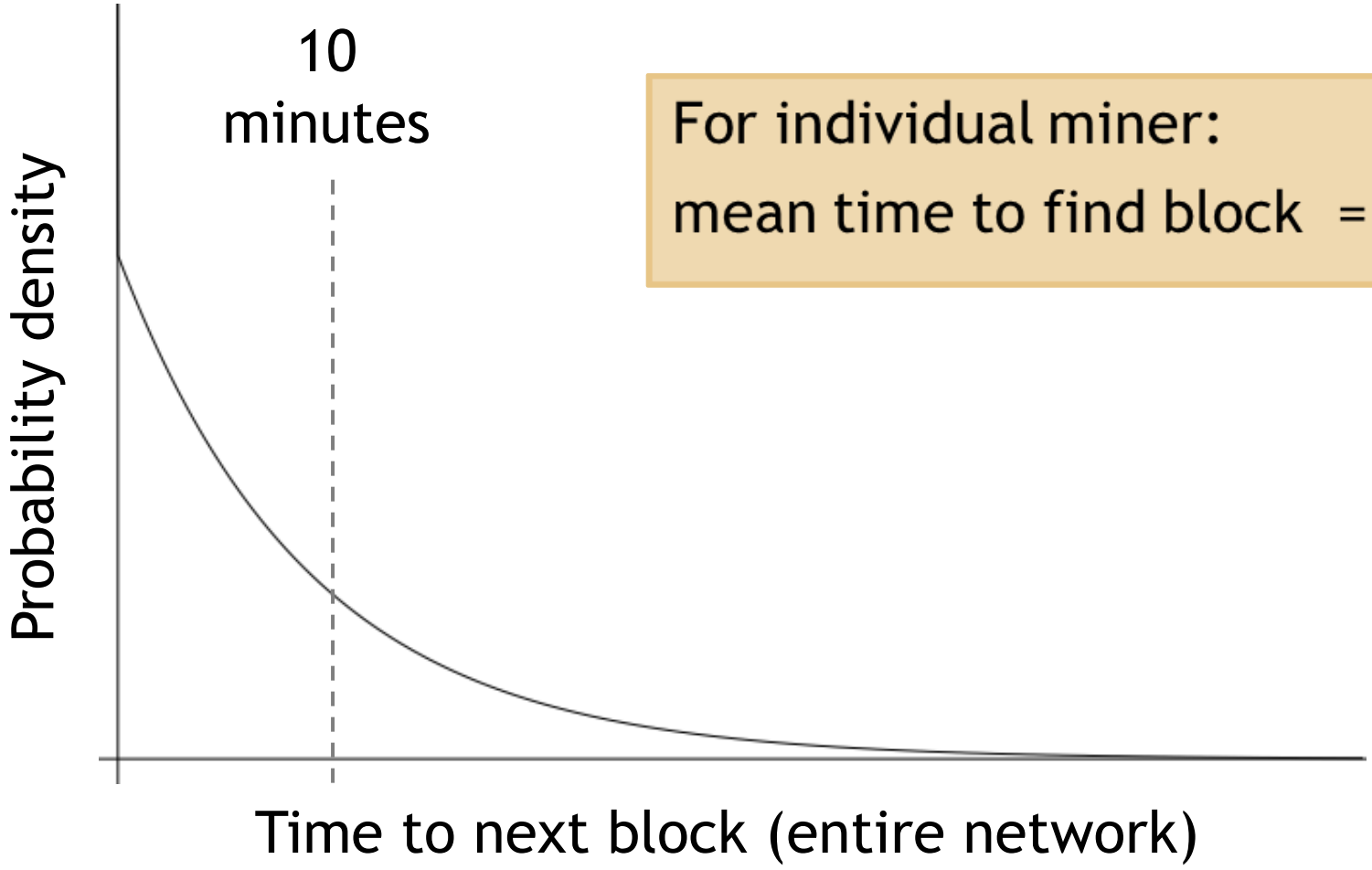
Goal: average time between blocks = 10 minutes

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

Key security assumption

Attacks infeasible if majority of miners weighted by hash power follow the protocol

Solving hash puzzles is probabilistic



For individual miner:

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

PoW property 3: trivial to verify

Nonce must be published as part of block

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

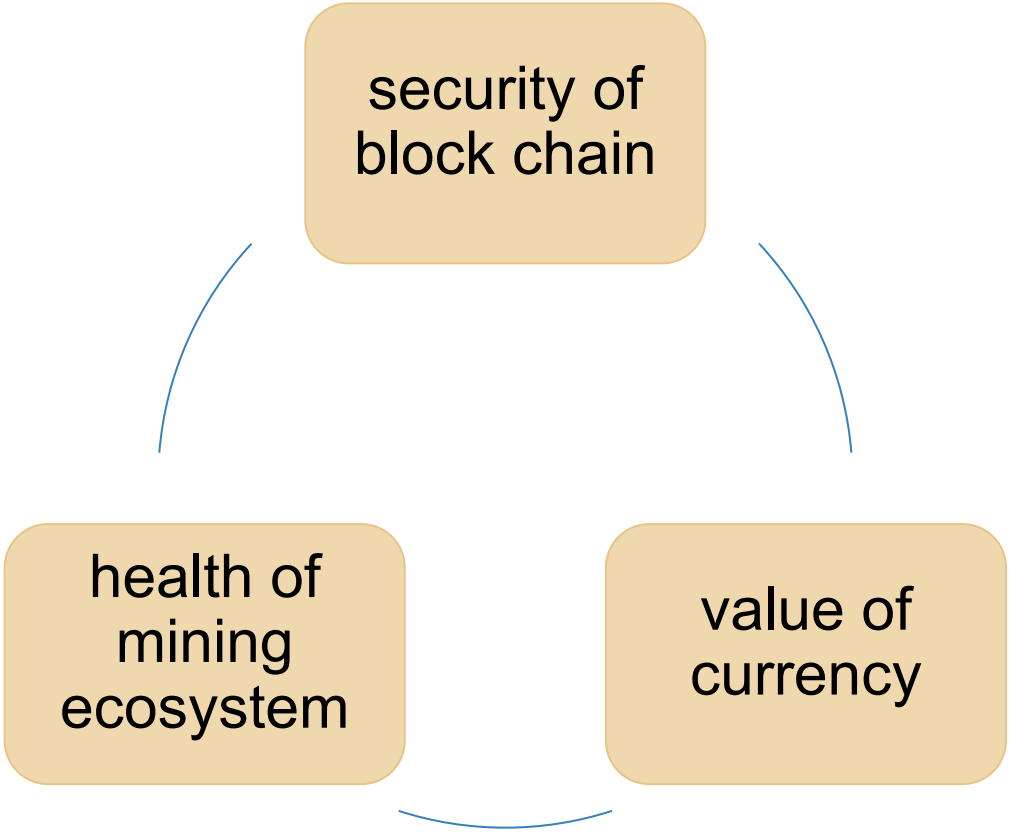
Mining economics

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

Complications:

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

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? ✓ ✓