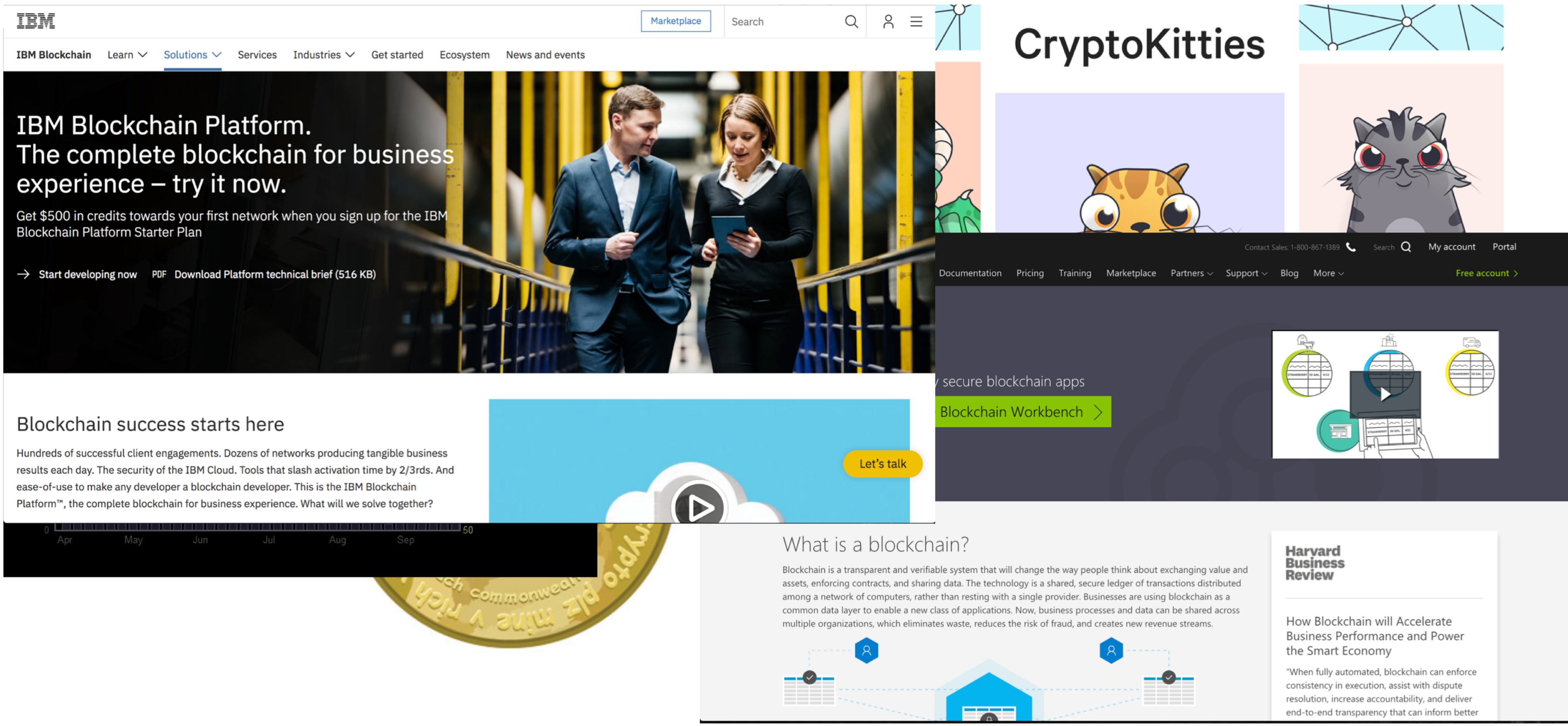
TRUTH JUSTICE ALGOS

Cryptocurrencies: Basic Concepts

Teachers: Ariel Procaccia and Alex Psomas (this time)

CRYPTOCURRENCIES



HOW BITCOIN WORKS: MAINTAINING A LEDGER

From:	To:	$$$
	Arvind	200
	Matt	200
….	….	….
Matt	Alex	50
Arvind	Jacob	20
Jacob	Georgios	100
….	….	….

Start simple.

Remove trust and scale up.

PROBLEM #1: AUTHORIZING TRANSACTIONS

- What if someone (Alex) tries to move money to their account without the owner's (Matt) authorization?
- Fix: Digital Signatures!

From:	To:	$$$
	Arvind	200
	Matt	200
….	….	….
Matt	Alex	50
Arvind	Jacob	20
Jacob	Georgios	100
….	….	….
Matt	Alex	1000
Matt	Alex	1000

PROBLEM #1: AUTHORIZING TRANSACTIONS

From:	To:	$$$	Signed
....	....	....	
....	....	....	
Alex	Georgios	100	Alex’s signature
Matt	Alex	1000	Matt’s signature
Matt	Alex	1000	
....	....	....	

BASIC CRYPTOGRAPHY 1: HASH FUNCTIONS

- Input:
  - Any string of any size
- Output:
  - Fixed size output (say 256-bits)
- Property #0: Efficiently computable
  - In fact linear time
- Property #1: Collision resistant
  - Basically impossible (computationally) to find a collision: inputs x and y that map to the same output $H(x) = H(y)$
  - Note: collisions exist. We ask that they are hard to find.

BASIC CRYPTOGRAPHY 1: HASH FUNCTIONS

- Property #2: Hiding
  - If a value x is chosen from a sufficiently big set, then given $H(x)$ it is hard to find x
  - Fancier: If a secret value r is chosen from a probability distribution with high min-entropy, then given $H(x||r)$ it is infeasible to find x
- Why a big set?

BASIC CRYPTOGRAPHY 1: HASH FUNCTIONS

- Example application: Commitment
- We want to run a sealed bid auction without a centralized authority
- How do you commit to a value?

1. Publish $H(nonce||value)$
   - $nonce =$ large random number
2. After you see everyone's $H(.)$, publish $value$ and $nonce$

- Everyone can verify whether someone lied!

BASIC CRYPTOGRAPHY 2: SIGNATURES

- Problem: I want to cryptographically sign a document
  - Only I should be able to sign it (**unforgeability**), but everyone should be able to check that my signature is valid
- Solution: Public key cryptography
- I have has a private key p_1
  - Only I know p_1
- I have a public key p_2
  - Everyone knows p_2
- Functionality:
  - $Sign(doc, p_1) = \text{signature}$
  - $Verify(signature, p_2, doc) \in \{Valid, Invalid\}$

RSA FOR ENCODING/DECODING

- Pick two large primes p and q
- $N = p \cdot q$
- Choose e relatively prime to $(p-1)(q-1)$
- Compute $d = e^{-1} mod (p-1)(q-1)$
- Public key: N and e
- Private key: d
- $Encode(x) = x^e mod\ N$
- $Decode(x) = x^d mod\ N$

RSA FOR SIGNATURES

- Pick two large primes p and q
- $N = p \cdot q$
- Choose e relatively prime to $(p-1)(q-1)$
- Compute $d = e^{-1} mod (p-1)(q-1)$
- Public key: N and e
- Private key: d
- $Sign(x) = x^{d} mod\ N$
- $Verify(x) = x^{e} mod\ N$

PROBLEM #1: AUTHORIZING TRANSACTIONS

From:	To:	$$$	Signed
….	….	….	
….	….	….	
Alex	Georgios	100	Alex's signature
Matt	Alex	1000	Matt's signature
Matt	Alex	1000	
….	….	….	

Sign(| Matt | Alex | 1000 | , Matt's private key) = Matt's signature

Verify(signature, Matt's public key, | Matt | Alex | 1000 |) ∈ { Valid, Not Valid }

PROBLEM #2: SPENDING MONEY YOU DON'T HAVE

What if someone (Georgios) tries to spend money they don't have?

From:	To:	$$$	Signed
….	….	….	
….	….	….	
Georgios	Matt	1000	Georgios' sign.
Georgios	Jacob	1000	Georgios' sign.
Georgios	Arvind	1000	Georgios' sign.
….	….	….	

PROBLEM #2: SPENDING MONEY YOU DON'T HAVE

- Fix: Scan past transactions and check flow of money.

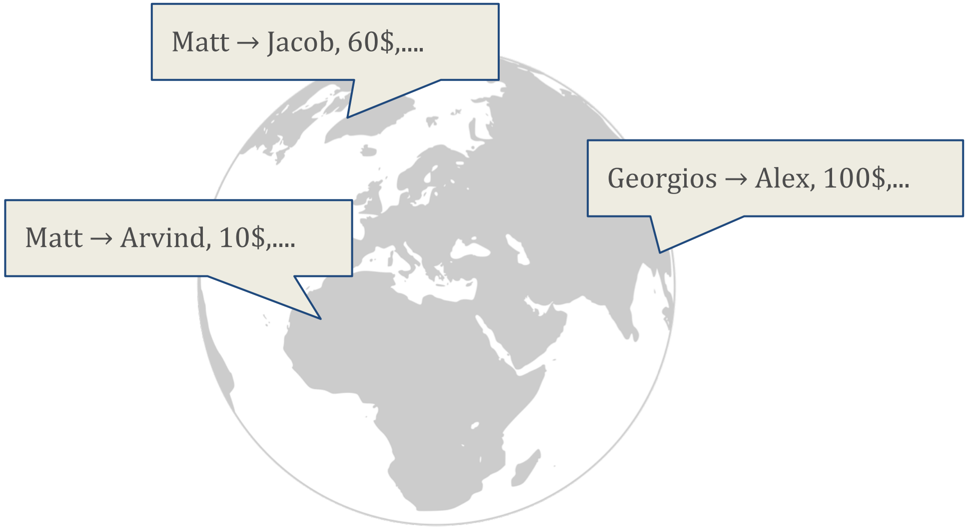
Make sure this money wasn't spent in this interval (#123 – #1100)

	From:	To:	$$$	Input	Signed
#123	Alex	Georgios	100	#51	Alex's sign.
....	....	....	....	....	
#256	Matt	Georgios	900	#100	Matt's sign.
....	....	....	....	....	....
#1100	Georgios	Arvind	1000	#123, #256	Georgios' sign.
....	....	....	....		

PROBLEM #3: DECENTRALIZE

<u>With a trusted center</u>

- Center maintains a single ledger
- Center adds transactions as they come.
- Center checks validity.
- Center makes sure no one double spends.
- Center adds new people to the system.



PROBLEM #3: DECENTRALIZE

<u>Without a trusted center</u>

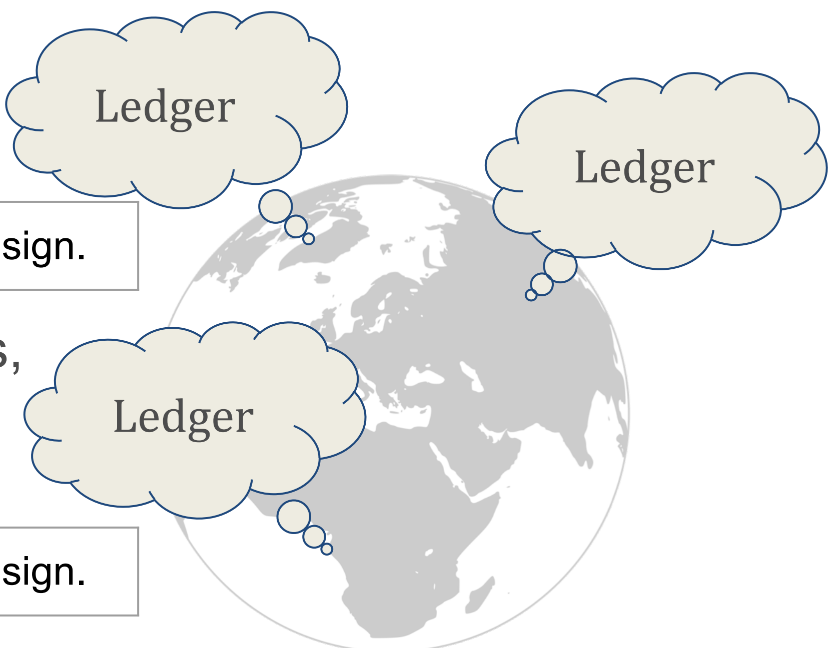
- Who maintains the ledger?
- Who has authority over which transactions are valid?
- How do we prevent double spending?

Scenario: Alex wants to buy a car from Matt.

Alex	Matt	10000$	Input = #127	Alex's sign.

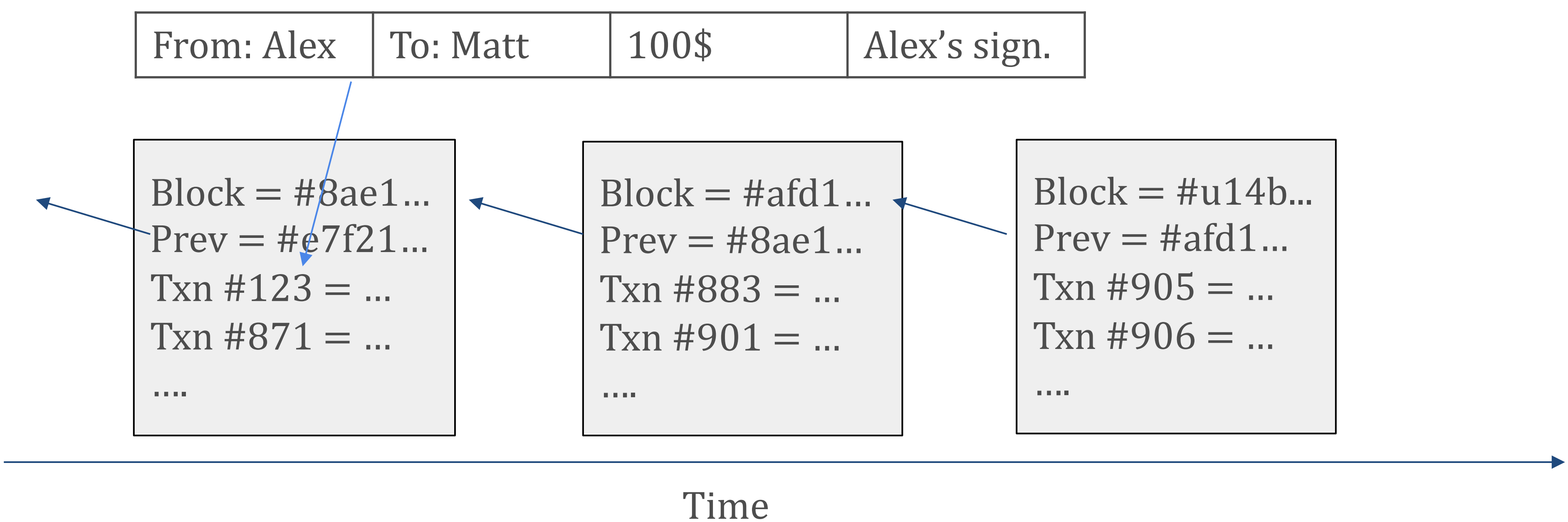
As soon as Matt gives Alex the keys, broadcast:

Alex	Jacob	10000$	Input = #127	Alex's sign.



BLOCKCHAIN

- Cryptocurrency maintains a directed graph of **blocks.**
- A block contains confirmed/valid transactions
- If a transaction is not in a block then by definition it is not valid/confirmed.



HOW ARE BLOCKS MADE?

- Blocks are made by **miners**
- Miners validate new transactions and add them to the blockchain
- Problem: which miner gets to make the next block?
  - Pick a participant uniformly at random
  - The environment is permissionless!
  - Users can create as many accounts (“Sybils”) as the want!
  - Effectively a race to ~~the bottom~~ see who can create more Sybils.

PROOF OF WORK

Miners compete to solve a "**crypto puzzle**"

Goal: The cryptographic hash of the entire text of a block plus an additional number (the nonce) must be in a certain range

SHA256 (
Block =...
Txn #905 = ...
Txn #906 = ...
....
nonce
) = 0x00000000000000….00b39d9ca51f07fef3429ae15.

A bunch of zeros

Idea: Solving the puzzle first is proportional to your **computational power**!

HOW ARE BLOCKS MADE?

- Blocks are made by **miners**
- Miners validate new transactions and add them to the blockchain
- ~~Problem: which miner gets to make the next block?~~
- The first miner to solve the crypto puzzle broadcasts their block. It’s easy to check that it is a valid block
- Problem: why would miners for all this?

PROOF OF WORK

For doing this intensive work, block creators are rewarded in two ways:

- Block reward: add a special transaction giving the miner a certain number of (new) bitcoins. Currently 12.5 Bitcoin per block.
- Transaction fees: “tips” from the participants of the transaction to the miner, if the transaction is included in the new block.

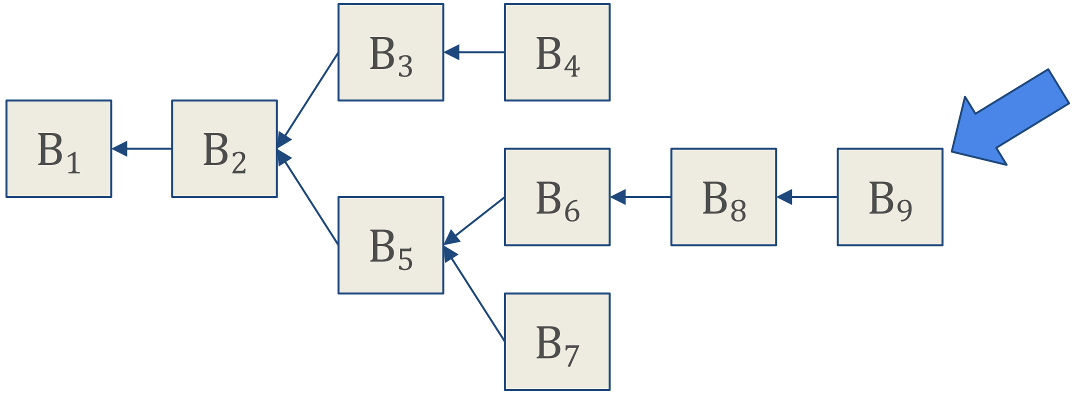
These rewards are “real” only if the block is in the “true” history, i.e. this block is an ancestor to future blocks.

DIFFICULTY ADJUSTMENT

- The number of leading zeros gets adjusted every 2016 blocks so that a block gets created every ~10 minutes
- The block reward is scheduled to be halved every 4 years
  - Eventually all rewards will come from transaction fees
  - Very different incentives

BRANCHES

- The network so far:

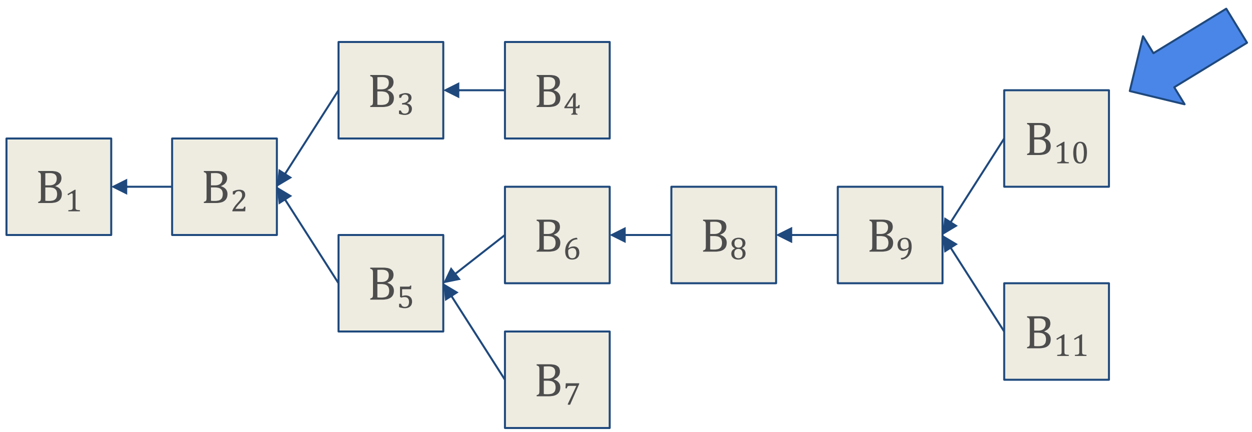


- Which block should a miner try to extend?

Longest chain

BRANCHES

- The network so far:

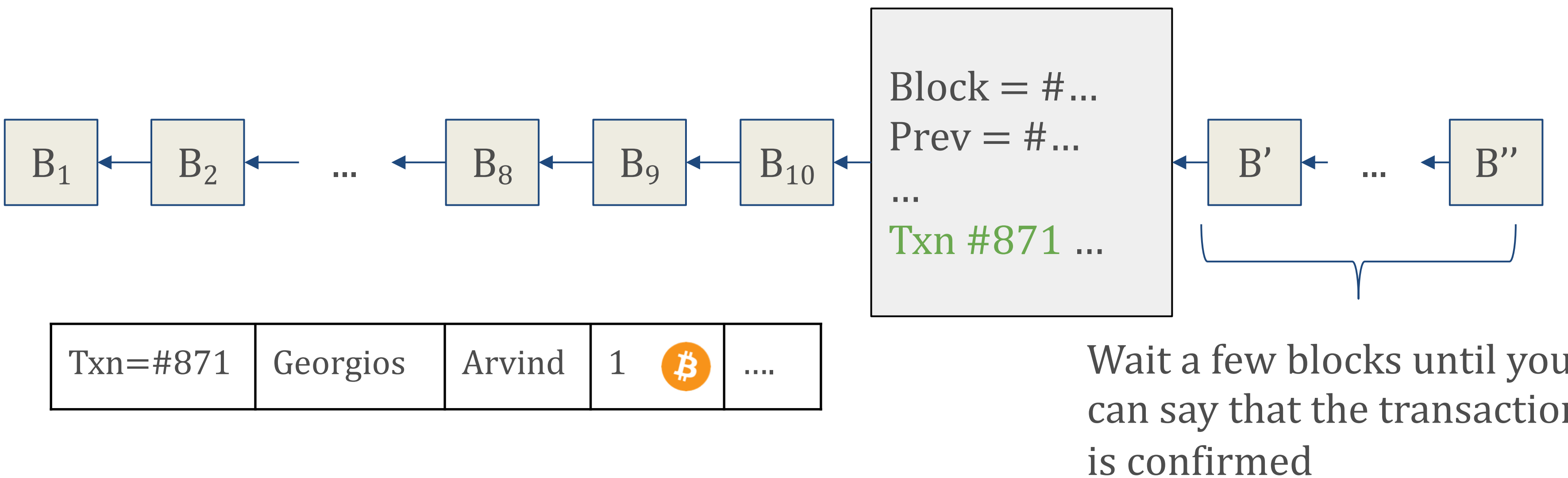


- More than one block is solved at the same time (plus natural network delays)
- Which block should a miner try to extend?

The first one you hear about

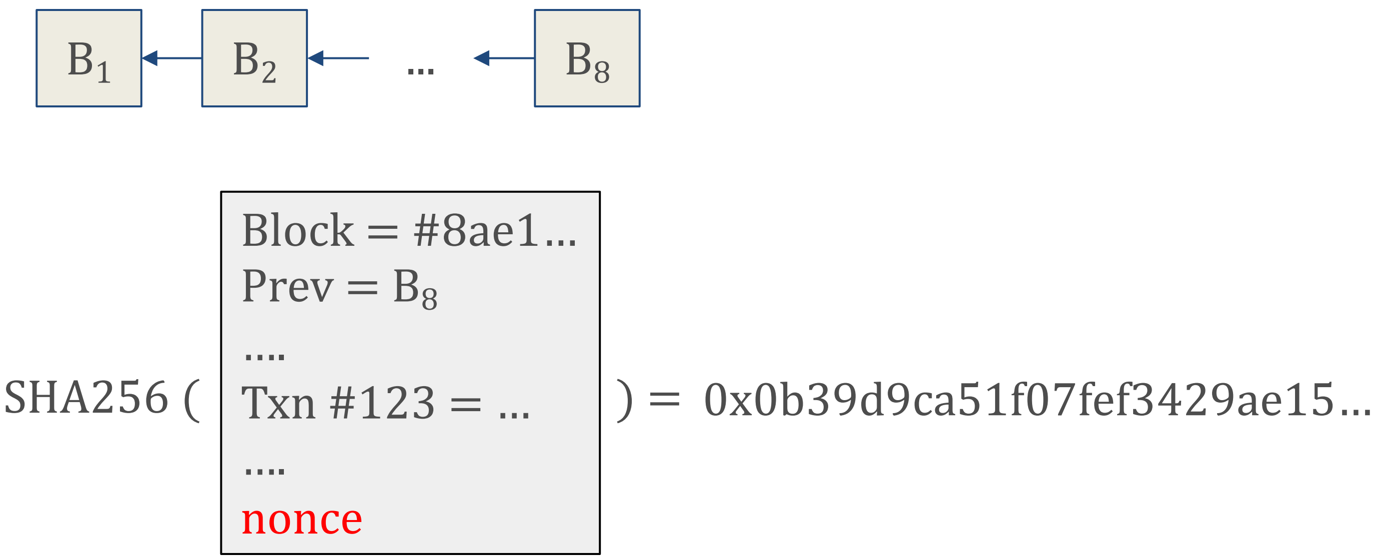
PROOF OF WORK: RECAP

View of someone who wants to make a transaction



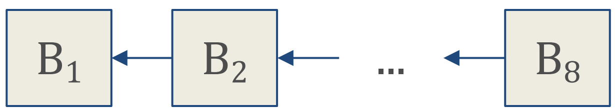
PROOF OF WORK: RECAP

View of a miner



PROOF OF WORK: RECAP

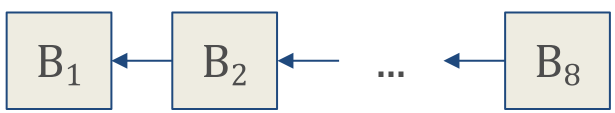
View of a miner



SHA256 (
Block = #8ae1…
Prev = B_8
….
Txn #123 = …
….
nonce'
) = 0x000000ef34244s1jd99a533g…

PROOF OF WORK: RECAP

View of a miner



SHA256 (Block = #8ae1…) = 0x1104000gf4jd8011889mdk3c…

Prev = B_8

….

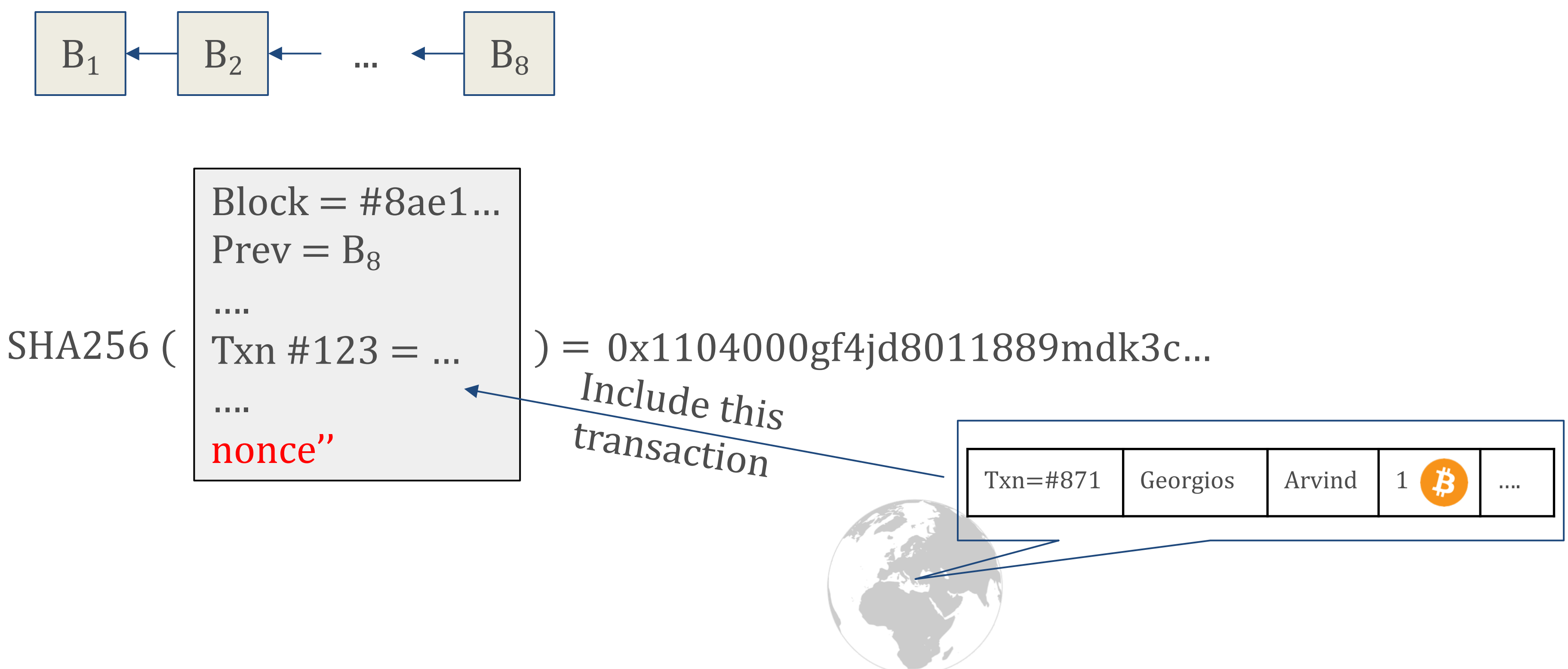
Txn #123 = …

….

nonce’’

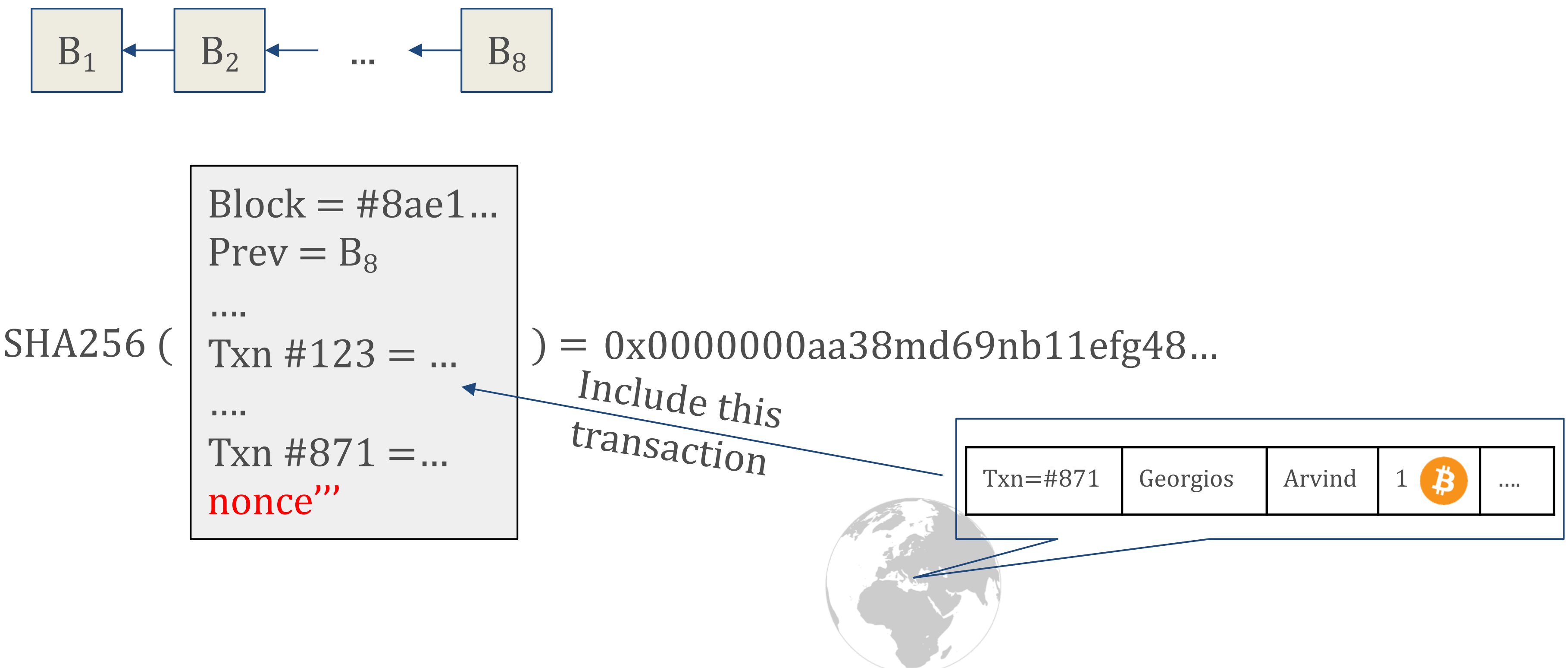
PROOF OF WORK: RECAP

View of a miner



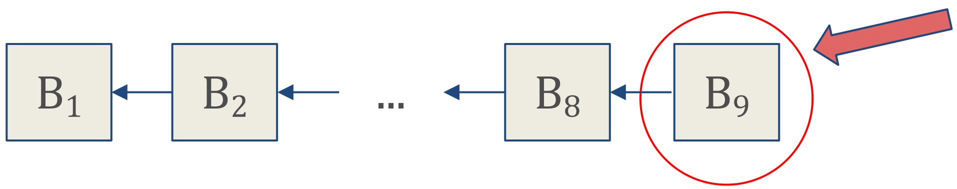
PROOF OF WORK: RECAP

View of a miner



PROOF OF WORK: RECAP

View of a miner



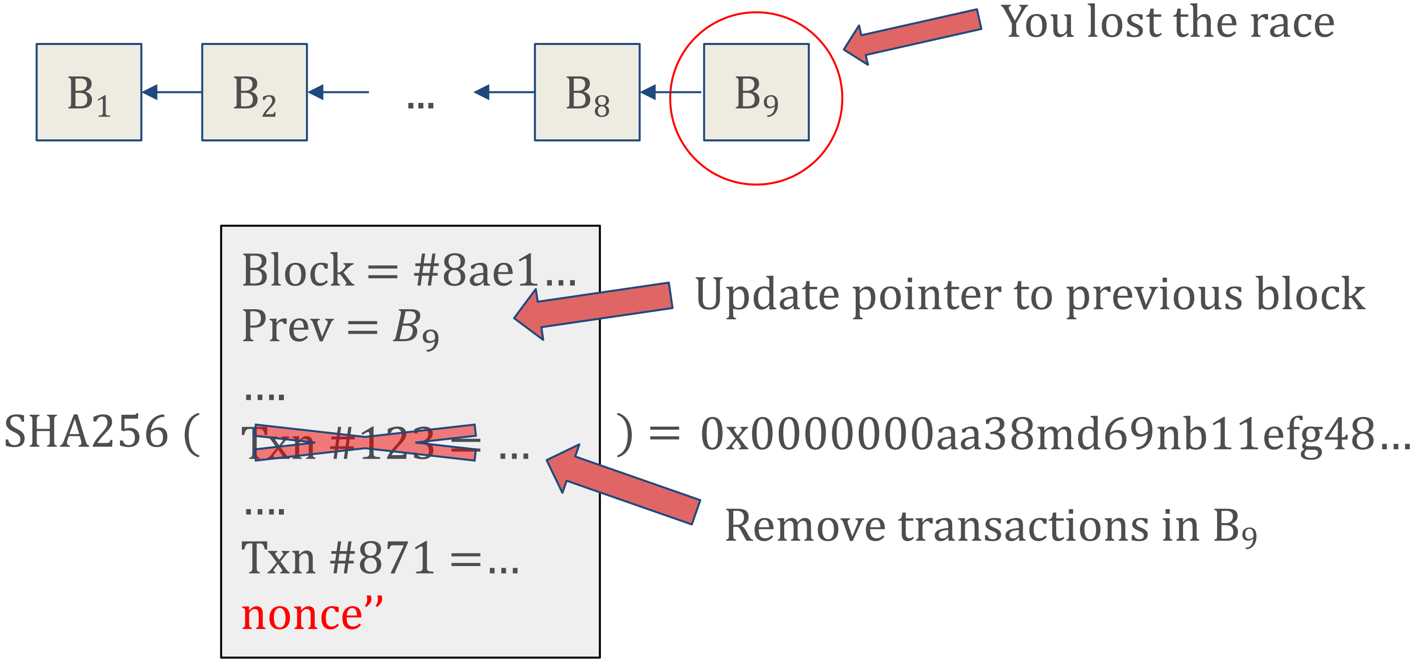
SHA256 (

Block = #8ae1…
Prev = B_8
….
Txn #123 = …
….
Txn #871 =…
nonce’’

) = 0x0000000aa38md69nb11efg48…

PROOF OF WORK: RECAP

View of a miner



PROOF OF WORK: RECAP

View of a miner

B_1 ← B_2 ← ... ← B_8 ← B_9

You lost the race

SHA256 (Block = #8ae1... Prev = B_9 .... ~~Txn #123 =~~ ... . Txn #871 =... nonce) = 0x0000000aa38md69nb11efg48...

Update pointer to previous block

Remove transactions in B_9

Keep trying!

POLL

1. Choose which block to try to extend
2. Forge transactions
3. Choose which transactions to include
4. Change the contents of published blocks
5. Win disproportionately often

Poll

What can a miner do?

1. 1,2 and 3
2. 1 and 3
3. 2, 4 and 5
4. 1 and 2

MINING POOLS

- Solo miner is very unlikely to find a new block, even with dedicated hardware
  - There are ~1.5 millions blocks a year
  - 12.5 bitcoins ~ 65000$
  - If you have 0.0000001% of the global mining power your expected annual income is 65000$
  - Too much variance!
- Miners create **pools**
  - If a miner in a pool mines a new block, the block reward gets split between all miners in the pool, proportional to their mining power

MINING POOLS

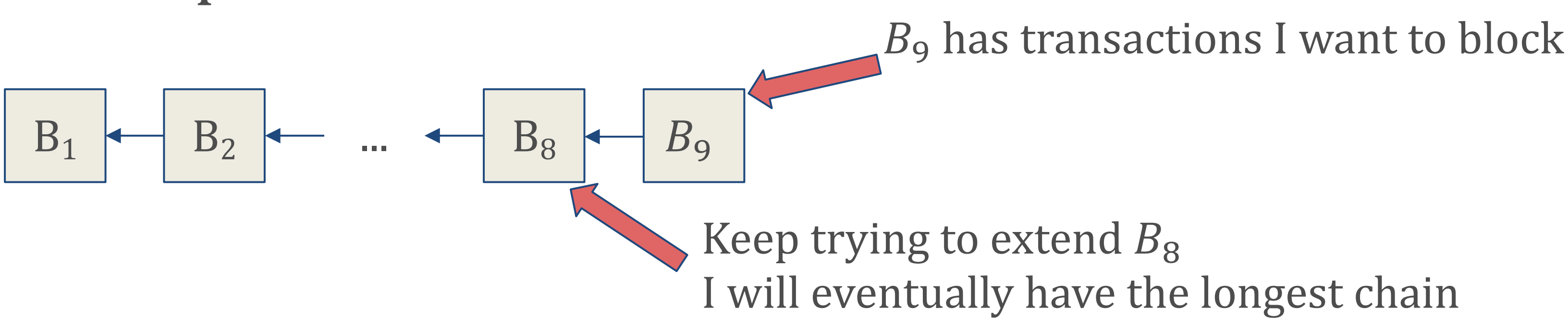
- How can you distribute rewards fairly?
- A valid block must hash to a number with (say) 70 leading zeros
- Typically, a (nonce that gives a) block with 65 leading zeros is useless
- Show it to the rest of the mining pool
- "Proof of effort"
  - Miners that are actively trying to find valid blocks and have a lot of computational power will produce these "partial" solutions often
- Distribute rewards proportional to partial solutions

IS BITCOIN SECURE?

- Can an adversary take control of Bitcoin?
- Say adversary has 51% of the computational power
- Can they steal money?
  - No!
  - If you add unsigned transactions sending money to yourself, even if you have the longest chain, the rest of the network will ignore your blocks

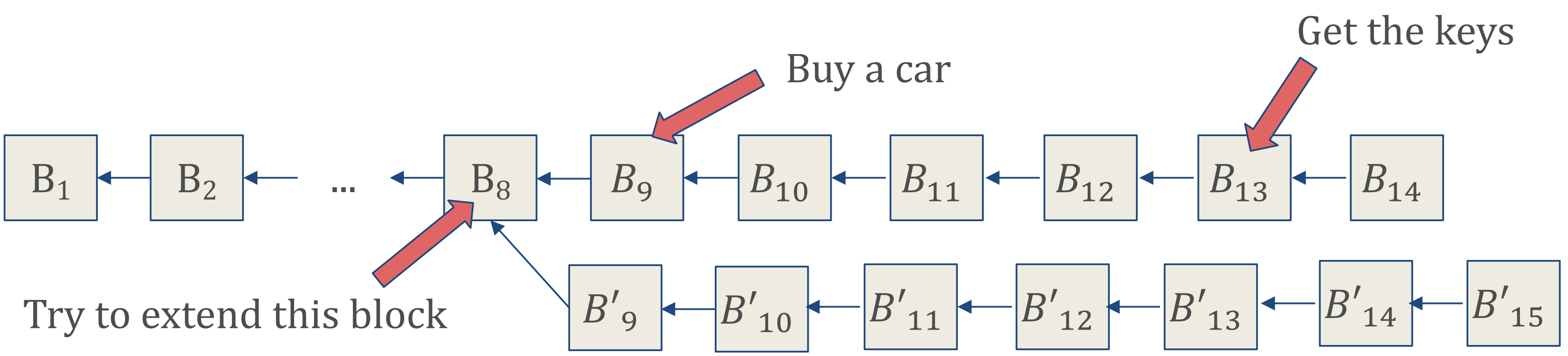
IS BITCOIN SECURE?

- Can they effectively remove someone from the network?
  - Yes!
  - If I don't like some participant, I can simply never include any of their transactions (anonymity makes this hard of course)
  - If someone mines before me and includes these transactions, I can ignore their block and mine on top of the old one



IS BITCOIN SECURE?

- Can they “double spend”?
  - Spend the same bit coin twice



- Since you have more computational power than the rest of the network, you’ll eventually catch up

IS BITCOIN SECURE?

- Bitcoin community: “Why would anyone do this?”
  - “If someone had 51% of all the mining power, they can make a bunch of money by mining normally”
  - “If they do these tricks, they’ll make Bitcoin not trusted, and therefore hurt their own investment”
- Arguing about incentives is precisely what we do in mechanism design!
  - Vs research in security, where you worry about trolls