BitCoin
“Consensus” without Paxos

Jinyang Li

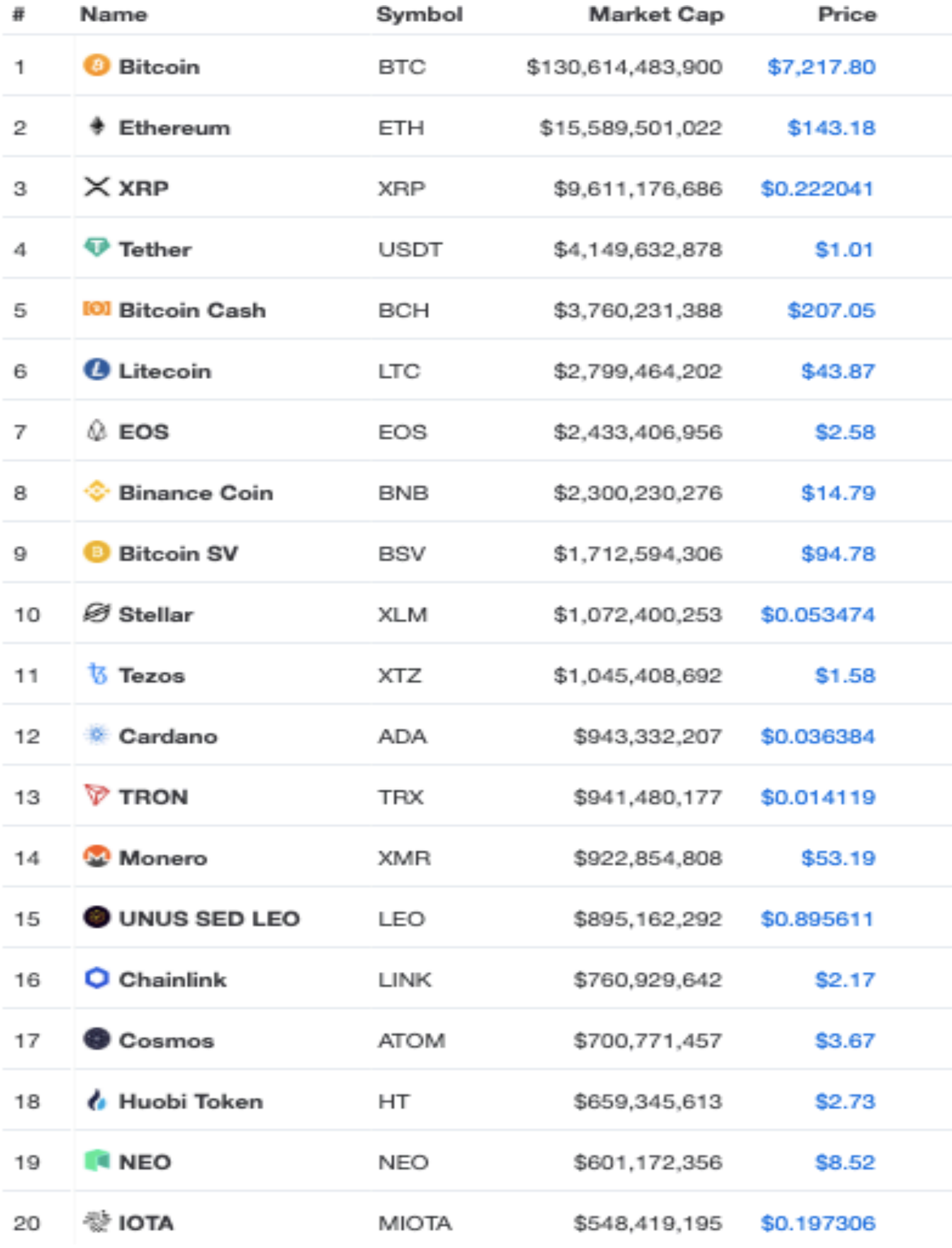
What we've learnt so far

- So far we discussed distributed systems within data centers
  - **closed** system
    - Managed by a single administrative entity (e.g. Google)
    - Only chosen machines participate
    - Participating machines are trusted (cooperative)
- Ideal consisency (linearizability)
  - Paxos for consensus (MultiPaxos for linearizable replication)

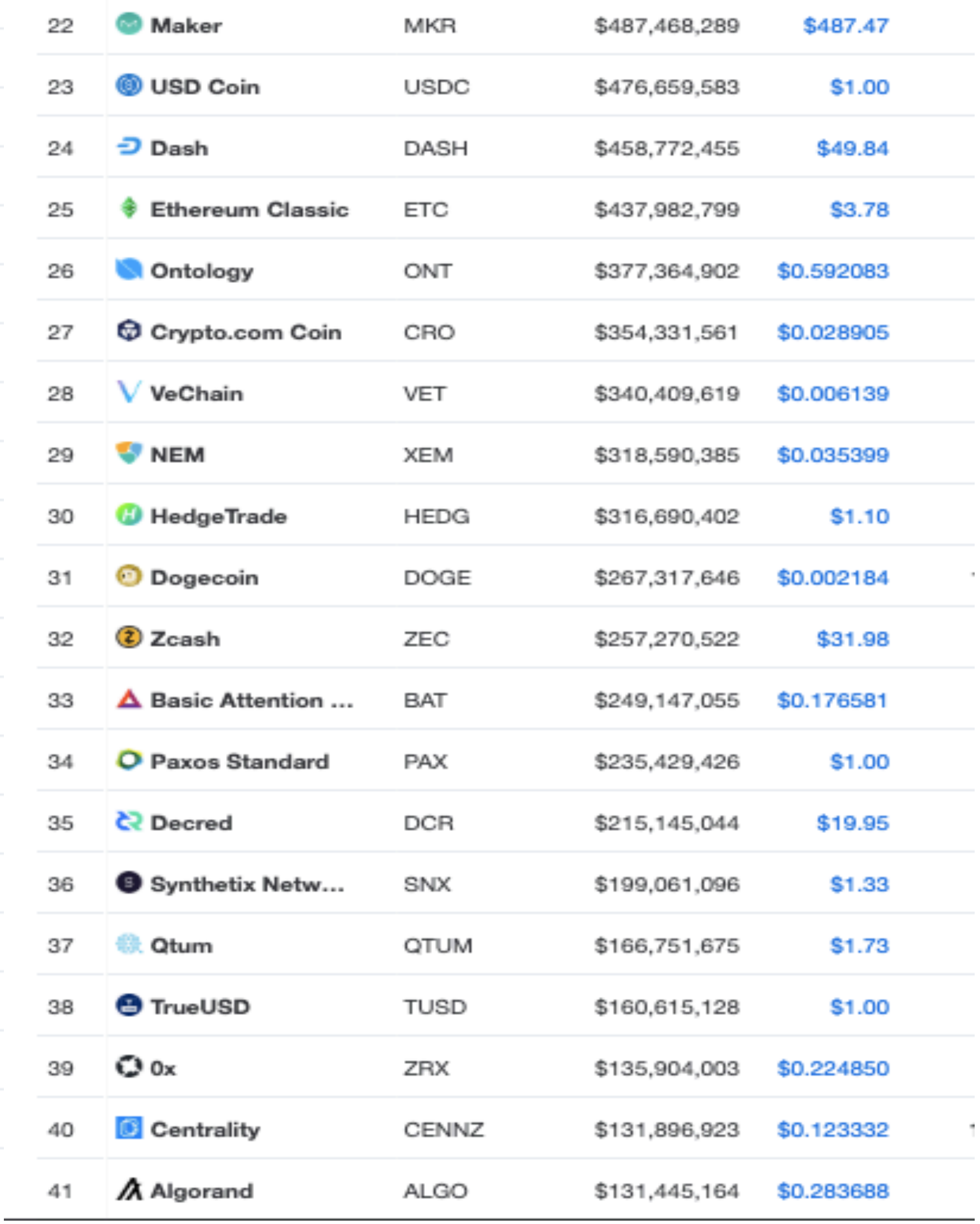
Today: BitCoin

- Very different from all other systems we’ve discussed in this class
- BitCoin is peer-to-peer (aka open system; aka decentralized)
  - any machine can participate in the protocol
  - no single administrative entity
- BitCoin is the first practical cryptocurrency

Many cryptocurrencies exist today



#	Name	Symbol	Market Cap	Price
1	Bitcoin	BTC	$130,614,483,900	$7,217.80
2	Ethereum	ETH	$15,589,501,022	$143.18
3	XRP	XRP	$9,611,176,686	$0.222041
4	Tether	USDT	$4,149,632,878	$1.01
5	Bitcoin Cash	BCH	$3,760,231,388	$207.05
6	Litecoin	LTC	$2,799,464,202	$43.87
7	EOS	EOS	$2,433,406,956	$2.58
8	Binance Coin	BNB	$2,300,230,276	$14.79
9	Bitcoin SV	BSV	$1,712,594,306	$94.78
10	Stellar	XLM	$1,072,400,253	$0.053474
11	Tezos	XTZ	$1,045,408,692	$1.58
12	Cardano	ADA	$943,332,207	$0.036384
13	TRON	TRX	$941,480,177	$0.014119
14	Monero	XMR	$922,854,808	$53.19
15	UNUS SED LEO	LEO	$895,162,292	$0.895611
16	Chainlink	LINK	$760,929,642	$2.17
17	Cosmos	ATOM	$700,771,457	$3.67
18	Huobi Token	HT	$659,345,613	$2.73
19	NEO	NEO	$601,172,356	$8.52
20	IOTA	MIOTA	$548,419,195	$0.197306



22	Maker	MKR	$487,468,289	$487.47
23	USD Coin	USDC	$476,659,583	$1.00
24	Dash	DASH	$458,772,455	$49.84
25	Ethereum Classic	ETC	$437,982,799	$3.78
26	Ontology	ONT	$377,364,902	$0.592083
27	Crypto.com Coin	CRO	$354,331,561	$0.028905
28	VeChain	VET	$340,409,619	$0.006139
29	NEM	XEM	$318,590,385	$0.035399
30	HedgeTrade	HEDG	$316,690,402	$1.10
31	Dogecoin	DOGE	$267,317,646	$0.002184
32	Zcash	ZEC	$257,270,522	$31.98
33	Basic Attention ...	BAT	$249,147,055	$0.176581
34	Paxos Standard	PAX	$235,429,426	$1.00
35	Decred	DCR	$215,145,044	$19.95
36	Synthetix Netw...	SNX	$199,061,096	$1.33
37	Qtum	QTUM	$166,751,675	$1.73
38	TrueUSD	TUSD	$160,615,128	$1.00
39	0x	ZRX	$135,904,003	$0.224850
40	Centrality	CENNZ	$131,896,923	$0.123332
41	Algorand	ALGO	$131,445,164	$0.283688

BitCoin’s (original) goal

Pros/cons of cash

- ✓ Portable
- ✓ no need for trusted 3rd party
- ✓ anonymous
- x Does not work online
- x hard to monitor/tax
- x need government to print them

Pros/cons of credit cards

- ✓ works online
- ✓ ✗ can repudiate
- x requires trusted 3rd party
- x tracks one’s purchases
- x can prohibit some transactions
- x easy to monitor/tax/control

BitCoin: e-cash without a central trusted party

What’s hard socially/economically

- Why does e-cash have value?
- How to pay for infrastructure?
- What should be the monetary policy?
- What about laws? (taxes, money laundering, drugs, terrorists)

What’s hard technically?

- Forgery
- Theft
- Double spending

Cryptography background

- Public key crypto

  public key

  private key

  - Each key comes in a pair K, K^{-1}
  - e ← Encrypt(data, K)), data ← Decrypt(e, K^{-1})
  - $\{data\}_{K^{-1}}$ ← Sign(data, K^{-1})), verify(signature, K)
- Cryptographic hash function (e.g. SHA-256)
  - h_x ← Hash(x)

    256-bit integer

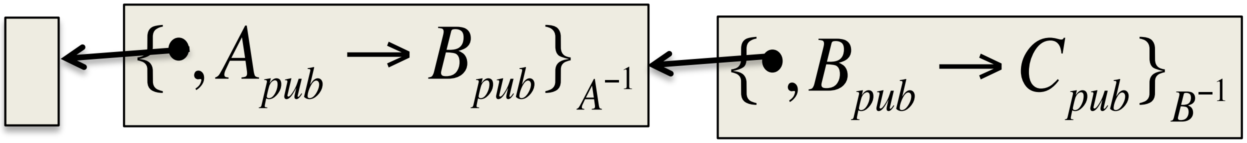
    X: A buffer of arbitrary length
  - Property:
    - deterministic: same input → same output
    - collision resistant: given h, it's highly unlikelyl 2^(-256) to find x' such that hash(x') = h = hash(x)

Key idea #1: Cryptocurrency

- Ownership of currency
  = possession of some private key
- Transfer of currency
  = signing "ownership" away to another party
- A "coin" is a transaction record
- T1: A transfers a coin to B $\{A \rightarrow B\}_{A^{-1}}$
- T2: B transfers the coin to C $\{B \rightarrow C\}_{B^{-1}}$
- How to ensure T2 is spending the same coin of T1? (i.e. how to link T2 to T1)

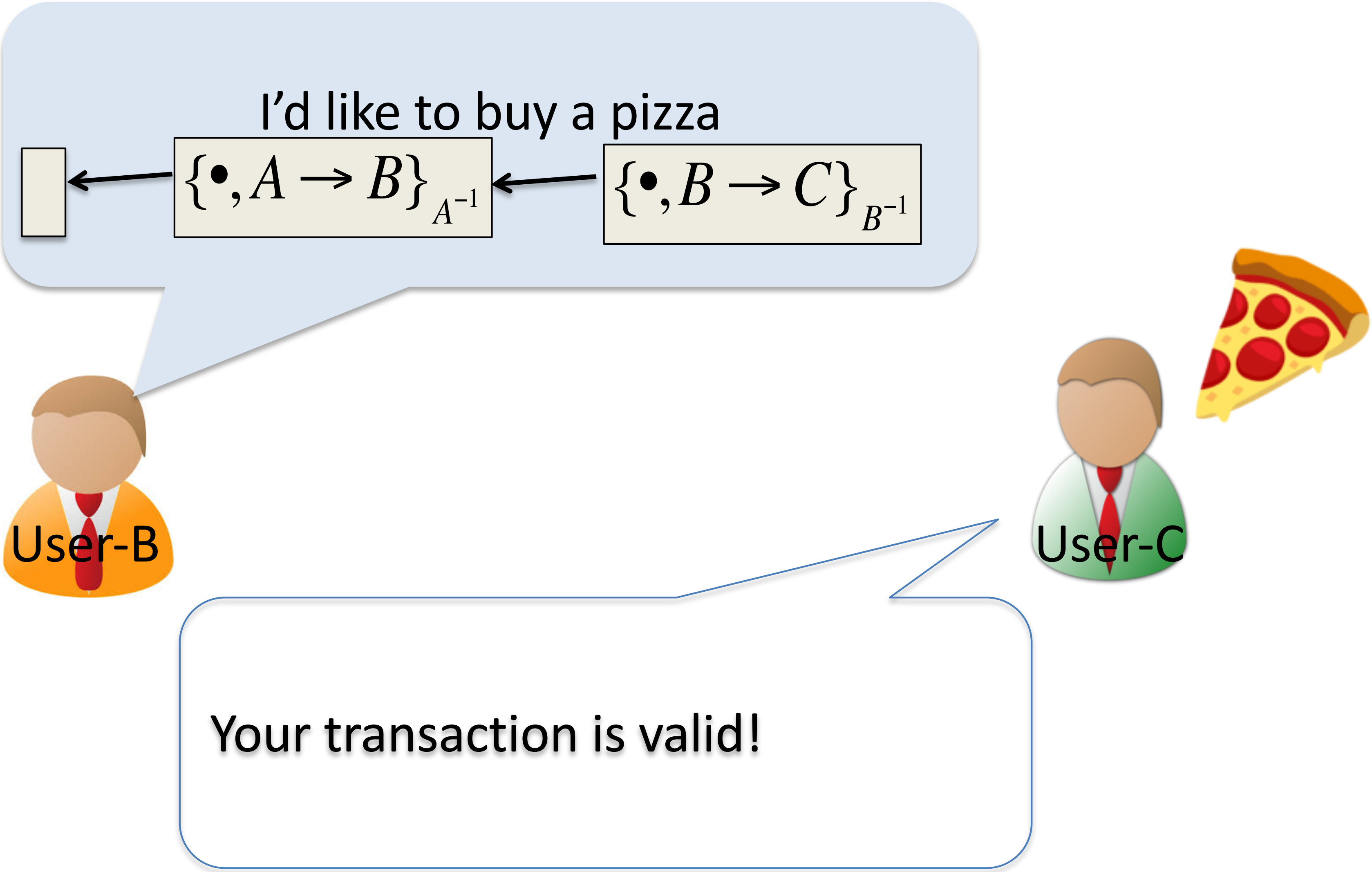
Key idea #1: Cryptocurrency

- Problem: How to link transaction records?
- Strawman: serial number
  - If T1, T2 contain the same serial#, then they refer to the same coin.
  - Problem: did T1 come before T2? or vice versa?
- Idea: a secure chain of transaction records
- T2: $\{hash(T_1), B_{pub} \to C_{pub}\}_B$



$\{\bullet, A_{pub} \to B_{pub}\}_{A^{-1}}$

$\{\bullet, B_{pub} \to C_{pub}\}_{B^{-1}}$



I’d like to buy a pizza

$\{\bullet, A \rightarrow B\}_{A^{-1}}$ ← $\{\bullet, B \rightarrow C\}_{B^{-1}}$

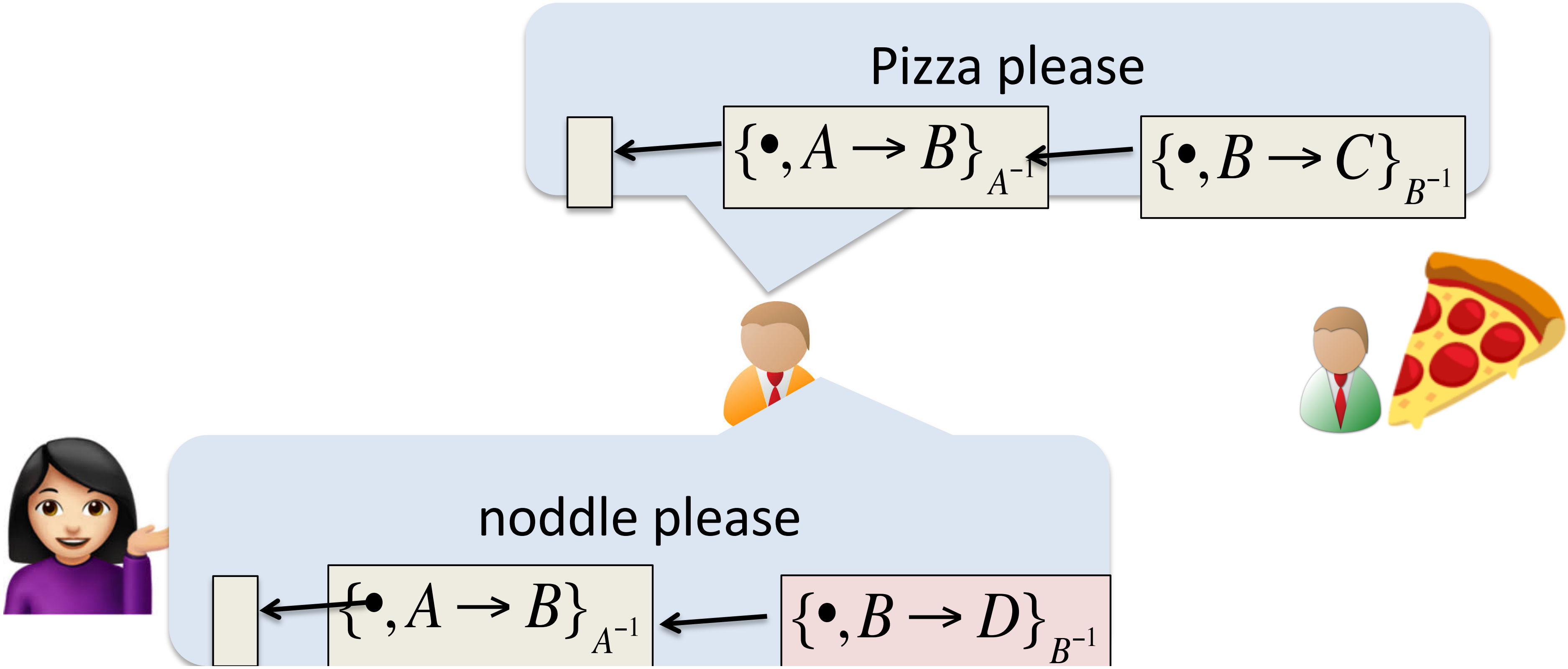
User-B

User-C

Your transaction is valid!

What’s hard technically?

- Forgery
- Theft
- ✗ Double spending



Pizza please

$\{\bullet, A \rightarrow B\}_{A^{-1}}$ $\{\bullet, B \rightarrow C\}_{B^{-1}}$

noddle please

$\{\bullet, A \rightarrow B\}_{A^{-1}}$ $\{\bullet, B \rightarrow D\}_{B^{-1}}$

How to defend against double-spending?

- Strawman: use a central trusted party (CP)
- Users submit all transactions to the CP
- CP verifies that no doublespending
  - User-B signs T2 and gives it to User-C. User-C asks CP to verify T2 before giving pizza to User-B.
  - Later User-B signs T3 to give the same coin to User-D. What happens?

✗ No longer peer-to-peer

Idea #2: Maintain a global log (ledger)

- All peers keep track of all transactions in a global log (“public ledger”).
  - Why log? (Why not a set?)
- Each transaction is replicated to all peers
- Forked log → double spending
- Problem: how to guarantee a non-forked log?
  - Can we run Raft/MultiPaxos among all peers?

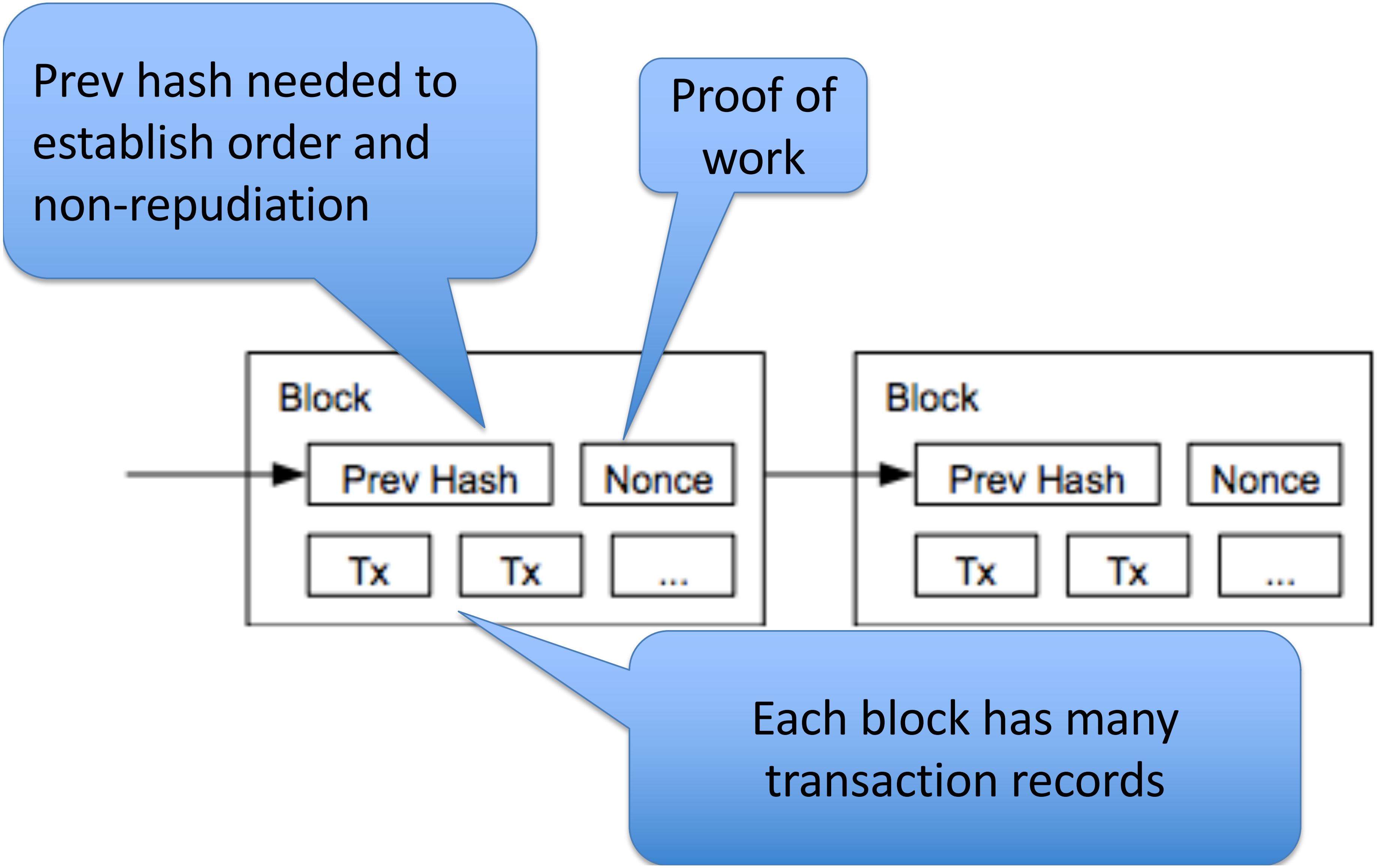
Why not use Paxos/Raft to maintain the global ledger?

- Paxos does not scale to 10,000 nodes
- Paxos is not secure against malicious nodes
  - There's a version of Paxos (PBFT, Castro&Liskov) that is secure if <1/3 nodes are malicious
- Vulnerable to Sybil attack
  - adversary joins the network with many identities so he controls >1/3 of all nodes

Idea #3: proof-of-work

- A peer can extend the log only after **provably** having done a lot of work.

The BlockChain



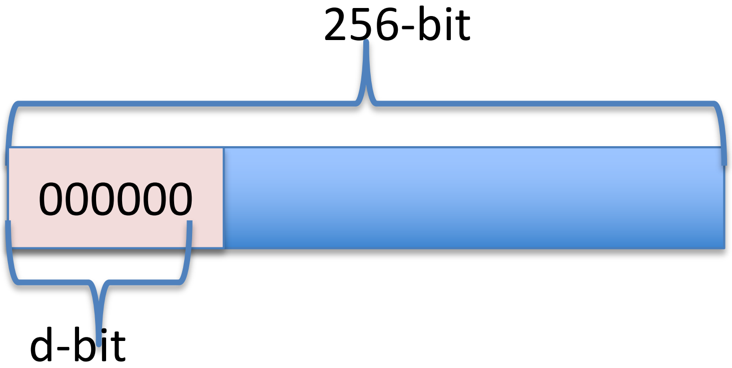
Prev hash needed to establish order and non-repudiation

Proof of work

Each block has many transaction records

The BlockChain: proof-of-work

- To extend the chain, peer needs to find nonce, s.t.:
- hash(block, nonce) =



- There's no better solution than brute-force
  - hash(block, 0) = ?
  - hash(block, 1) = ?
  - hash(block, 2) = ?
  - ….
- Running time? Difficulty= 2^d

How to recover from “fork”s

- Two peers might “simultaneously” find different legitimate next blocks → forks in the chain
- Resolved by taking the longest chain as the main blockchain
- Unlike Paxos, blockchain does not guarantee consensus
  - It’s okay to temporarily disagree as long as eventual agreement is reached in reasonable time.

Dealing with transient forks

- A valid block may be on a main branch or a fork...
- A transaction is confirmed only after its block is followed by 5 valid successor blocks.

How difficult should proof-of-work be?

- What if set to be too hard?
  - limited transaction rate
  - longer transaction latency
- What if set to be too easy?
  - Higher chances of forking the main chain→ lots of wasted blocks.
- BitCoin: difficulty is set so that it takes entire network 10 minutes to find the next block
  - ~5 blocks wasted per day
  - How long to confirm a transaction?

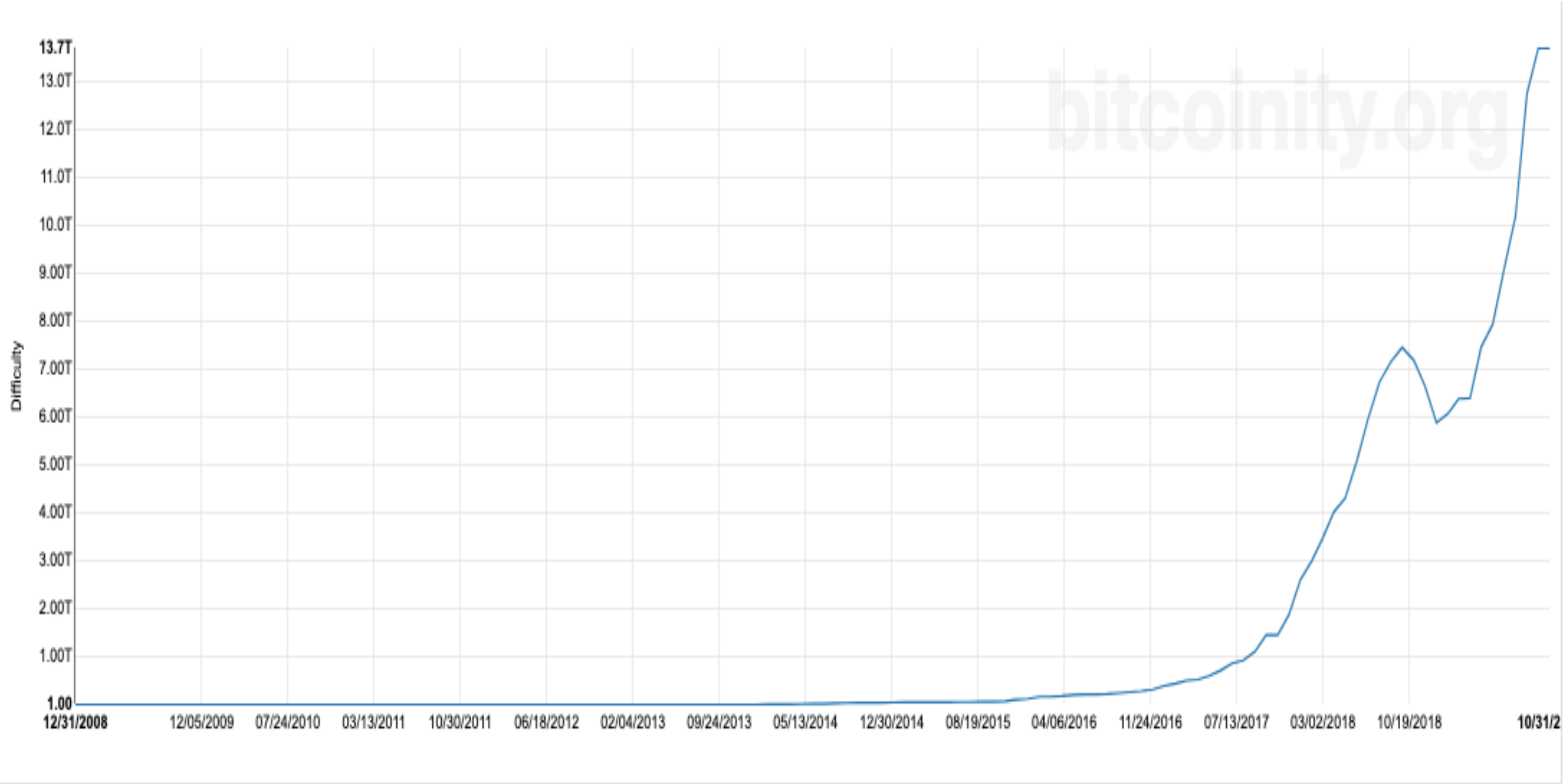
How hard should proof-of-work be?

- How do peers agree on difficulty for block #n?
  - More peers → harder for each peer
- For every 2016 blocks found, each peer sets the difficulty for the next (2016) blocks to be:
  - 2 weeks / T

Time taken to find the prior 2016 blocks,
according to their timestamps

- BitCoin’s transaction rate? (1MB block size, avg. transaction size 150B)
  - (1MB/150B)/600sec = 11 transactions/sec

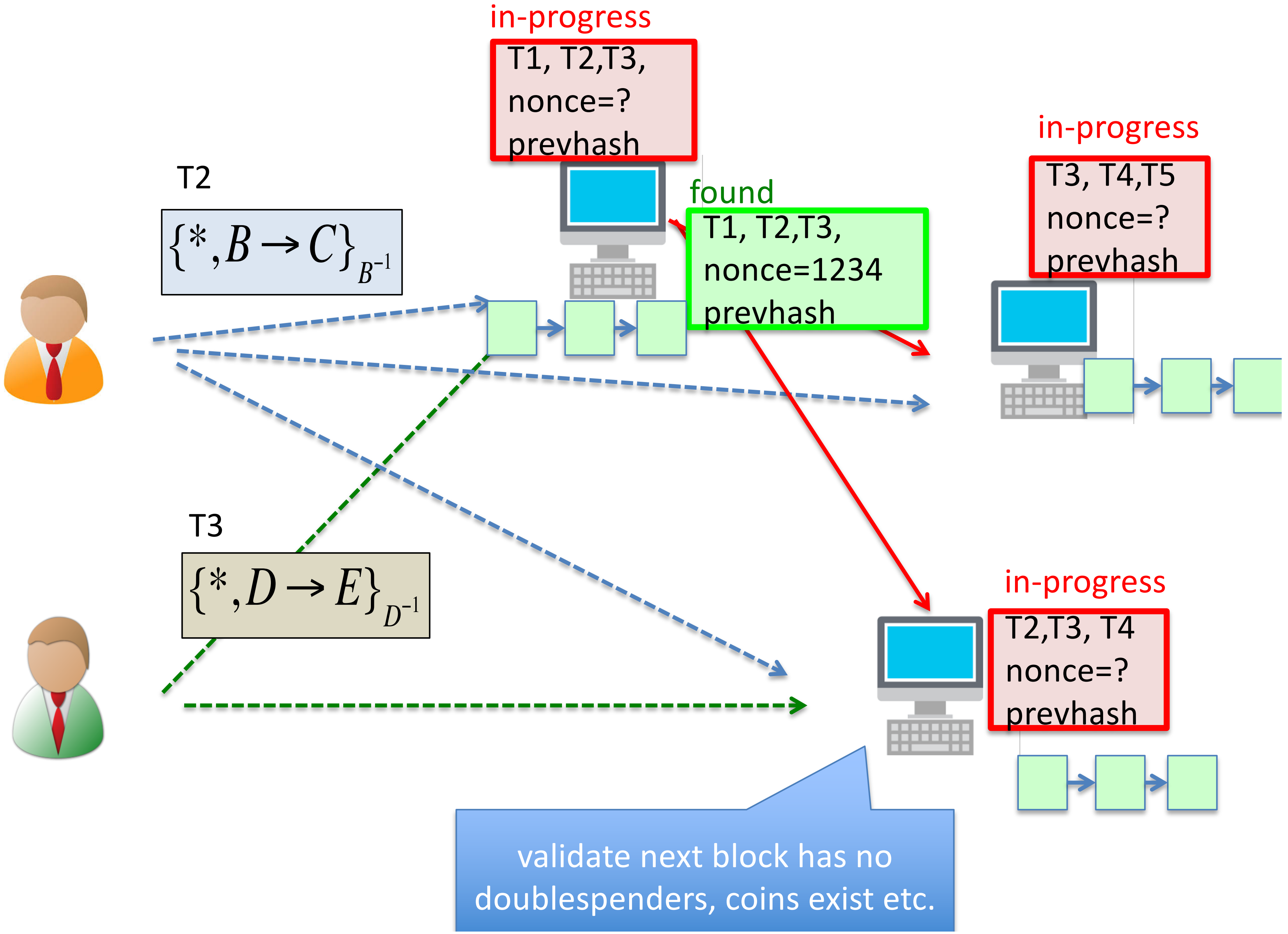
Bitcoin’s difficulty over years



Bitcoin’s incentives

- Why do people want to help with chain extension?
- Each new block contains a reward X coins, hence extending blockchain is called “mining”
  - this is how money gets minted
  - X halves every 210,000 blocks (~ 4 years), eventually stops after ~21 million coins
  - Currently x=12.5
- Miners charge users a transaction fee to include their transaction in the next block

The overall process



in-progress

T1, T2,T3,
nonce=?
prevhash

T2

$\{*, B \to C\}_{B^{-1}}$

found

T1, T2,T3,
nonce=1234
prevhash

in-progress

T3, T4,T5
nonce=?
prevhash

T3

$\{*, D \to E\}_{D^{-1}}$

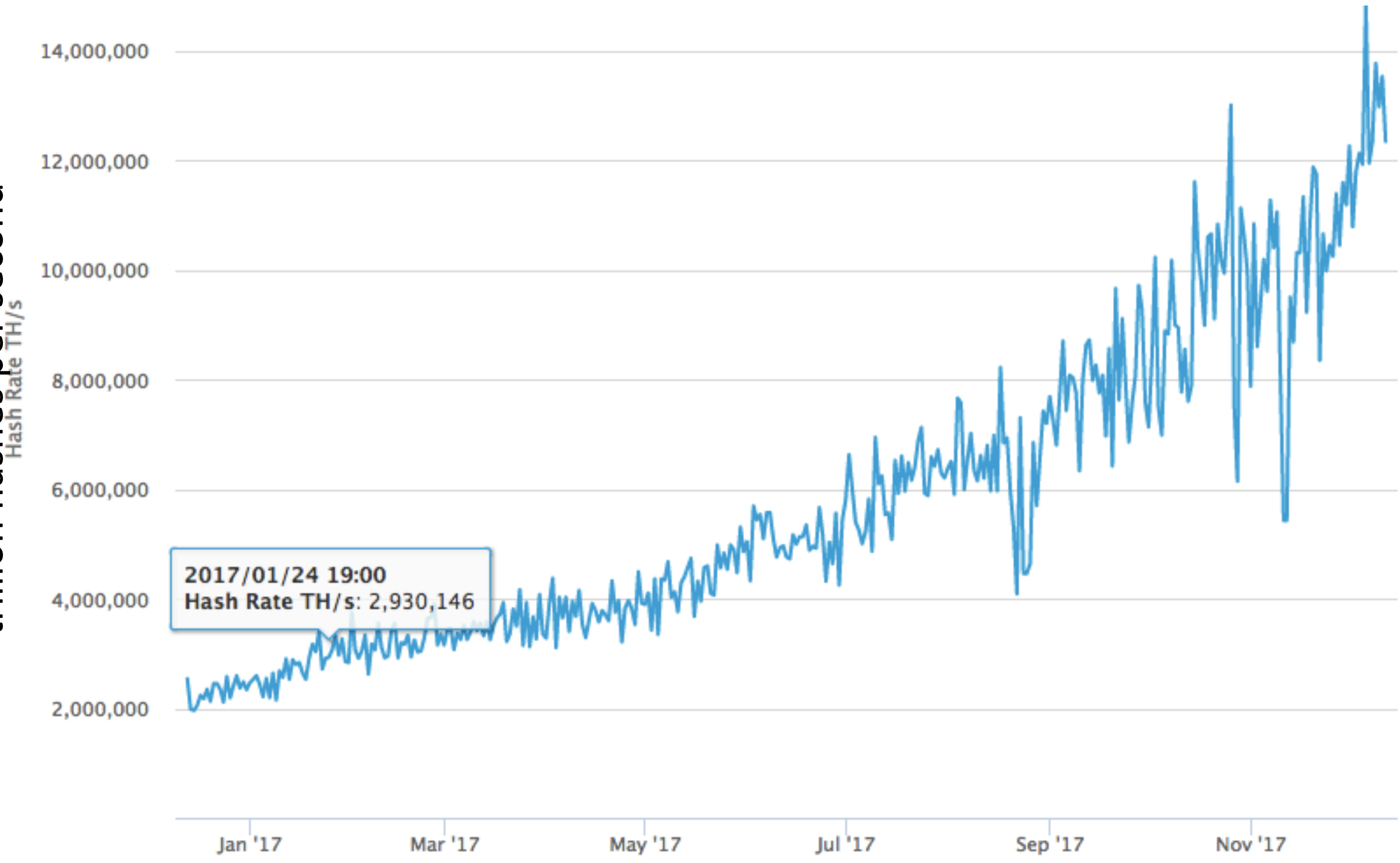
in-progress

T2,T3, T4
nonce=?
prevhash

validate next block has no doublespenders, coins exist etc.

Shall I become a BitCoin miner now?

trillion hashes per second



Intel core i7: 24MHashes/sec

top-of-theline GPU: 1GHashes/sec

ASIC: 1000 GHashes/sec

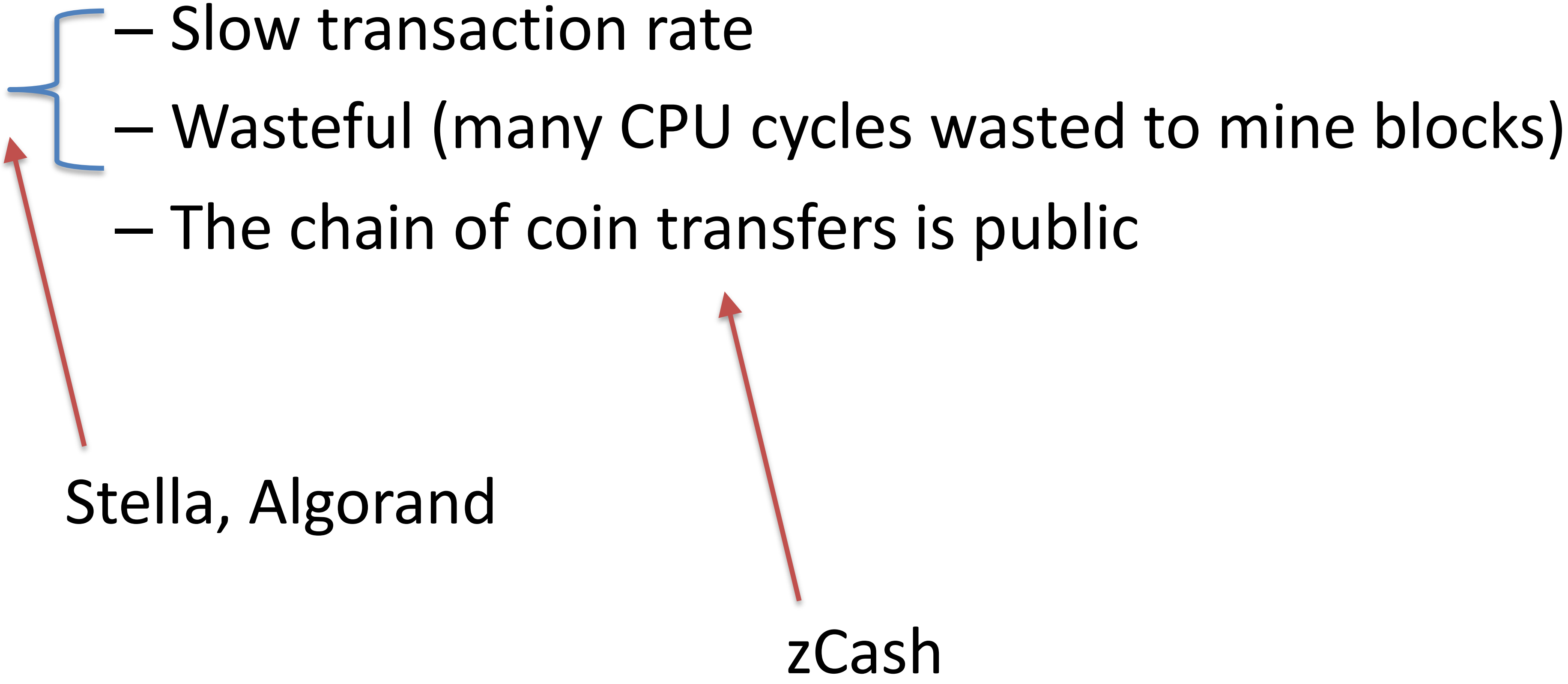
Can Bitcoin scale well?

- Size of ledger grows over time
  - currently at 253GB
- Cost of signature checks substantial
- Need to go back to very old blocks to check validity of coins

Has BitCoin succeeded?

- In replacing cash/credit cards?
- Downsides of Bitcoin vs. cash
  - no true anonymity (ledger is public information)
- Downside of Bitcoin vs. credit cards
  - no disputes
  - no loss/recovery
- ✗ Transactions take a long time to confirm.
- ✗ With the soaring price, transaction fee is high ($20 in early 2018)

Alternative Cryptocurrencies



- BitCoin’s main problems:
  - Slow transaction rate
  - Wasteful (many CPU cycles wasted to mine blocks)
  - The chain of coin transfers is public

Stella, Algorand

zCash

Algorand’s approach at a high level

- Overall idea: Use Byzantine Agreement to agree on a ledger
  - BA avoids forking under certain assumptions
    - > 2/3 users are honest
- Challenges:
  - (Security) How to be resilient against Sybils?
    - Controlling >1/3 users is easy if an adversary can create arbitrarily many pseudonyms
  - (Scalability) How to make BA scale?
  - (Availability) How to defend against targeted attacks?

Algorand uses proof-of-stake

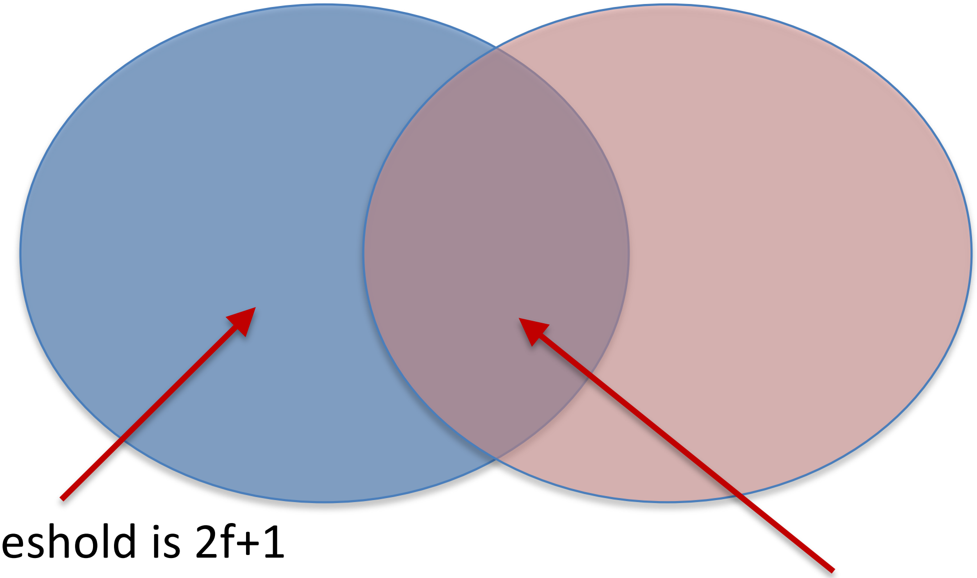
- Money as “weights”
- PKs associated with weights = relative fraction of money
  - Weights = # of votes a node can cast in BA
- Proof-of-stake is resilient to Sybil attacks
  - Attacker has to split wealth between pseudonyms
  - Total weights do not change by adding more pseudonyms

Algorand scales BA by sampling

- In traditional BA, every node broadcasts → does not scale
- Algorand samples a random committee using weights
  - Sampling computation uses private key, produces a non-interactive proof
  - Selected users originate messages; others gossip

Scale BA by sampling

- How large should the committee be?
  - Need n >= 3f+1 participants to deal with f bad users
  - Traditional BA wait for 2f+1 votes on the same value
  - But selection is random!
    - No fixed n/f

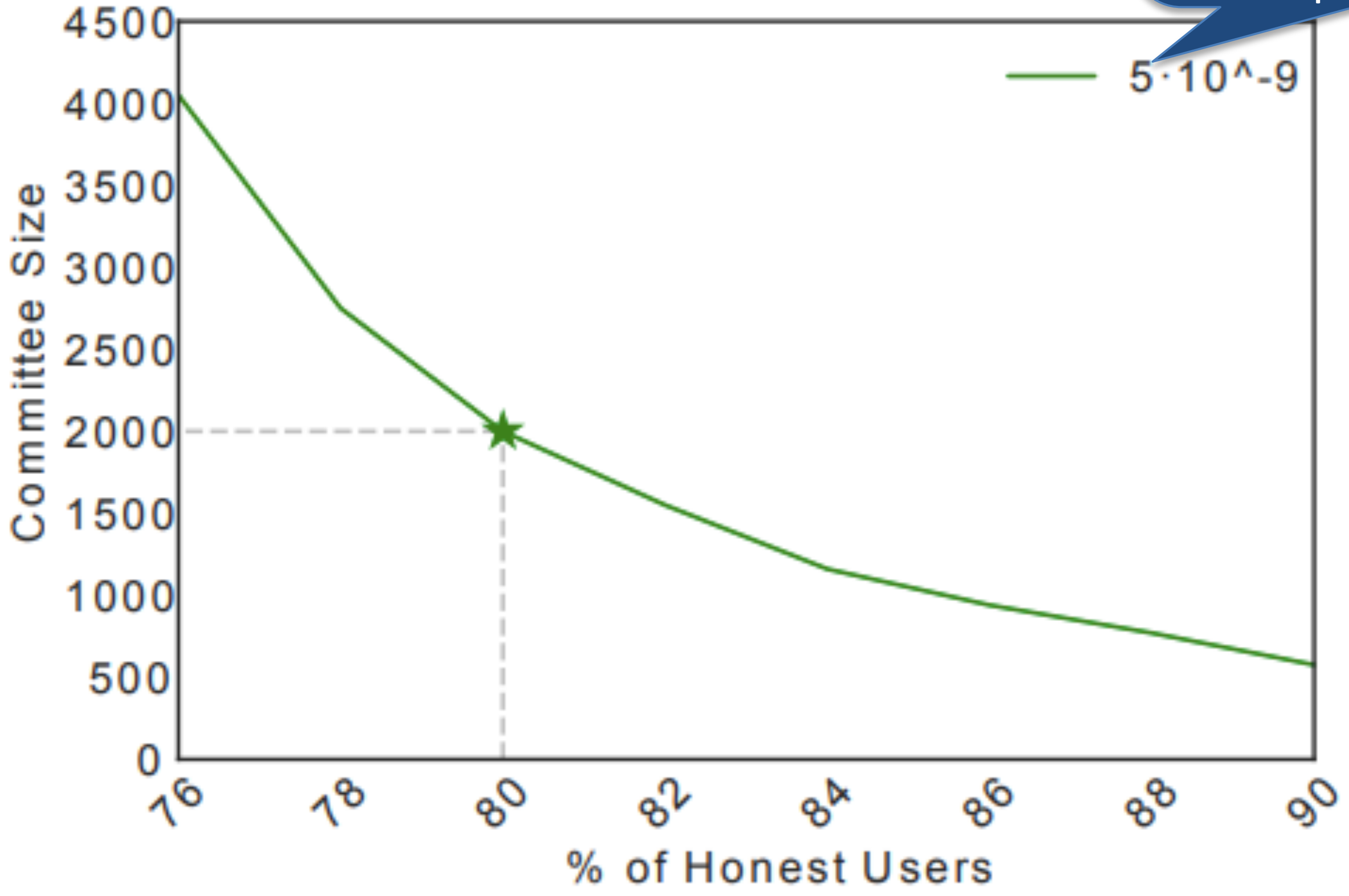


Vote threshold is 2f+1

Intersection must contain >=f+1 nodes for safety

Scale BA by sampling

- Algorand's threshold for votes



Probability of a committee contains >1/3 bad members for some step of the protocol

Want to learn more about cryptocurrency?

Take Prof Joseph Bonneau's cryptocurrency class next Fall.

Final Exam Logistics

- Open book, no laptop/ipads
- Cover topics from the entire semester
- Length and format are similar to midterm
- Practice materials:
  - Preparation questions
  - Last year’s final will be posted on Piazza