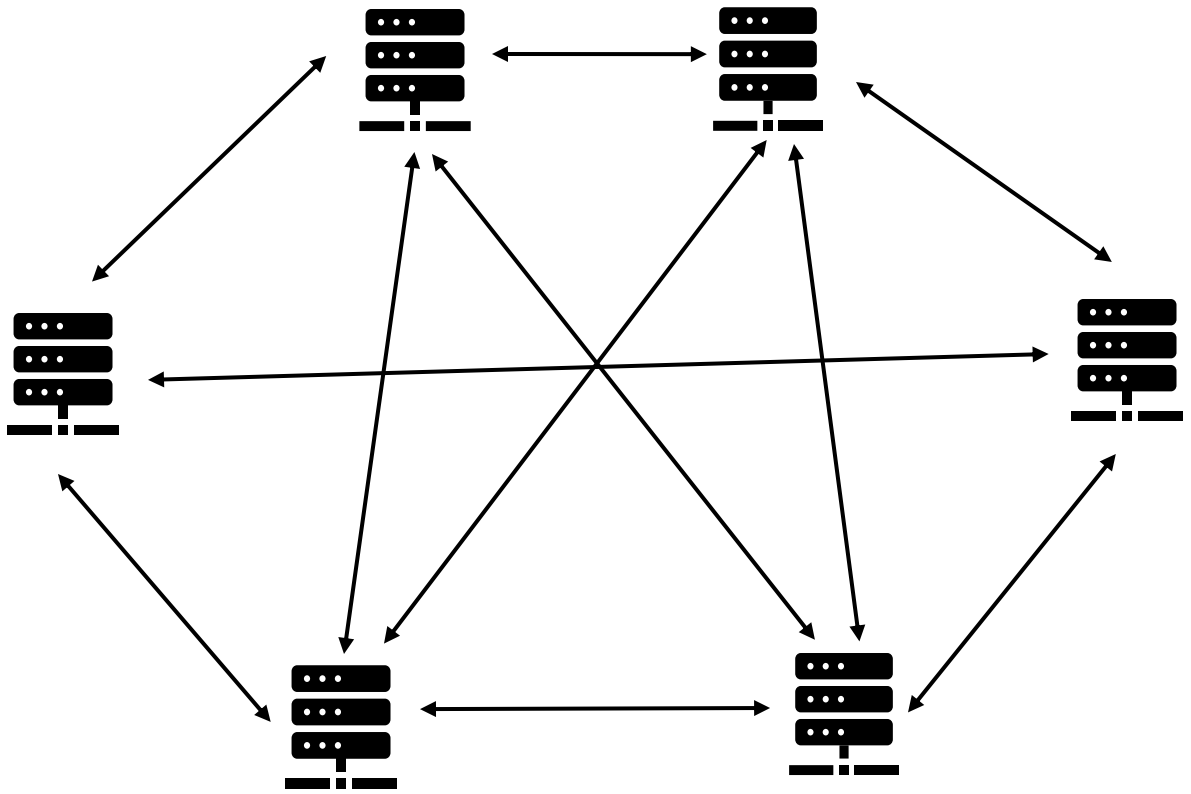
Nakamoto Consensus

CS839 – Kai Mast

Lecture Notes

- I assigned each of you for two lectures
- Please let me know if you have conflicts, or switch with another student
- Will send you invites for the gitlab repository soon

So far: Permissioned Blockchains



- Pre-defined set of participants
- All-to-all communication
- Can accommodate at most a few dozen members

Today: Bitcoin

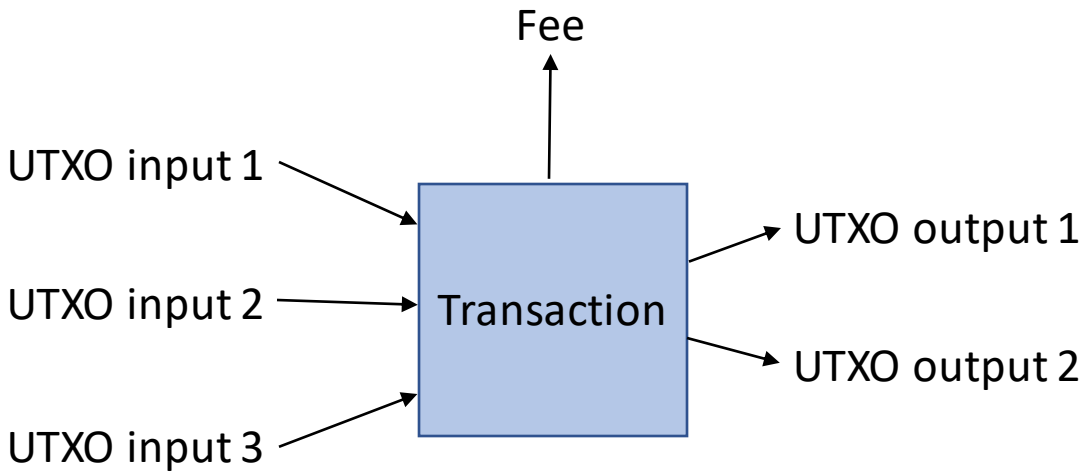
- Published as a white paper in 2008 under a pseudonym
- First fully decentralized digital currency
- Open membership: anyone can join or leave at any time
- But, more importantly, an entirely new type of consensus protocol
  - Useful for many applications besides cryptocurrencies

Satoshi Nakamoto
Pseudonym probably inspired by the Pokémon character Satoshi ("Ash" in the English version)

Proof-of-Work

- **Goal:** Prevent Sybil attacks
- **Idea:** Participants prove that they "wasted" some amount of compute cycles
- In Bitcoin PoW is used a mechanism for leader election
- Originally proposed to combat spam email [Dwork&Naor;1993]

The UTXO Model



- There is no notion of accounts (or wallets) in Bitcoin
- Instead, money is stored in the form of *Unspent Transaction Outputs (UTXOs)*
- Transactions consume and generate at least one UTXO
- Each UTXO has an owner (public key)

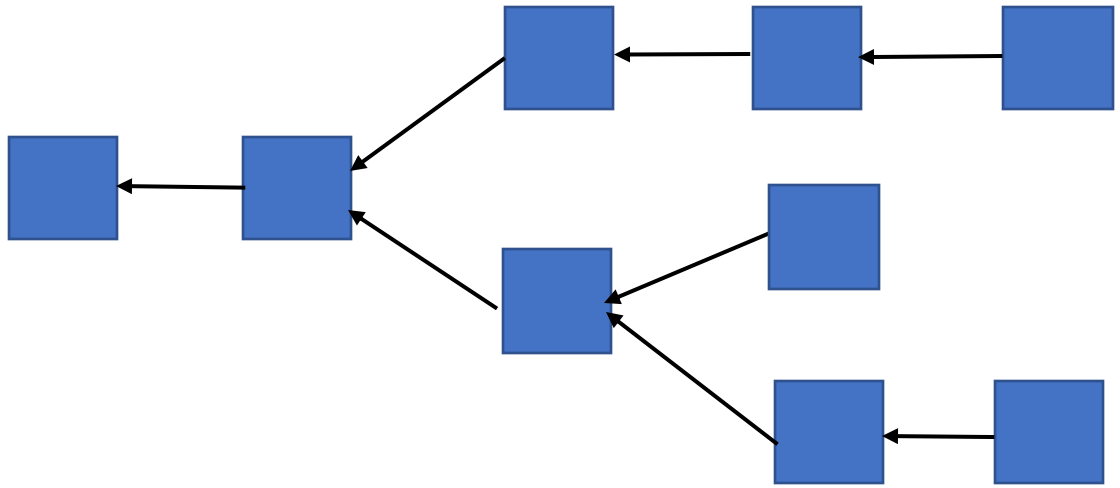
The UTXO Model: Why?

- Allows to maintain and store less chain on the blockchain
  - The only state is the UTXO set
  - Old UTXOs can be pruned easily

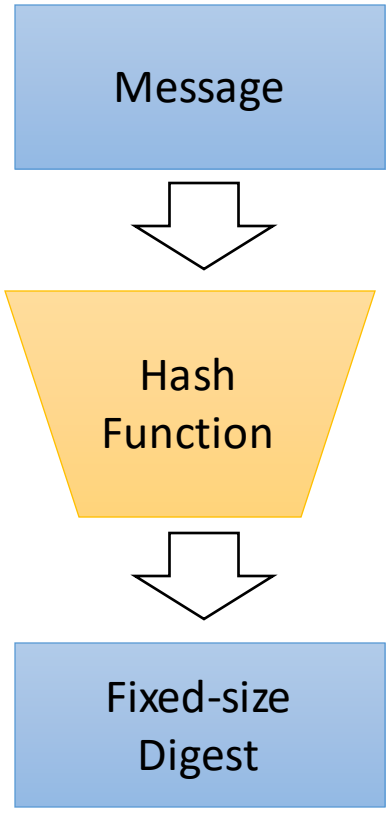
- But cannot easily store arbitrary state
  - Only allow a single "writer" per data item

The Bitcoin Blockchain

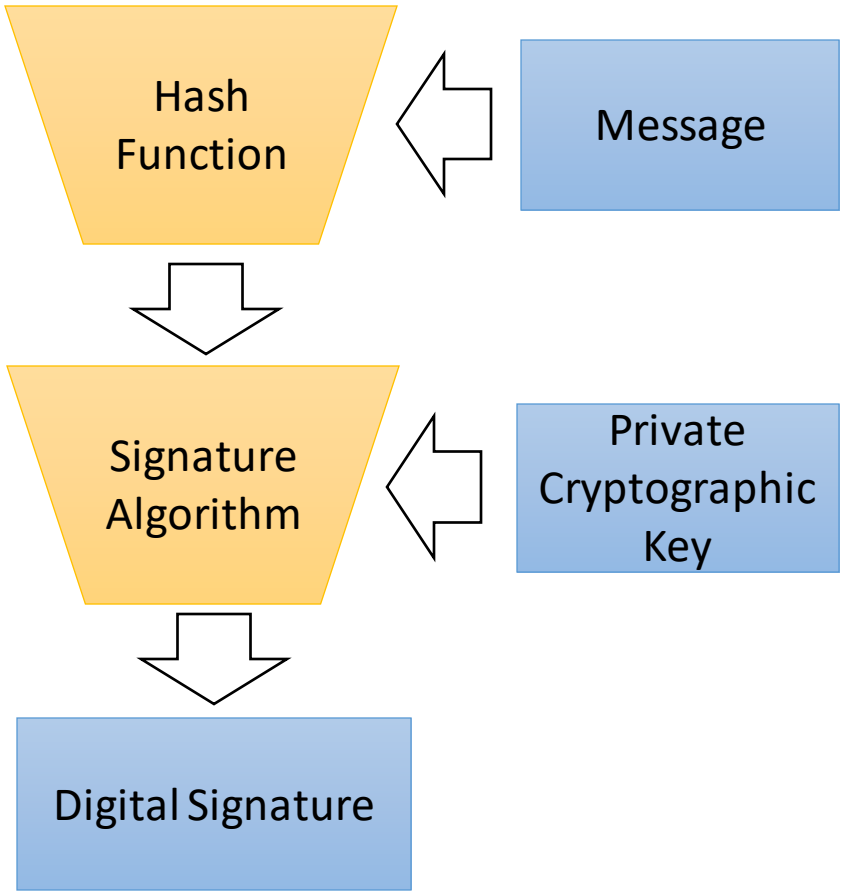
- Blocks contain a sequence of transactions
- Blocks are generated by miners
  - A miner is any party in the network that runs the PoW mechanism
- Every block has a single parent (except the g*enesis block*)
- But blocks can have multiple children leading to *forks*



Recap: Cryptographic Primitives

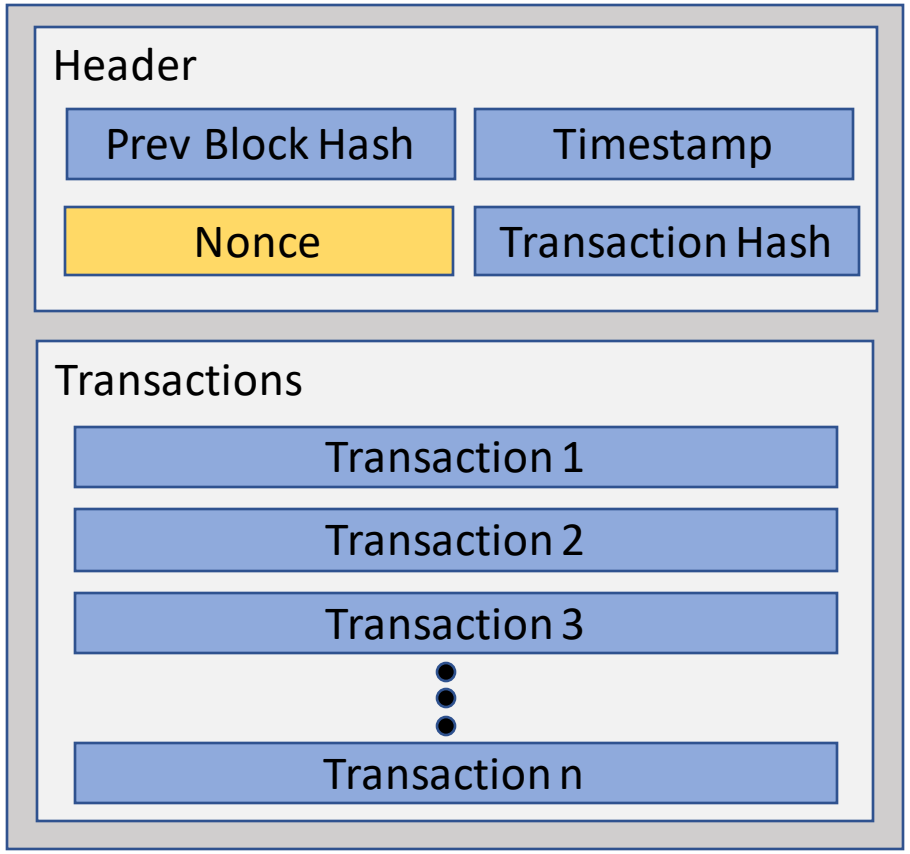


Hash Functions



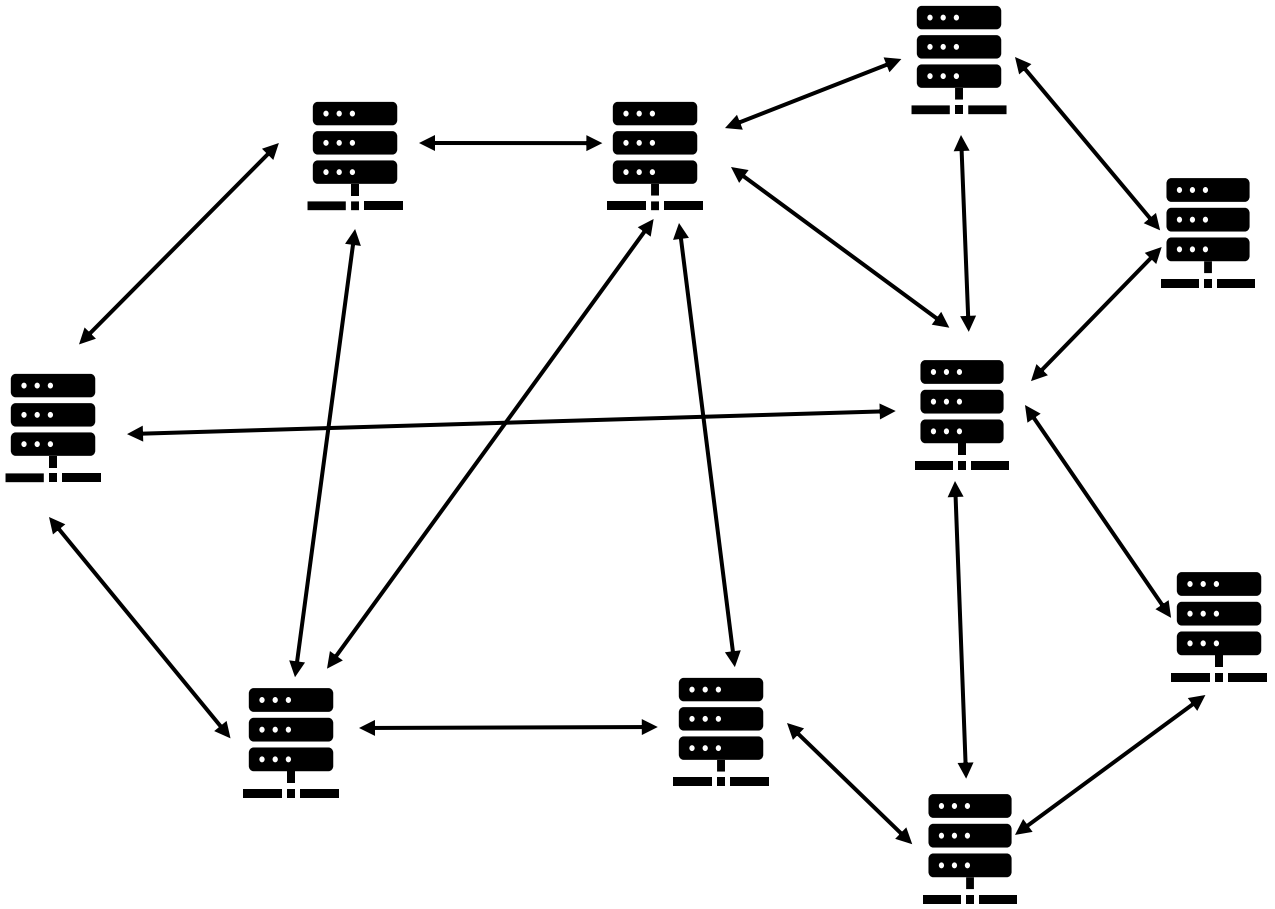
Digital Signatures

PoW in Bitcoin



- Miners pick a random nonce value
- For a block to be valid, hash of the block's header needs to be below some difficulty value
- Difficulty is adjusted as a function of the networks computing power
- Chance of mining a valid block *independent* of time spent mining

The Bitcoin Network



- Nodes are sparsely connected
  - Usually about 4 connections per node
  - No limit on network size
- Bitcoin uses epidemic/gossip protocols to spread messages across the network

Gossip Protocols 101

```
on_start(self):
  self.seen = set()

on_receive(self, msg):
  if msg.id in self.seen:
    return #ignore

  self.seen.insert(msg)
  for peer in self.peers:
    if peer.id != msg.sender:
      peer.send(msg)
```

Advantages:

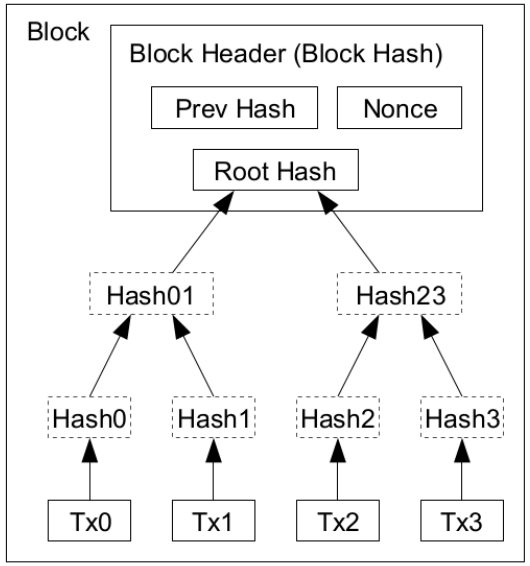
- We need no knowledge of the network topology
- Very resilient to network partitions and node failures

Disadvantages:

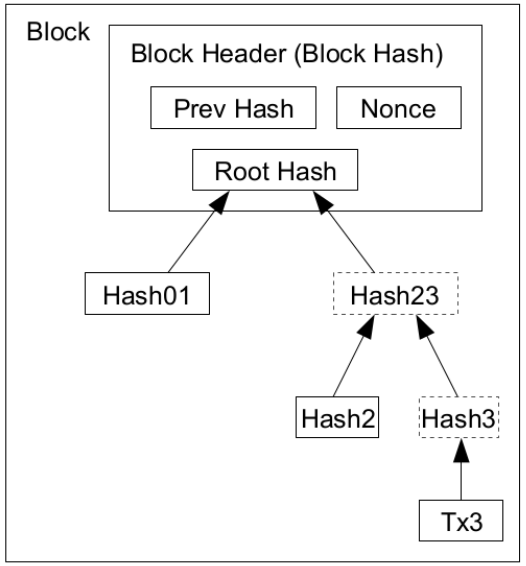
- Longer propagation delay
- More redundant messages

Pruning Old State

- We can discard transactions where all outputs are guaranteed to have been spent
- Partially pruned Merkle tree is still valid
- In reality, most nodes keep all transactions in cause new participants join the network



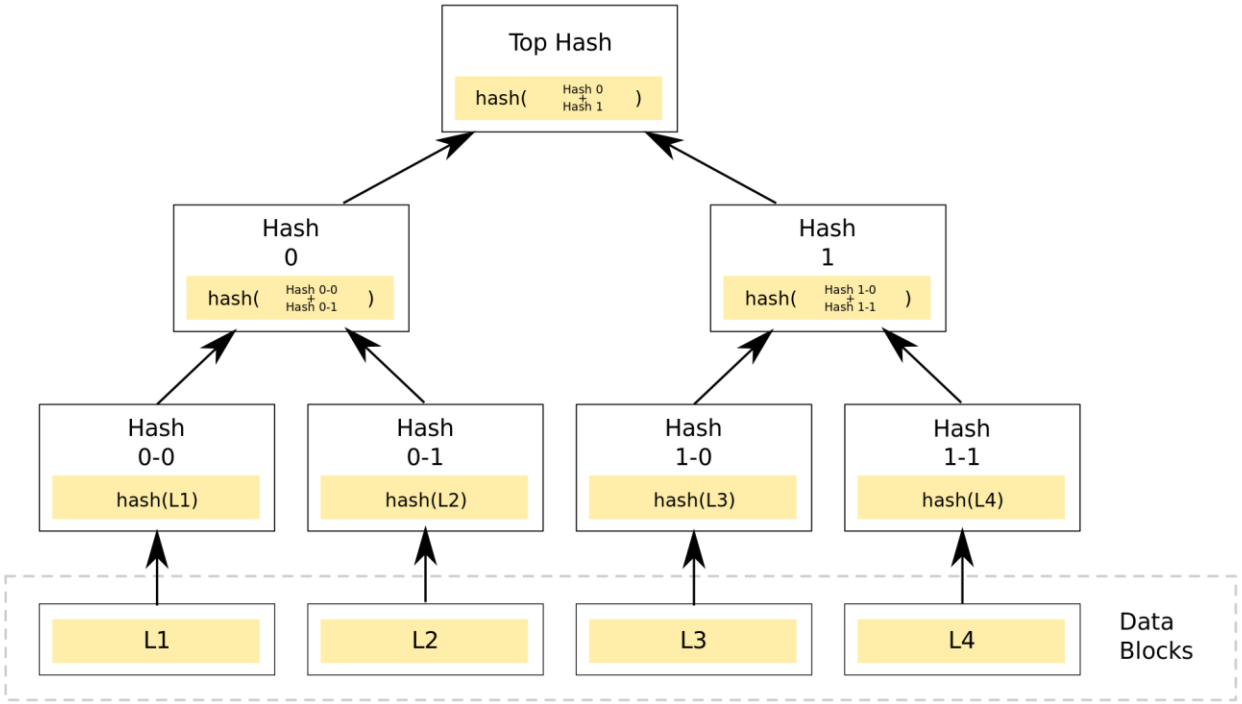
Transactions Hashed in a Merkle Tree



After Pruning Tx0-2 from the Block

Merkle Trees

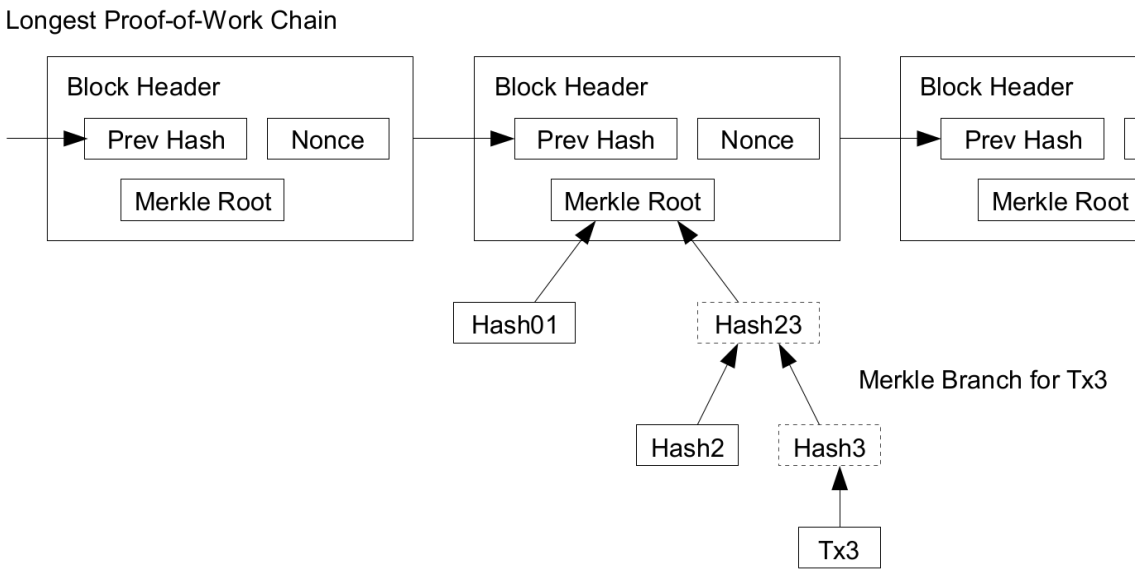
- Recursively "fold" a list of data items into hashes
- Only root of the tree needs to be stored on the blockchain
- Any "leaf" can be verified against the root using a Merkle proof
  - Remember it is virtually impossible to find a cryptographic hash collision



Source: Wikipedia

Simple Payment Verification

- We can participate in the Bitcoin Network even without storing the full state
- If a user provides a Merkle-proof of their payment, we only need access to block headers to verify it
- This is even more useful when dealing with off-chain protocols!



Nakamoto Consensus: Forks

What happens when two blocks are mined at (roughly) the same time?

- A fork is created: a block has two or more children
- Longest chain wins
- Miners pick the longest chain; or a random chain if they are equally long
- Eventually one chain will grow faster than others
- Other chains will be discarded (orphaned)

Nakamoto consensus is probabilistic protocol!

Nakamoto Consensus: Incentives

- Each block mines a certain amount of bitcoin (block reward)
- Additionally, transactions pay a fee to the block creator
  - Higher fee makes transaction more likely to be included in a block

Bitcoin's safety is directly determined by Bitcoin's value:

- The higher the reward the more power it is economical to spend on mining
- More mining power means higher resilience against malicious miners

Finality in Bitcoin

- Once a block is created, transactions are not immediately considered as committed/finalized
- We must wait until a transaction is "buried" deep enough into the blockchain, so that it being orphaned is very unlikely
- Usually, in Bitcoin, we wait about an hour

```
q=0.1
z=0     P=1.0000000
z=1     P=0.2045873
z=2     P=0.0509779
z=3     P=0.0131722
z=4     P=0.0034552
z=5     P=0.0009137
z=6     P=0.0002428
z=7     P=0.0000647
z=8     P=0.0000173
z=9     P=0.0000046
z=10    P=0.0000012

q=0.3
z=0     P=1.0000000
z=5     P=0.1773523
z=10    P=0.0416605
z=15    P=0.0101008
z=20    P=0.0024804
z=25    P=0.0006132
z=30    P=0.0001522
z=35    P=0.0000379
z=40    P=0.0000095
z=45    P=0.0000024
z=50    P=0.0000006
```

Solving for P less than 0.1%...

```
P < 0.001
q=0.10   z=5
q=0.15   z=8
q=0.20   z=11
q=0.25   z=15
q=0.30   z=24
q=0.35   z=41
q=0.40   z=89
q=0.45   z=340
```

"Figure" from the paper: q is the mining power of the attacker

Discussion: Network Asynchrony

What assumptions does Bitcoin make about the network?

- Bitcoin assumes a synchronous network!
  - Most people/protocols in the Bitcoin/Ethereum space assume a 5-minute upper bound on latencies
- If not, there might be a longer chain that has not been discovered yet

Discussion: Changing Block Size / Interval

Can we just increase the block size or block creation intervals for higher throughput?

Remember, forks appear when:

- Blocks are created at the same time
- Block are created before the miner has "seen" other new blocks

- Larger block sizes result in slower blog propagation
- Slower block propagation increases the likelihood of forks
- Higher block creation frequencies make it more likely for blocks to be created at the same time

That’s all for today

- Next time: Ethereum and Smart Contracts
- Please look at the schedule for lecture notes