EECS 491
Introduction to Distributed Systems

Bitcoin and Research at Michigan

December 5, 2019

Harsha V. Madhyastha
Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing
Digital Currency

- Say Bob pays for purchase with digital cash

- What properties must seller check?
  - Valid (i.e., not forged)
  - Not already spent
  - Owned by Bob (i.e., not stolen)

- How do shopping portals check these today?
  - Rely on bank, Visa, Mastercard, …
Eliminating Centralized Trust

● Imagine public ledger of transactions
  ◆ Every transaction (“A paid $X to B”) in ledger
  ◆ Signed with A’s private key, so not forgeable

● Benefits:
  ◆ Any user cannot spend more money than earned

● How to discover current state of ledger?
  ◆ Don’t want to rely on central service
Public Ledger

- Distributed system comprising 1000s of replicas
- **Broadcast** transactions; *valid if majority report*
- No money stolen unless private key leaks

- How to identify majority? Majority of IP addrs?
- What can Mallory do if she controls majority?
  - Export different views of ledger to different users
  - Double spending!
- **Sybil attack:** Same user runs many nodes
Desired Properties of Ledger

- **Strongly consistent**
  - All users must see the same set of transactions

- **Append-only**
  - Can only add transactions
  - Cannot remove transactions
Bitcoin: Key Idea

- Every node **must do work to participate**
  - Called mining

- To double spend, need to **control majority of CPU resources** in system
Do work by hashing

- Leverage one-way hash functions
  - Given $h$, hard to find $m$ such that $h = \text{hash}(m)$

- Bitcoin relies on partial collisions
  - $m$ whose hash has most significant bit is 0?
  - $m$ whose hash has most significant bits are 00?
  - Assuming output is randomly distributed, complexity grows exponentially with # bits to match
Structure of Bitcoin Ledger

- Each miner picks a set of transactions for new block
- Links to previous block by including its hash
- Pick nonce for header
Find nonce such that

\[
\text{hash (nonce || prev_hash || block data)} < \text{target}
\]

i.e., hash has certain number of leading 0’s

- At any time, all nodes in race to identify nonce for next block
- Target set such that new block every 10 minutes
- Incentive for any node to participate?
Randomized leader election

- Each time a nonce is found:
  - New leader elected for past epoch (~10 min)
  - Leader decides which transactions comprise block

- Probability of a node selected as leader?
  - Proportional to node’s % of global hashing power
Coping with Forks

- “Correct” nodes accept longest chain
  - Older a block, the “safer” it is from being deleted
- Common practice: Transaction 6 blocks deep “committed”
Scaling Bitcoin

- Scaling limitations
  - 1 block = 1 MB max (~ 2000 transactions)
  - 1 block every 10 mins → 3-4 txns / sec

- Visa peak load comparison
  - Typically 2,000 txns / sec
  - Peak load in 2013: 47,000 txns / sec

- Joining requires full download of ledger

- High energy consumption
Impact of Bitcoin

- Idea of public ledger (called blockchain) widely applicable
  - Remove dependence on centralized trust

- Impact of using blockchain to create digital currency
  - Time will tell …
Announcements

- **Final exam:** 10:30 – 12:30pm on Dec. 13th
  - Will post room assignments on Piazza
  - Project 4 and material after Fall break, but all material fair game
  - Case studies: Focus on techniques used
- **Take sample exam before next Tuesday**

- **Submit teaching evaluation**
Topics after Midterm

- **Partitioning state** (consistent hashing, DHTs)
- **Distributed transactions** (serializability)
  - Two-phase locking and optimistic concurrency control
- **Azure storage**: Making P/B replication scalable
  - Client-driven view change, append-only log
- **Spanner**: Efficient transactions at global scale
  - 2PL with Leader-based Paxos, Snapshot reads, TrueTime
- **Performance at scale**:
  - Look-aside caching, Hedged requests