Updates

• A3 due Sunday

• Example client code was corrected (see @220)

• Send Finn your A3 trace.json files (see @232)

• Jaafar is leaving us: his office hours end this week

• Two TAs joining over next two week. Their office hours schedule TBD

• A4 will not be released until after the reading week
Key ideas: Concepts

- Proof of Work (PoW)
  - Crypto puzzle - originally invented for SPAM email

- Blockchain (Dist. Ledger)
  - Ordering on operations (txns)
    - Read/Write
    - Shared State

- P2P + Byzantine fault model
  - Arbitrary peer behavior

- Eventually Consistency?
  - If you wait long enough
    - Then everyone will observe same state

Alternative
Proof of State (PoS)

Alternatives
Private Blockchains (not open)
Intercepted: Man in the middle

Alice → Bob

"I give 1 BTC to Bob" → msg not signed: if Alice has a BTC

Charlie → Double Spending

↓

Alice → Bob → Bob can check that msg is really from Alice

\[ \text{sign}_A(m) \]

× MIM: at most can replay the msg
× Double Spending still a problem

Charlie

1. \text{get (1 BTC)}
2. OxFF

1. \text{get (1 BTC)}
2. OxFF

Alice → Bob

3. \text{sign}_A(m)

3. \text{sign}_A(m)

Bob

5. Ack

BANK

Central entity: provides accounting
Trusted
Provides uniqueness guarantees

4. Checks if OxFF valid
  + Alice has a BTC to give
Bank $\rightarrow$ Distributed P2P Context

"Make everyone the Bank" $\Rightarrow$ Bank is public/transparent

$\Rightarrow$ all peers in the system track the ledger of txns

A $\rightarrow$ B

P2P network $\sim$ Bank

\[ \text{PoW} \]

\[ \text{Blockchain} \]

\[ \text{double spending} \]

\[ \text{Concurrency} \]

\[ \text{Incentives} \]

\[ \text{Reward P2P peers} \]

\[ \text{Trust} \]

Assumptions about

majority of nodes

non-malicious

\[ C \xrightarrow{tx_2} B \]

\[ tx_1(A \rightarrow B) \]

\[ tx_2(A \rightarrow C) \]

A $\rightarrow$ B

\[ \text{Sybils} \]

\[ "txn committed" \]

if majority of

P2P netw.

know about it

Any two [majorties] overlap

Requires to

Know the

# of nodes in

system

\[ \Rightarrow \]

Easy to Join

\[ \downarrow \]

Easy to create

"Sybils" by 1 person

\[ \Rightarrow \]

Sybil Attack
Proof of Work (PoW)

1. Make validation of txns in the network "difficult" (Why? A: Sybil attacks)
   ⇒ You need real physical resources (CPU cycles for computing PoW)

2. Incentives for nodes to compute PoW
   - Reward for solving a PoW ⇒ # of BTC
   - Scales with amount of CPU cycles

3. Transactions come with a fee that is given to a node that "validates" it using PoW

```
txn_1, ..., txn_n
\node{"miner"}

txn_1: Identity (public key)
nonce of miner

Block```

M1 Check txn_i valid (consistency check)
M2 Solve a cryptopuzzle (PoW)
\[ h = \text{sha-256 hashing fn.} \]
Find a nonce value s.t.
\[ h(\text{Block}) \leq \text{target value} \]
\[ i.e., h(\text{Block}) = 0x00...00\text{SAF42...} \]

Key Conditions for PoW
1. Difficult to find nonce
2. Easy to verify the nonce
Mining generates reward to miner (in BTC form)

⇒ Race between miners to mine blocks ⇒ Mining pools for cooperation

Miners have to balance

# of 

transactions in a block

with the fact that other miners are already mining

Select some

# of 

transactions

(Bound on block size)

BTC reward is generated until ~2140

After 2140

Mining is incentivized using only transaction fees

**Missing:** Ordering of transactions

\[ +xu_1 \leq +xu_2 \]

To mine a block, a miner has to pre-commit to where the block will go in the blockchain

\[ h(B_1) \\ +xu_1 \\ nonce \]

\[ h(B_2) \\ +xu_2 \\ nonce' \]

\[ h(B_3) \\ . . . \]

\[ h(B_{21}) \]

Blockchain (actually a tree)
Miners < Work along the longest chain (that they know)
Keep track of all forks (the entire tree) < Race cond. in mining

In short term "longest chain" is unclear < Network latency
But... in long term "longest chain" is stable < Network connectivity

⇒ txn is not "confirmed" Unless
  1. txn is on longest chain ⇒ Essential for total order
  2. Must have 5 blocks that follow it ⇒ heuristic
      "6 confirmations"

Implications:
1. Blocks are immutable: "ledger" ⇒ Append Only
2. Difficult to create a fork
   + Convince network to follow it
     ⇒ Requires maj. 1/3 CPU power

\[ \text{txn}_A \leftarrow \text{txn}_B \leftarrow \text{txn}_C \leftarrow \text{txn}_D \leftarrow \text{txn}_E \leftarrow \text{txn}_F \leftarrow \text{rest of network} \]  
   Need to mine
   a longer chain than the network
   \[ \text{txn}_A \text{ and } \text{txn}_B \text{ conflict: } "\text{double spend}" \]
Bitcoin Overview

1. Flooding Txns
2. Mining process
3. Flooding Blocks (that include txns)

1. Validate txns
2. Generate blocks

The End