Distributed Systems for Information Systems Management

Bitcoin and Blockchain
Central Principle

• Give the good guys easy problems to solve and give the bad guys hard problems to solve.
Primitives

• SHA-256
• Signatures
• Tamper evident Merkle Trees
• Block chain
• Proof of Work
Satoshi Nakamoto

• Person or persons who wrote the bitcoin paper and wrote the bitcoin code in 2007 – 2010
• The transaction in the genesis block reads: “The Times 3 January 2009 Chancellor on brink of second bailout for banks”
• Speculation on who he/she/they are continues
• Underlying Bitcoin is blockchain
Blockchain

- Q’s parent is P. Each block refers to its parent block.
- The parent block is the previous block added to the chain.
- Each block contains the hash of its parent inside its own header (tamper evident).
- The genesis block has no parent. It does have 50 bitcoins (initial money supply).
- Each header has a Merkle root summary of ~ 500 transactions.
- Block identified by its hash or its height. Genesis has height 0.
- Each block has a proof of work.

Change T forces a change to P and Q. Proof of work needs recomputed too.
Blockchain

Block 12 contains the hash of Block 11

Make a modification to Tx1 and describe the state of the system.

From Wikipedia
Proof Of Work

• Consider a difficulty of 2:
  
  ```
  int i = 0;
  h = SHA-256(someData + i);
  while(two leftmost hex digits of h are not 00) {
      i = i + 1;
      h = SHA-256(someData + i);
  }
  ```

• On average, how many times will this loop execute?
• Prob(“00”) = 1/16 * 1/16 = 1/256. 256 trials expected.
• Hard to compute proof of work but easy to verify.
• The difficulty can change. We can force this to take 2 weeks on average.
• If computers become faster we can increase the difficulty
• Country of Ireland burns electricity at same rate as bitcoin.
Version 1

- Alice signs a message and sends it to Bob
- $A \rightarrow B \quad \{A \text{ gives } B \text{ 1 } BC\}K_{APriv}$
- Positive: Alice shows intent
- Positive: Alice cannot send a payment signed by Carol. Alice does not know Carol’s $K_{CPriv}$
- Problem: Replays from Alice are meaningless. Does Bob now have several coins or 1?
- Problem: Bob could make 5 copies. How many coins does he have now?
Version 2

• Alice signs a message and a serial number and sends it to Bob - and the serial numbers are unique.

• A -> B \{A gives B 1 BC, 123990245\}K_{APriv}

• Positive: Alice shows intent

• Positive: Replay not of concern with Bob.

• Problem: How does he know she has the money to spend?

• Problem: She can double spend with Charlie.
  A -> C \{A gives C 1 BC, 123990245\}K_{APriv}

• Problem: Where do we get all these unique serial numbers?
Version 2 With a Bank

- Alice gets a unique serial number from the bank
- Alice signs a message and the serial number and sends it to Bob
- $A \rightarrow B \quad \{A \text{ gives } B \ 1 \ BC, \ 123990245\}K_{APriv}$
- Bob checks with the bank before providing service to Alice. Does Alice have this to spend?
- The bank maintains a ledger of who owns what.
- Positive: No replays – everyone checks with the bank before accepting
- Positive: No double spending
  - $A \rightarrow C \quad \{A \text{ gives } C \ 1 \ BC, \ 123990245\}K_{APriv}$
- Problem: Centralized model
Version 3 No bank – the community keeps track

- Alice signs a message and a serial number and sends it to Bob
- \[ A \rightarrow B \quad \{A \text{ gives } B \ 1 \ BC, \ 123990245\}K_{APriv} \]
- Bob verifies – he checks his copy of the ledger. Does Alice own the bitcoin?
- Bob accepts and Bob tells everyone about the transaction
- Everyone updates their own copy of the ledger
- Positive: Decentralized model
- Problem: Alice double spends with Charlie.
- Problem: How do the others fix their ledgers?
Version 4 Everyone Verifies

- Alice signs a message and a serial number and sends it to Bob
- $A \rightarrow B \quad \{A \text{ gives } B \text{ 1 BC, 123990245}\}K_{APriv}$
- Bob asks everyone to verify
- If enough verifications, Bob provides service and asks everyone to update their copy of the ledger
- Positive: Double spending will normally be detected
- Positive: Decentralized model
- Problem: Sybil Attack possible and allows the double spend.
Version 5 Everyone Verifies with POW

- Alice signs a message and a serial number and sends it to everyone.
- A -> All: \( \{A \text{ gives B} 1 \text{ BC, 123990245}\}K_{A\text{Priv}} \)
- The transactions are collected by each verifier.
- Verifiers maintain a queue of pending transactions.
- To verify:
  1. Check ledger for double spend.
  2. Perform POW (solve a puzzle).
  3. Announce block is verified.
- The first to do so, gets cash in a coinbase transaction.
- It is easy for others to check this result against the blockchain and check the POW. The POW selects a random node.
- Alice will likely not be the first to verify. POW makes a Sybil attack very expensive.
Try to double spend

- A -> All \{A gives B 1 BC, 123990245\}K_{APriv}
- A -> All \{A gives A 1 BC, 123990245\}K_{APriv}

Suppose both transactions are placed on the peer to peer network. Miners verify each transaction and then collect the pending transactions into blocks. An honest miner will not include inconsistent payments in their block. One of these double spend transactions will be rejected. The block of transactions includes a payment transaction to the miner. Miners really want that payment and want their block included on the blockchain. Miners attempt to solve the proof of work puzzle. On top speed hardware, it will likely take two weeks.
Try to double spend

After about 10 minutes, some miner produces a block. This miner won the race. Other miners receive the new block and verify transactions and the proof of work. It is easy to verify the proof of work and all transactions in the block. If the block passes all checks, it is added to the blockchain. Unless she controls a lot of nodes, Alice cannot predict the winner of the race to solve POW.
Transactions

- Many transactions live in the block’s Merkle tree.
- You do not spend from an account. You spend from previous transactions.
- Input address (points to the output of an earlier transaction)
- Output address
- Signed by owner of funds
- All of the input is consumed
Transactions

- Transactions are between pseudonyms: hash of public keys.
- $2^{160}$ possible addresses
- $2^{63}$ grains of sand on earth
- $2^{160} / 2^{63} = 2^{97}$
- For each grain of sand, we have $2^{97}$ addresses.
Task of Bitcoin Miners(1)

• Listen for transactions
• Validate transactions – signature is correct and no double spending
• Maintain blockchain and listen for new blocks
• Validate each block that you receive by validating each transaction in the block and checking that the block has a valid nonce.
Task of Bitcoin Miners (2)

- Assemble a candidate block from validated transactions.
- The candidate block will extend the longest known valid chain.
- Find a nonce for your block that makes it valid.
- Publish your block and hope it gets accepted.
- Profit if accepted.
Task of Bitcoin Miners(3)

• If two different blocks are mined and announced at around the same time, it results in a two block fork.

• So, which to build on?

• The default is to build on the block you heard about first.
Smart Contracts

- First proposed by Nick Szabo in 1996
- In Ethereum, decentralized application or dapp (Turing complete)
- An agreement between two parties that is stored on the blockchain. Setting the agreement in stone.
- Disintermediating the legal system
- We trust the blockchain.
- The contract has an ID and so do the parties involved.
- If P then Q.
Why might Blockchain (not Bitcoin) be Disruptive?

(Notes from Oracle)

• Parties who may not trust each other still need to engage in transactions.
• So, they commonly use a Trusted Third Party (TTP) to maintain a single ledger of the transactions.
• The TTP takes time, typically days.
• The TPP charges a fee per transaction.
• A private blockchain is fast and charges no fee.
Bibliography

• See video from Berkeley:
  https://www.youtube.com/watch?v=fgSvXFZ1GuU

• Read a paper:
  http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/