Lecture 3 Review - Centralized and Decentralized Cryptocurrencies

Rylan Schaeffer and Vincent Yang

April 19, 2016

Note: This lecture is based on Princeton University's BTC-Tech: Bitcoin and Cryptocurrency Technologies Spring 2015 course.

Agenda

- Final Project
  - Rudimentary cryptocurrency
  - Groups vs. Pairs vs. Single
  - Report on some study/article about Bitcoin/Cryptocurrencies
  - Other ideas
- Homework 3
- Lecture Review/Clarification
  - Why decentralize?
  - Problems to solve/How proof of work solves them
  - Mining/Motivations/Why it's important (Proof of work)
  - Shortest vs. Longest blockchain
  - Double spending
  - 51% attacker

Advantages to Centralization

- Automation:
  - Easily manage a large number of keys e.g. Mastercard Europe
  - Maintain secure infrastructure and improve operations/efficiency
- Centralized Monitoring:
  - Record everything that happens easily; brings transparency
- Centralized Policy
- Easily update and track keys
- Easily update cryptographic schemes - swap out algorithms

CentralizedCoin

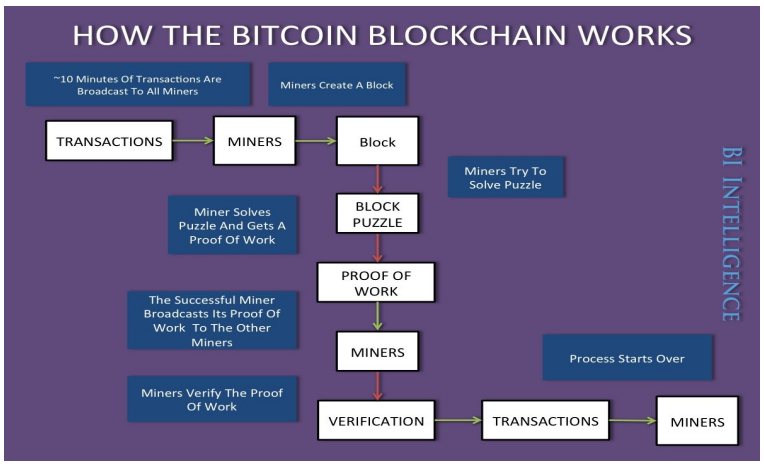
Decentralized Cryptocurrencies

- How does Bitcoin deal with Decentralization?

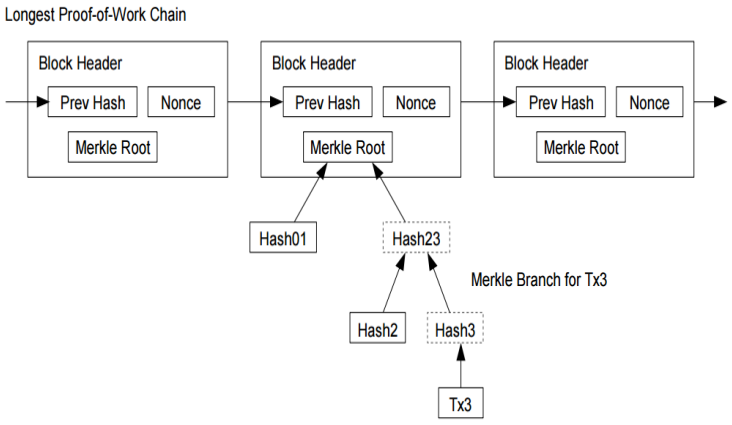
  Problems to address, since it's no longer centralized

  1. Who maintains ledger of transactions?
  2. Who determines which transactions are valid/invalid?
  3. Who creates new coins
  4. Who chooses when rules change

Distributed Consensus (Solves 1 and 2)



Block



The public ledger (block chain) was started on Jan 3, 2009 presumably by Satoshi Nakamoto. The first block is called the Genesis Block. This is a special case in the source code.

Proof of work:

- Puzzle Friendliness: No solving strategy for finding $H(k|x) = y$ is better than trying random values of x.
- Select nodes in proportion to computational power
- *Give example of leading 0's*
- Difficult to compute
- Motivation to subvert the process (picking a hopefully honest node), so reward honest nodes
- Use bitcoins to incentivize honest nodes - mining. Reward only if it becomes legitimate transaction
- Every 210,000 blocks (4 years), block reward is cut in half. Geometric sum - 21 million bitcoins
- Nonce published as part of the block.
- Proof of work details
  - Mining: Hash function with nonce appended to beginning (ex: 0's)
  - HashCash - with SHA-256 (used twice)
  - Less than some value
  - Items used: version, prevBlockHash, merkleRoot, timestamp, bits, nonce
  - Selecting nodes based on processing power/proportional
  - Hash puzzles - to make blocks, it needs to find a nonce where
  - $H(nonce \mid\mid prev\_hash \mid\mid tx \mid\mid tx \mid\mid ... \mid\mid tx) < target$
  - Nonce: Random number that is only used once
  - Hash puzzle properties: difficult, parameterizable cost (10 minutes variable target), trivial to verify
  - 10 minutes: reduce inefficiency from having many blocks
  - $meantimetonextblock = \frac{10minutes}{fraction\ of\ hash\ power}$
  - proof of stake - proportion to ownership of currency (used in other cryptocurrencies)
  - http://www.righto.com/2014/09/mining-bitcoin-with-pencil-and-paper.html
- Difficulty: what network hash rate results in a given difficulty?

  Currently, the entire network of miners makes about 30 trillion attempts a second.

  Difficulty is adjusted every 2016 blocks based on the time it took to find them.

  At the desired rate of one block every 10 minutes, 2016 blocks should take exactly 2 weeks to find.

  If the previous 2016 blocks took more than two weeks to find, the difficulty is reduced.

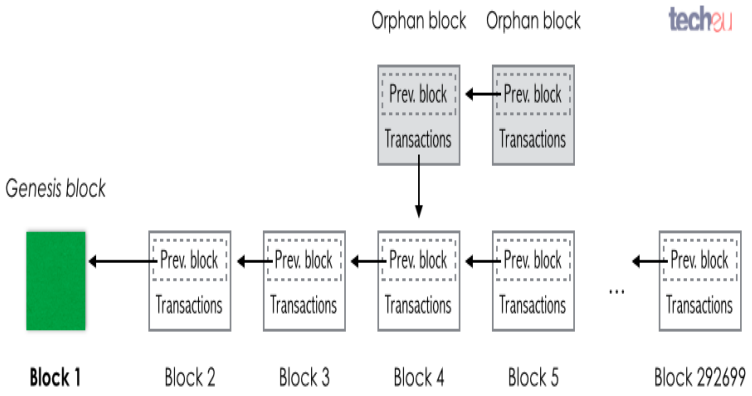
  If the previous 2016 blocks took less than two weeks to find, difficulty is increased.

  Hash is a random number between 0 and $2^{256\text{-}1}$.

  Since a lower target makes Bitcoin block generation more difficult, the maximum target is the lowest possible difficulty.

  Bitcoin Mining Pools

Double Spending



Say Alice pays Bob, and an honest node broadcasts this. Bob accepts that he's been paid. Alice then gets to broadcast her own transaction. She then makes a block with the *prevBlock* hash as the one before her payment to Bob. one of these blocks will be accepted.

Incentives: Solves 3

- Can't penalize, but can reward nodes for working correctly.
  - *Incentive 1*: Block Reward (25 BTC, halves every 4 years). (Solves 3)
    21 million max - block reward is how new coins are created; run out in 2140.
  - *Incentive 2*: Transaction Fee
    Incentive to have your transaction verified
  - Remaining problems:
    How to pick random node
    How to avoid free-for-all rewards
- The miner gains if reward > cost

  reward = block reward + tx fees

  mining cost = hardware cost + operating costs
- Aspects of decentralization in Bitcoin
  - Peer to peer network, open to anyone, low barrier for entry
  - Mining is open to everyone
  - Updates to software - core developers trusted by the community who have a lot of power (Solve Problem 4)
- New Problems from being Decentralized and Attacks:
  - Blocks have a tendency to extend the block they hear about first
  - Orphan Block
  - Latency, not all nodes connected, internet connection, malicious nodes
  - Attacks:
  - Stealing - even if Alice gets to decide the next block, she can't steal because she has to create a valid transaction; can't forge signatures

  - Denial of Service - even if Alice never validates Bob's transactions, an honest node will eventually do so.
  - Sybil attack

    Can't gain more power by having more accounts

    Satoshi's original paper had 1 cpu = 1 vote
  - Zero-confirmation transaction

    Bob gives Alice product before transaction has been verified

    6 blocks; double spend probability goes down exponentially

    Never a 100% guarantee

- Solves the problem of not trusting a central authority - Problem 4
- 51% attacker
  1. CANNOT change block reward
  2. CANNOT spend other people's Bitcoin
  3. CANNOT change number of coins generated per block
  4. Suppress transactions
  5. Destroy confidence in Bitcoin

Changing the rules

- Two types of changes - soft forks; hard forks

  Soft forks are forward compatible; new rules are subset of old rules. Only applied if over 51% agree.

  Hard forks are backward compatible; old rules are subset of new rules. Everyone needs to upgrade to new.

Source: `http://people.dsv.su.se/~matei/courses/IK2001_SJE/Chaum90.pdf`
Source: `http://blog.koehntopp.de/uploads/Chaum.BlindSigForPayment.1982.PDF`