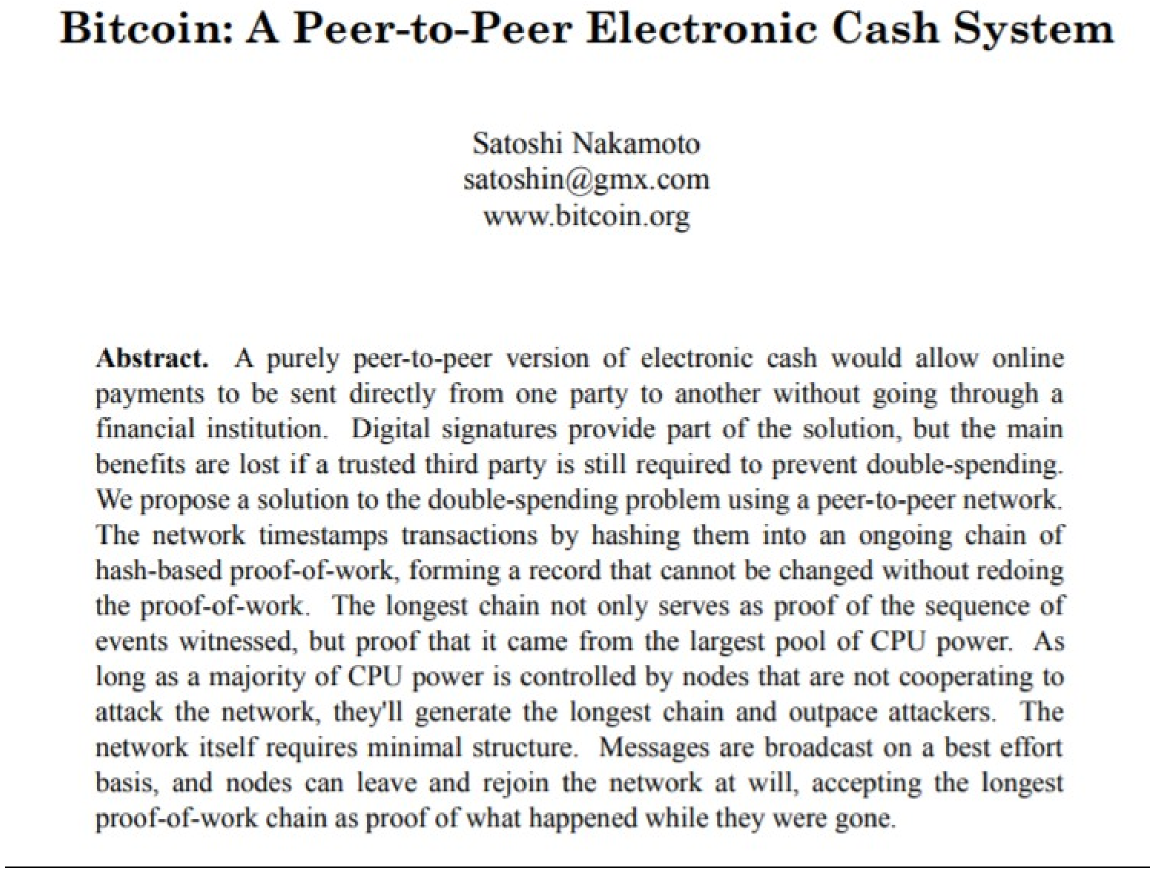
BLOCKCHAIN

Fundamentals of Blockchain Technology

Eoin Connolly (2021)

The White Paper

- Paper published October 2008
- Version 0.1 of the code released 9 January 2009
- https://bitcoin.org/bitcoin.pdf
- Initial posting



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 the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

What is a Blockchain?

- A chain of blocks where each block is linked to the previous
  - (like a linked list with hashes)
- Bitcoin is the first implementation
- There were other digital currencies before Bitcoin
  - Digicash
  - e-gold
  - Paypal
- What is different about Bitcoin?

Bitcoin

- Peer to peer network
- Decentralised
- Shared Ledger
- Immutable (?)

How does it work?

- Cryptographic building blocks
  - Hashes
  - Public and Private Key Pairs (asymmetric cryptography)
  - Digital Signatures
  - Merkle Trees

Hash

- Technopedia “A cryptographic hash function is a type of security mechanism that produces a hash value, message digest or checksum value for a specific data object.”
- Hashes have no input other than the algorithm and the base data
- Regardless of length of base data, hashes are always same length. e.g. SHA-256 always produces hashes that are 256 bits long
- Example hash calculator:
  - http://www.xorbin.com/tools/sha256-hash-calculator

Hash

- Collision resistance is the key feature of a hashing algorithm
- SHA-256 is 256 bits, or 10^76 combinations
- Number of atoms in the observable universe 4 x 10^79
- Birthday paradox: brute force of 256bit hash requires 2^128 attempts (3 x 10^38)

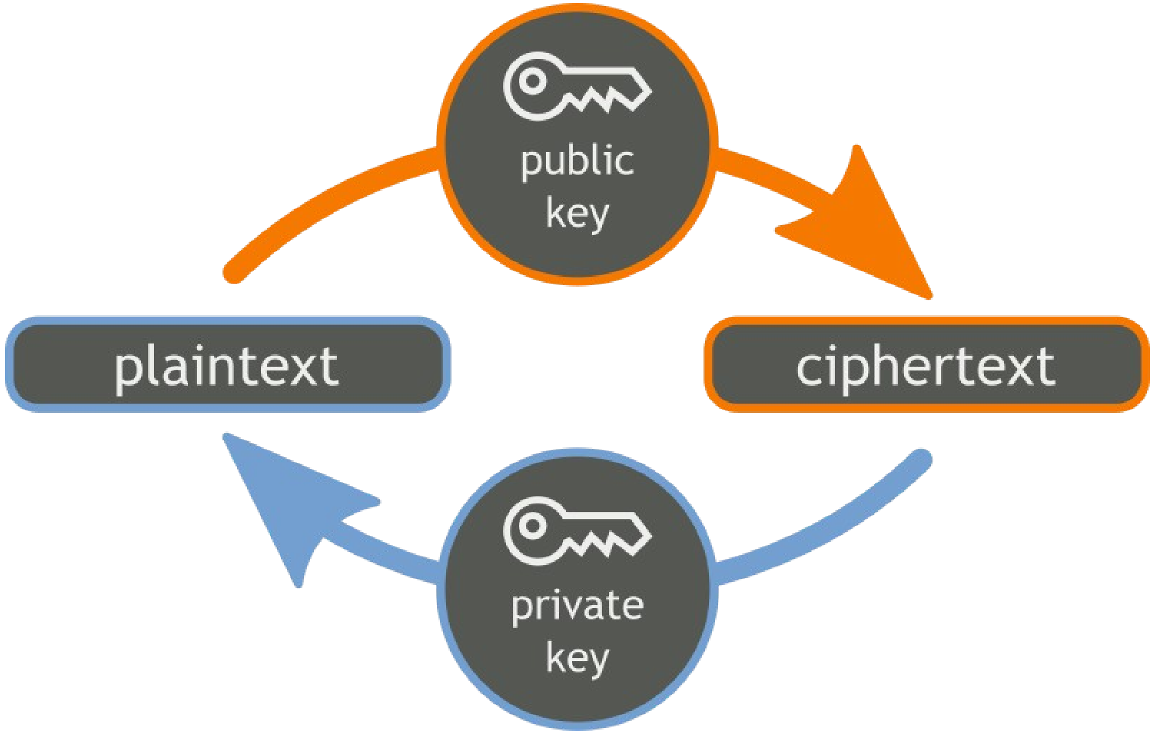
- What can we do with hashes?

Asymmetric Cryptography

- Public Key cryptography invented by Ron Rivest, Adi Shamir, Leonard Adleman in 1976 (RSA)
- Also invented in secret by Clifford Cocks in GCHQ 3 years earlier :)
- Unlike symmetric cryptography different key is used to encrypt and decrypt
- Usually called the Private Key and the Public Key
- The RSA public key is used by others to encrypt data they wish to share with you
- You use the private key to decrypt this data
- (both keys can be used interchangeably)

Asymmetric Cryptography

- Also used in:
- PGP (combination of symmetric and asymmetric cryptography)
- HTTPS (key exchange)
- Note ECC is slightly different (we'll cover that in more detail later)



Digital Signature

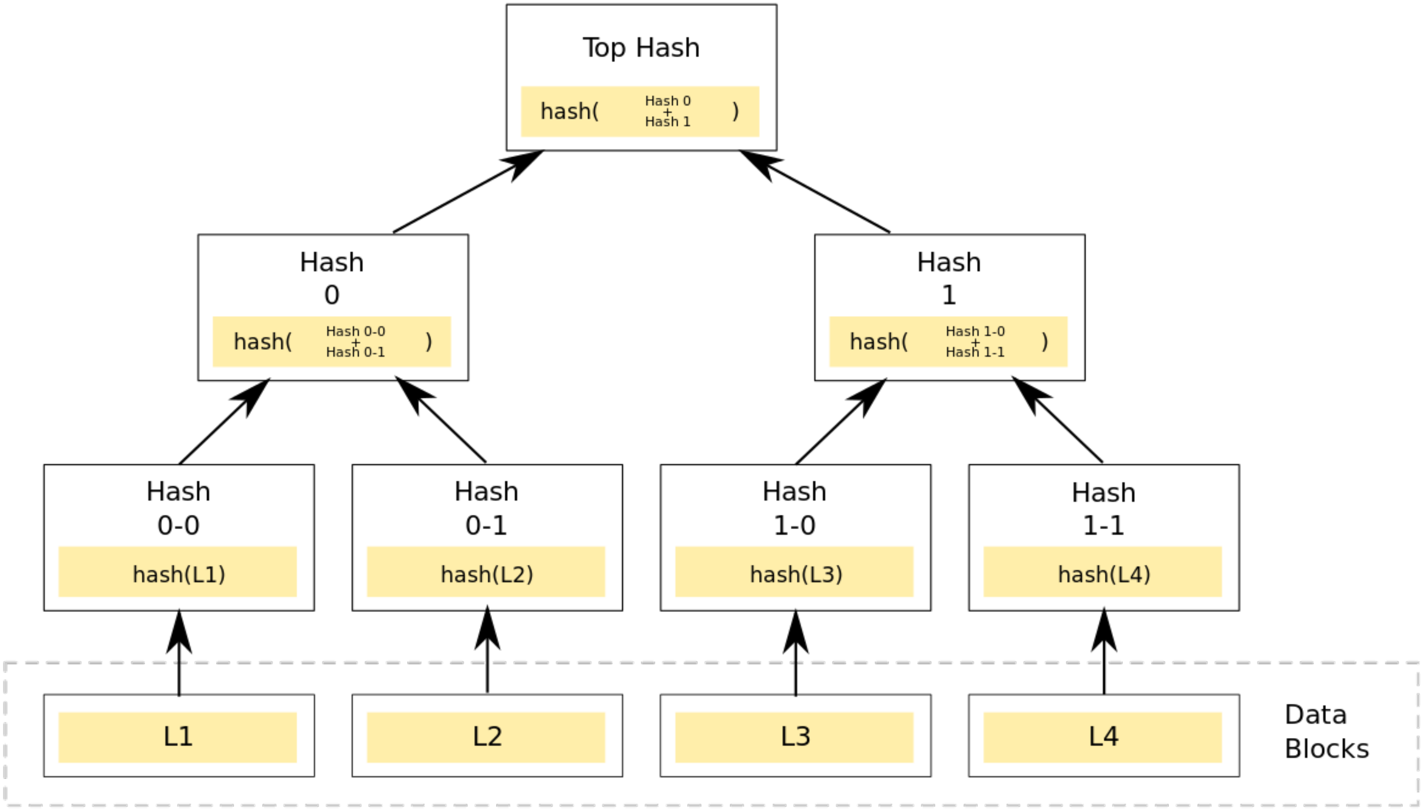
- A **digital signature** is a mathematical scheme for demonstrating the authenticity of digital messages or documents
- It is created using a combination of hashes and asymmetric cryptography

- RSA
- You first create a **hash** of the file/message/document/statement
- Then you encrypt the hash with your **private key**
- To validate, anyone can decrypt the encrypted hash with your public key

- Q: What purposes can this have?

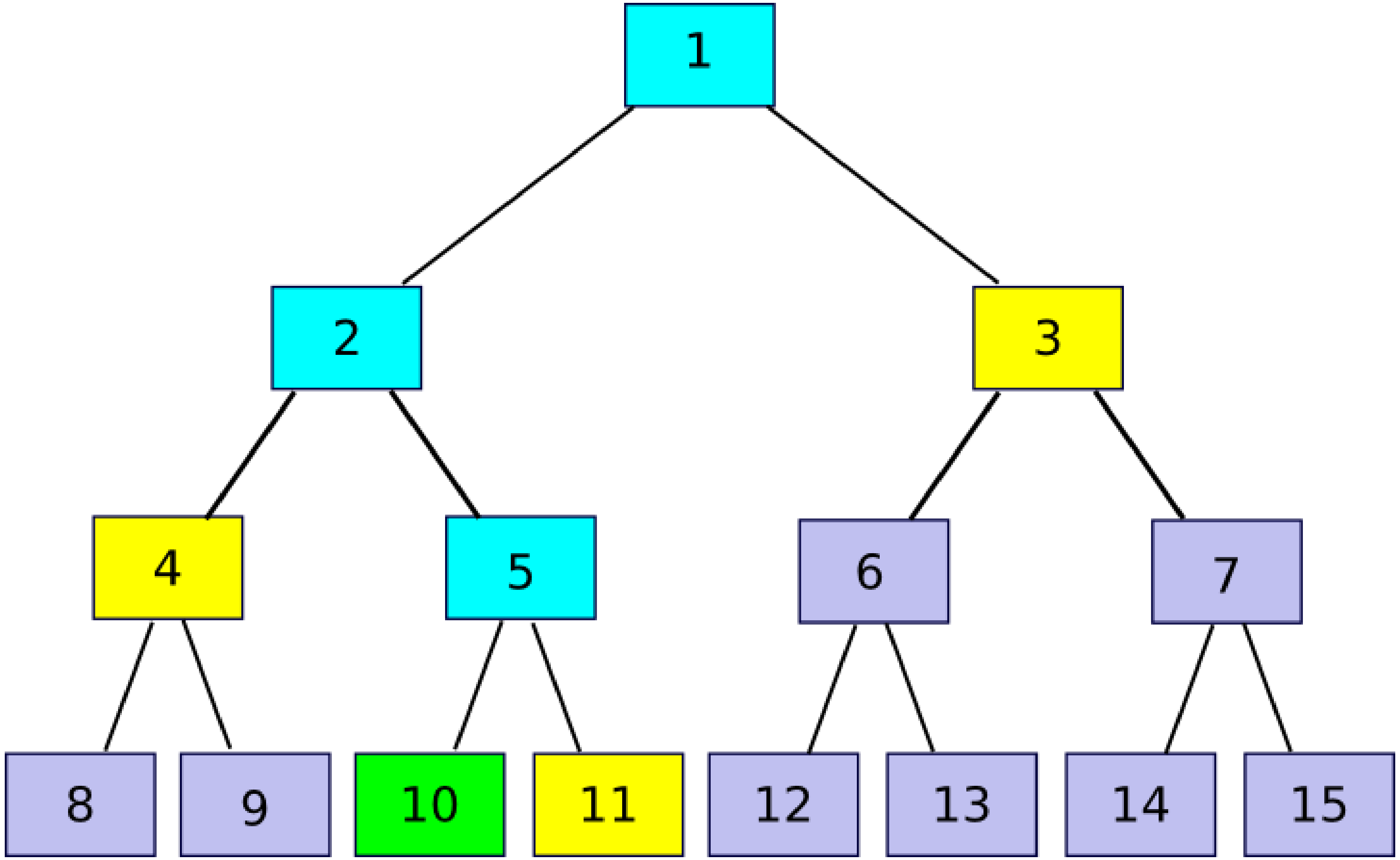
Merkle Trees

- A **merkle tree** is a tree of hashes
- Named after Ralph Merkle who patented them in 1979
- The top hash is also called the **Merkle Root**
- Merkle Trees can be used to provide **Merkle Proofs** – proof that a piece of data is part of the data set represented by the Merkle Root



Merkle Proof

- How do we prove (10) is part of the **merkle root** (1)?

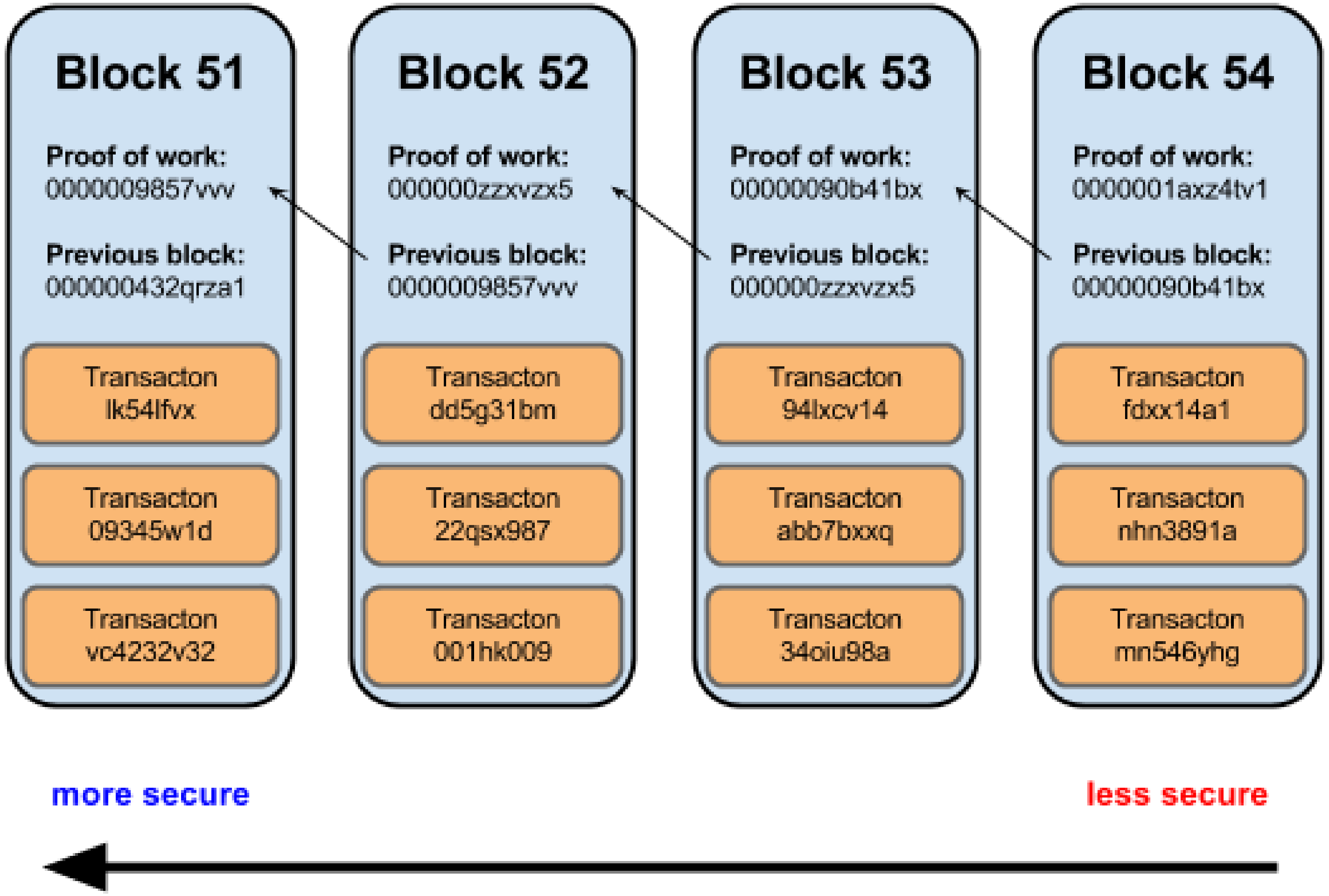


Back to Blockchains

- A blockchain is a chain of blocks
- Inside each block is:
  - Header data
    - Version Number
    - Hash of Previous Block
    - Transaction Merkle Root
    - Timestamp
    - Difficulty Target
    - Nonce
  - Transactions

Blocks

- How is this immutable?



Block 51

Proof of work:
0000009857vvv

Previous block:
000000432qrza1

Transacton
lk54lfvx

Transacton
09345w1d

Transacton
vc4232v32

Block 52

Proof of work:
000000zzxvzx5

Previous block:
0000009857vvv

Transacton
dd5g31bm

Transacton
22qsx987

Transacton
001hk009

Block 53

Proof of work:
00000090b41bx

Previous block:
000000zzxvzx5

Transacton
94lxcv14

Transacton
abb7bxxq

Transacton
34oiu98a

Block 54

Proof of work:
0000001axz4tv1

Previous block:
00000090b41bx

Transacton
fdxx14a1

Transacton
nhn3891a

Transacton
mn546yhg

Blocks are “more secure” as you go further back in the chain

http://www.ybrikman.com/writing/2014/04/24/bitcoin-by-analogy/

Proof of Work

- A **proof-of-work** function is an economic measure to deter denial of service attacks by requiring work to be completed before the service can be accessed
- Bitcoin uses the **Hashcash** proof of work function (Adam Back, 1997)
- Hashcash originally proposed to counter email spam
- POW to achieve decentralised consensus is the key novelty of the Bitcoin paper

Bitcoin Proof of Work

- Each block has a difficulty (adjusted every 2016 blocks: ~2 weeks)
- Difficulty is intended to keep the block interval at ~10 minutes
- A valid block has a hash value which is *lower* than the target difficulty
- Sample of current Bitcoin block hash
  - 000000000000000000cd9ce061de1770365b0841980cde2d8f79502385beb89c

- Q: how many hashes would you have to generate to have a chance of creating the hash above?

Proof of Work

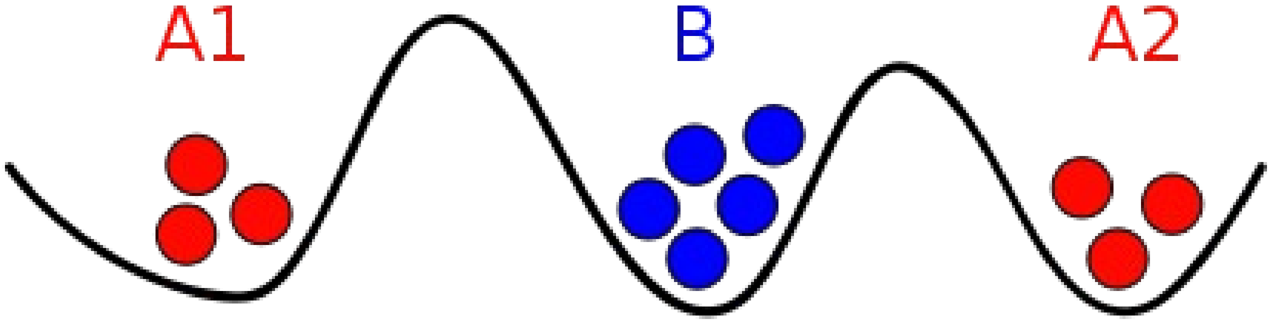
- How does proof of work provide immutability?
- How does the block hash prove work has been completed?

- Current Bitcoin network hashrate: 138,000,000,000,000,000,000H/s

- How much would it cost to obtain 51% of Bitcoin's hash rate?
  - Assume 10c KWh cost
  - Antminer S9
    - 14 TH/s
    - 1372W power consumption
    - €1589 (ex VAT)

Byzantine Fault Tolerance

- The Byzantine General’s Problem - Lamport, Shostak, Pease (1982)
  - generalised version of 2 generals problem
  - All generals must coordinate attack
  - Attacks are coordinated by messengers
  - Messengers must travel through enemy territory
  - Messengers can forge messages
  - Generals can send multiple messages
  - An uncoordinated attack will not succeed



PBFT & Bitcoin

- No solution of PBFT possible if >=1/3 of generals act maliciously
- Bitcoin solves the Byzantine Fault Tolerance problem by using Proof of Work, which increases the cost/time to generate a message compared to propagation time, which ensures a consensus is reached around longest valid chain

Other Consensus Mechanisms

- Proof of Stake
- PBFT
- Ordering Services

Recommended Reading

- Satoshi White Paper
  - https://bitcoin.org/bitcoin.pdf

- Wei Dai’s b-money
  - http://www.weidai.com/bmoney.txt

Ethereum Yellow Paper

  - https://ethereum.github.io/yellowpaper/paper.pdf

- Digital Gold (Nathaniel Popper)
  Mastering Ethereum (Andreas Antonpolous)