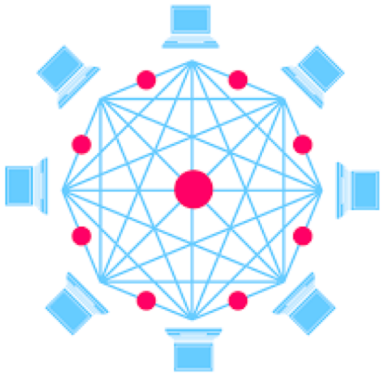
Foundation of Blockchain

What is Blockchain?

- **Distributed Database**
  - **Consistent:** It cannot conflict with some other data that's already in the database
  - **Immutable:** It's append-only
  - **Canonical:** Everyone agrees on what the state of the things in the database are



Source: https://medium.com/@jimmysong/why-blockchain-is-hard-60416ea4c5c

Four Benefits

1. Distributed Data Management
2. Logic Reliability
3. Digital Scarcity
4. Incentive Mechanism

Hash

A function that takes an input string of any length and gives out an output of a fixed length.

Input	Output
Hi	8f434346648f6b96df89dda901c5176b10a6d83961dd3c1ac88b59b2dc327aa4
あいうえお	fdb481ea956fdb654afcc327cff9b626966b2abdabc3f3e6dbcb1667a888ed9a

Hash

1. **Deterministic**
   You will always get the same result if the input is the same.
2. **Quick Computation**
   The hash function should be able to return the output quickly.
3. **Pre-Image Resistance**
   You cannot determine input from output.

8f434346648f6b96df89dda901c5176b
10a6d83961dd3c1ac88b59b2dc327aa4

Hash

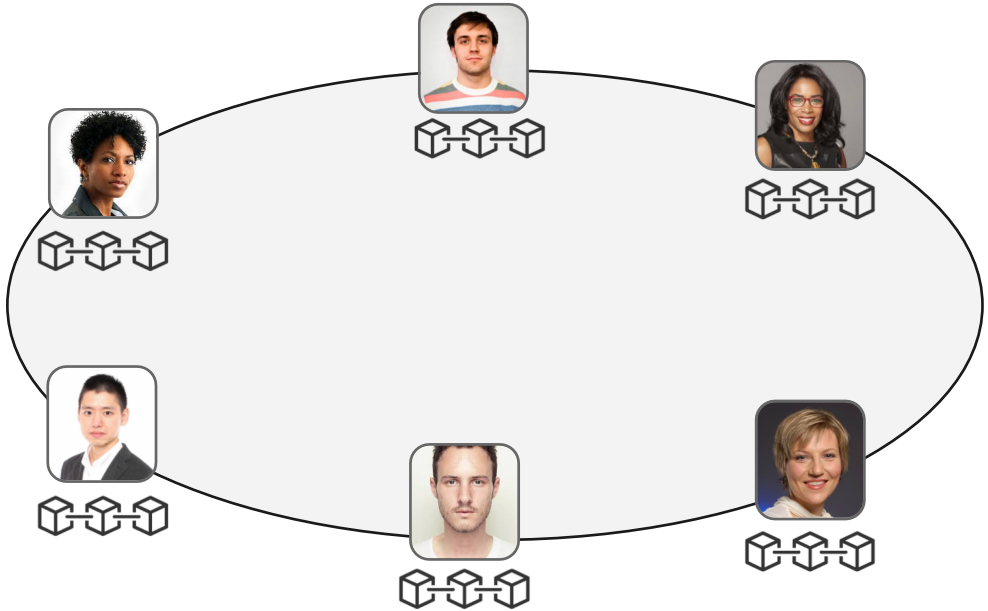
4. Small Changes in the input changes the hash

5. Collusion Resistant

The output of one input will not be the same of another output.

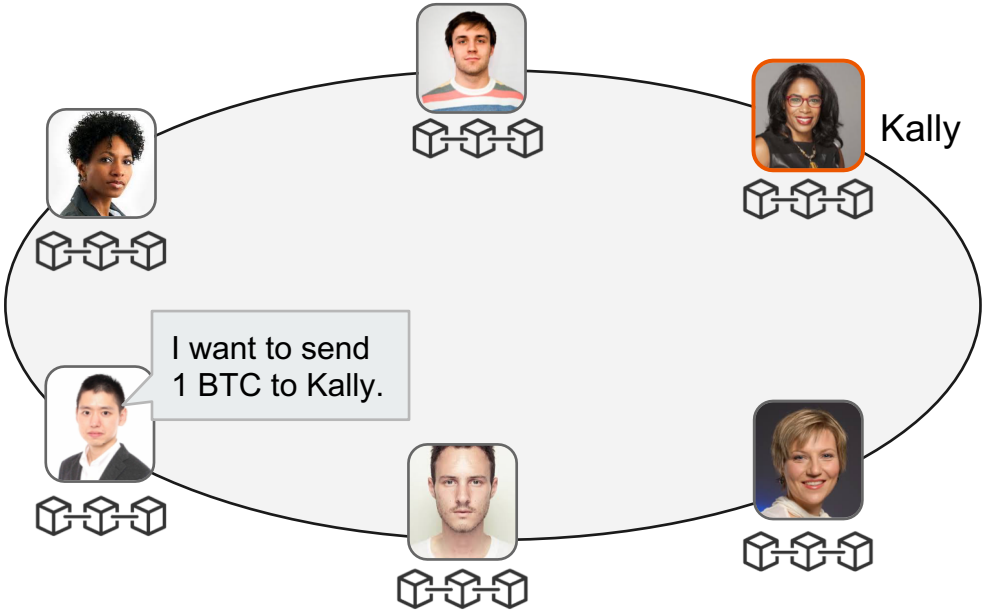
Input	Output
Hi	8f434346648f6b96df89dda901c5176b10a6d83961dd3c1ac88b59b2dc327aa4
Hii	a1a3b09875f9e9acade5623e1cca680009a6c9e0452489931cfa5b0041f4d290

Transaction

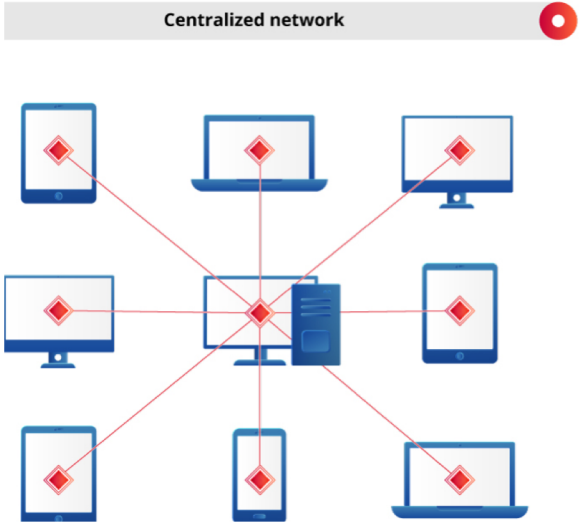


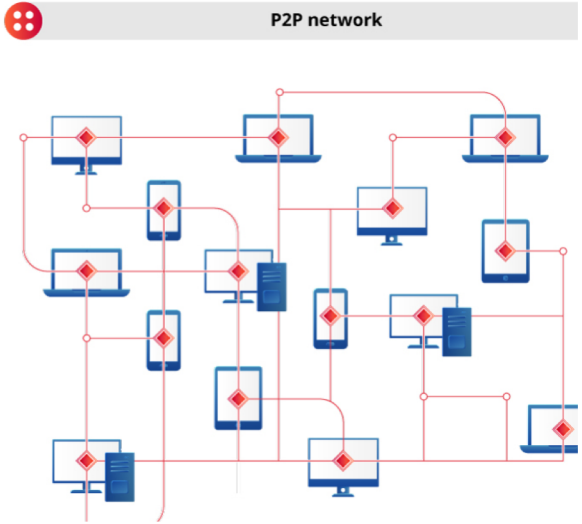
Everyone has the same data.

Transaction

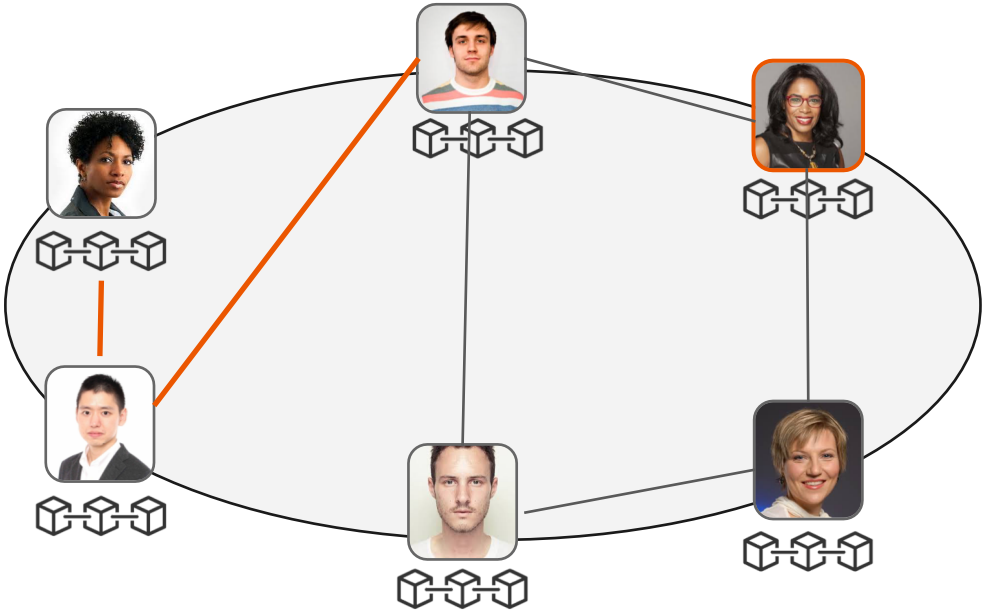


Transaction

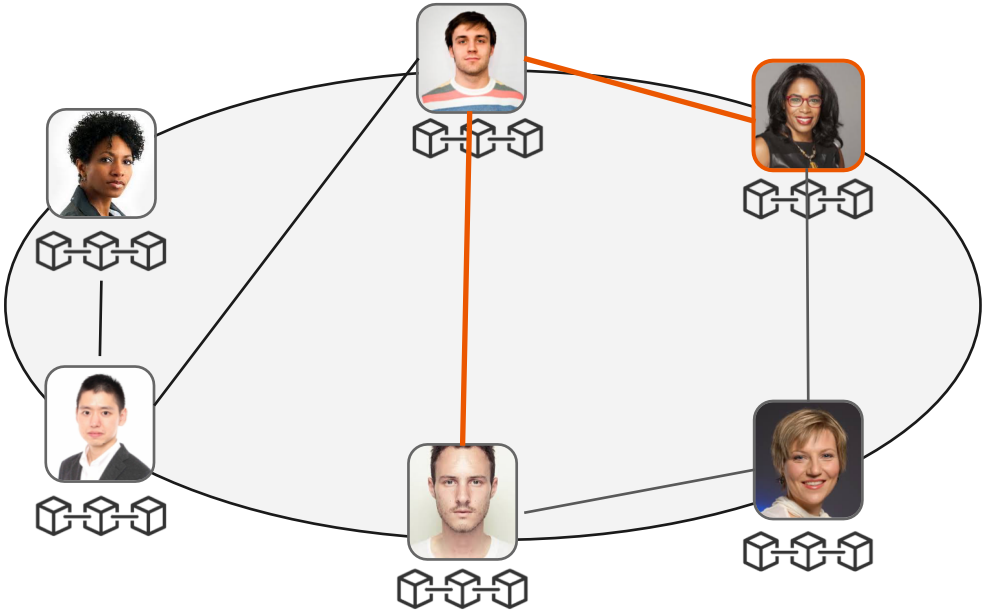




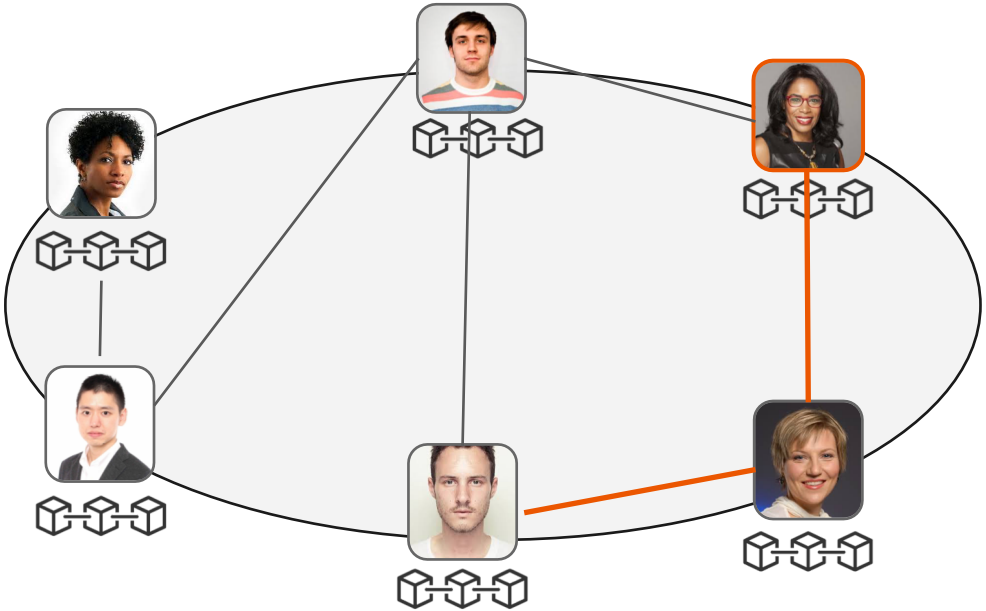
Transaction



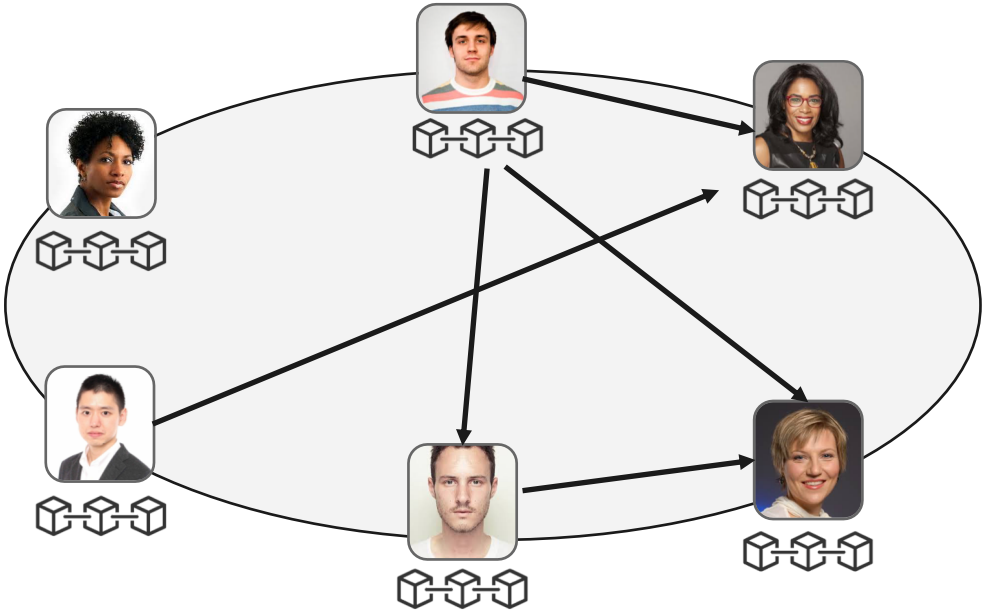
Transaction



Transaction



Transaction



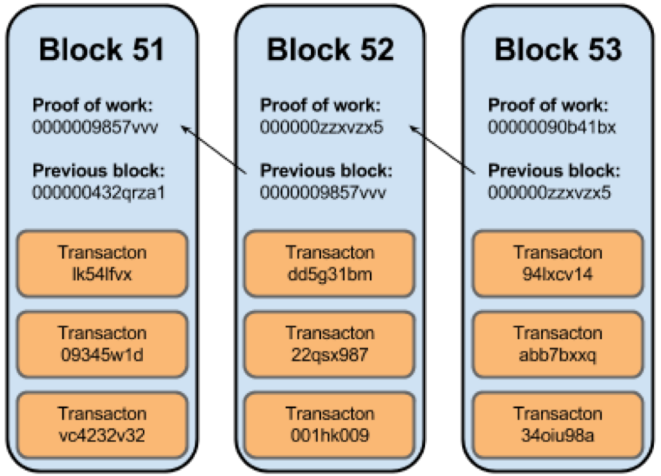
Transaction

Kojiro -> Kelly, Amount 1BTC
AAA -> BBB, Amount 3BTC
CCC -> DDD, Amount 3BTC
EEE -> FFF, Amount 0.0001BTC
GGG -> HHH, Amount 100BTC

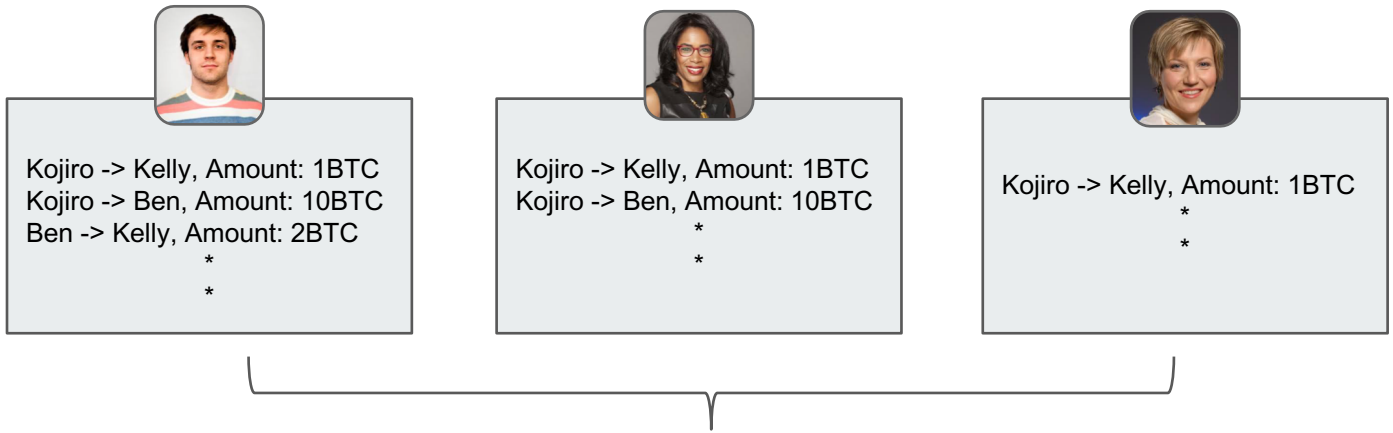
These transactions will be put into a block.

Block

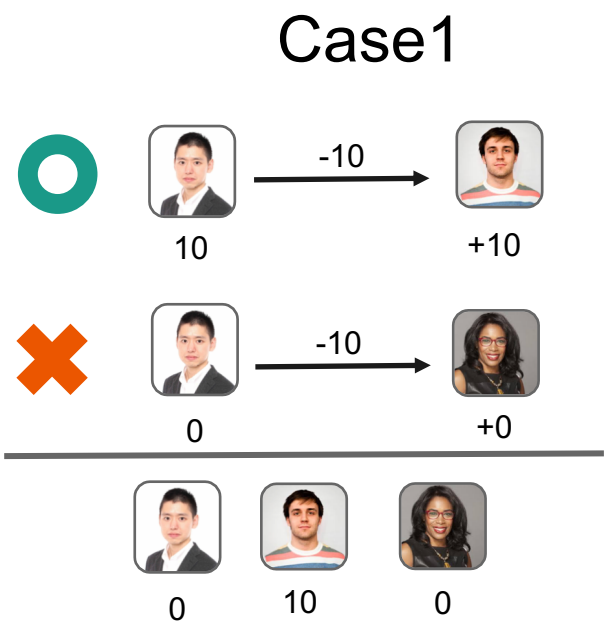
- All data is saved into “block”
- “Block” is something like space for data
- New block gets generated periodically
- Each block is linked to the previous block

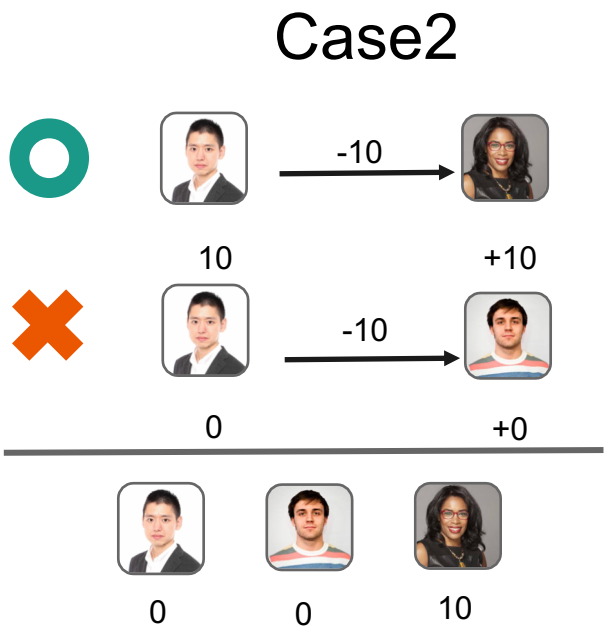


Consensus Problems

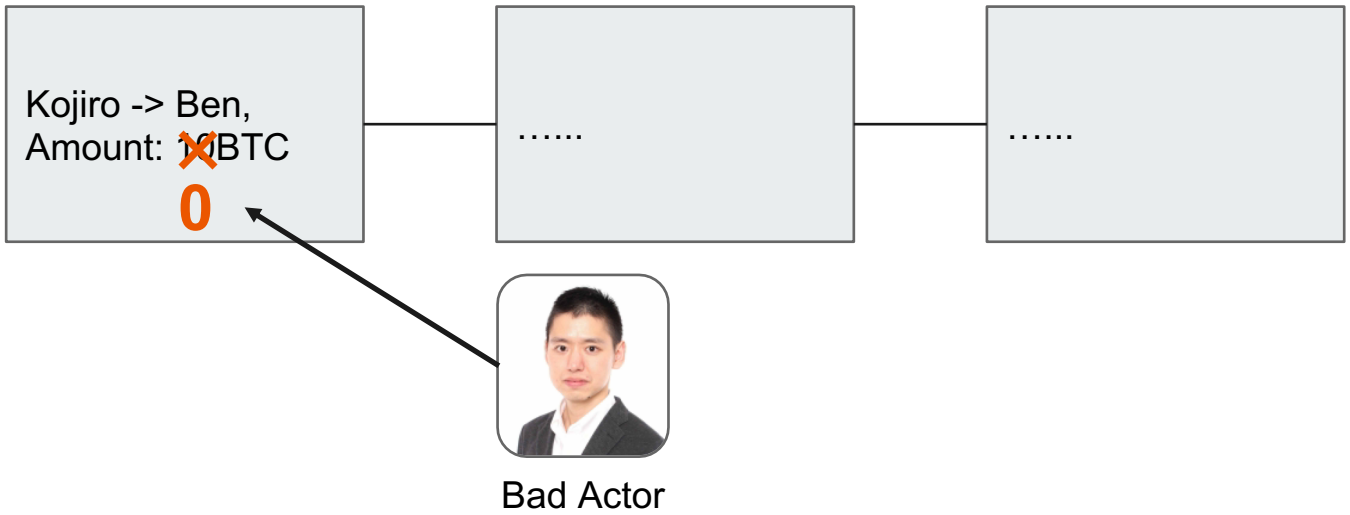


Consensus Problems





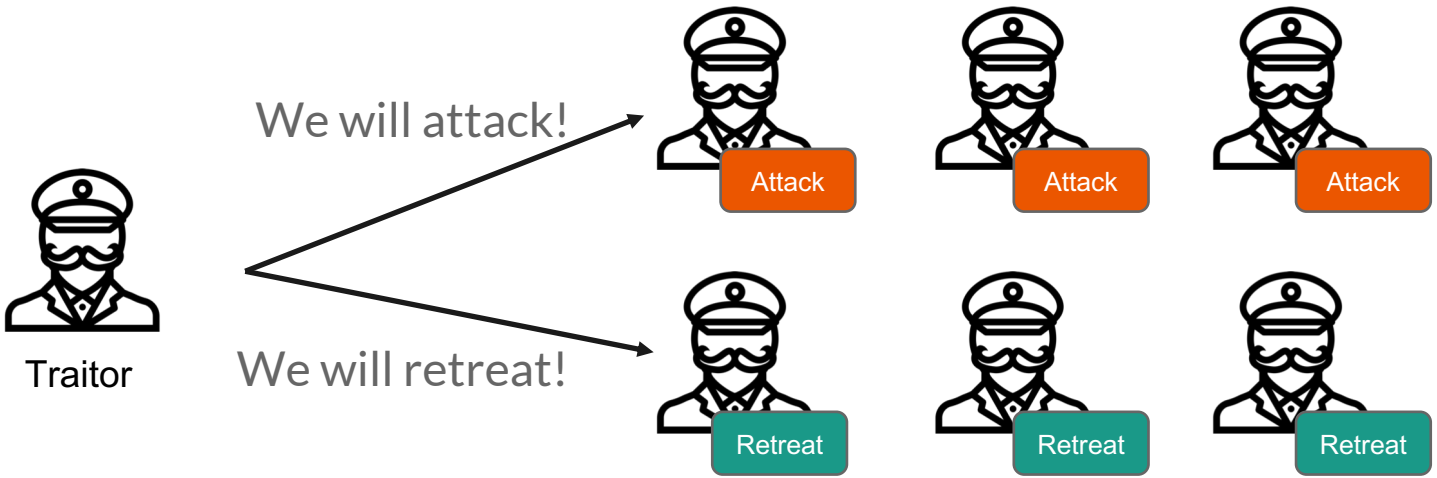
Consensus Problems



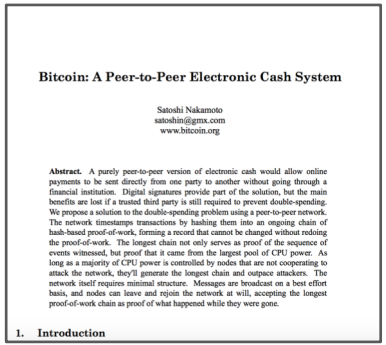
Consensus Problems

Byzantine Generals Problems

Consensus Problems



History of Bitcoin



Bitcoin: A Peer-to-Peer Electronic Cash System

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

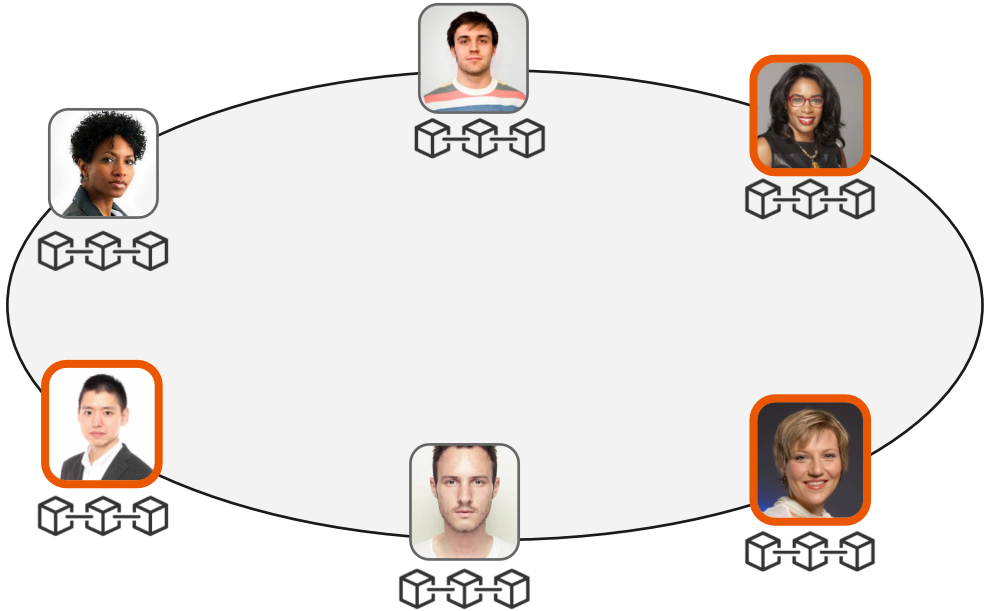
Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required 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.

1. Introduction

- The 9 pages paper was written by unknown person called “Satoshi Nakamoto”
- “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.”

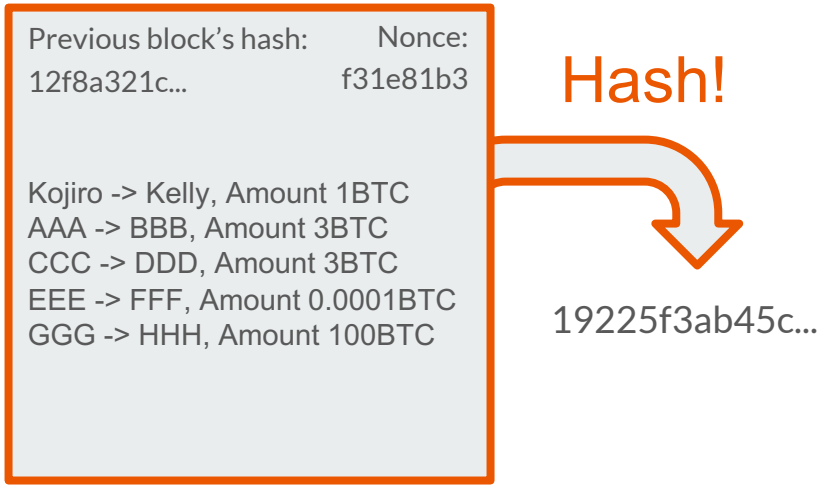
Source: https://bitcoin.org/bitcoin.pdf

Proof of Work



Miners compete for rewards .

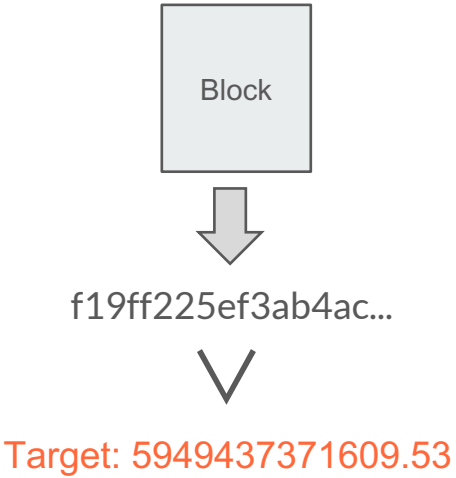
Proof of Work



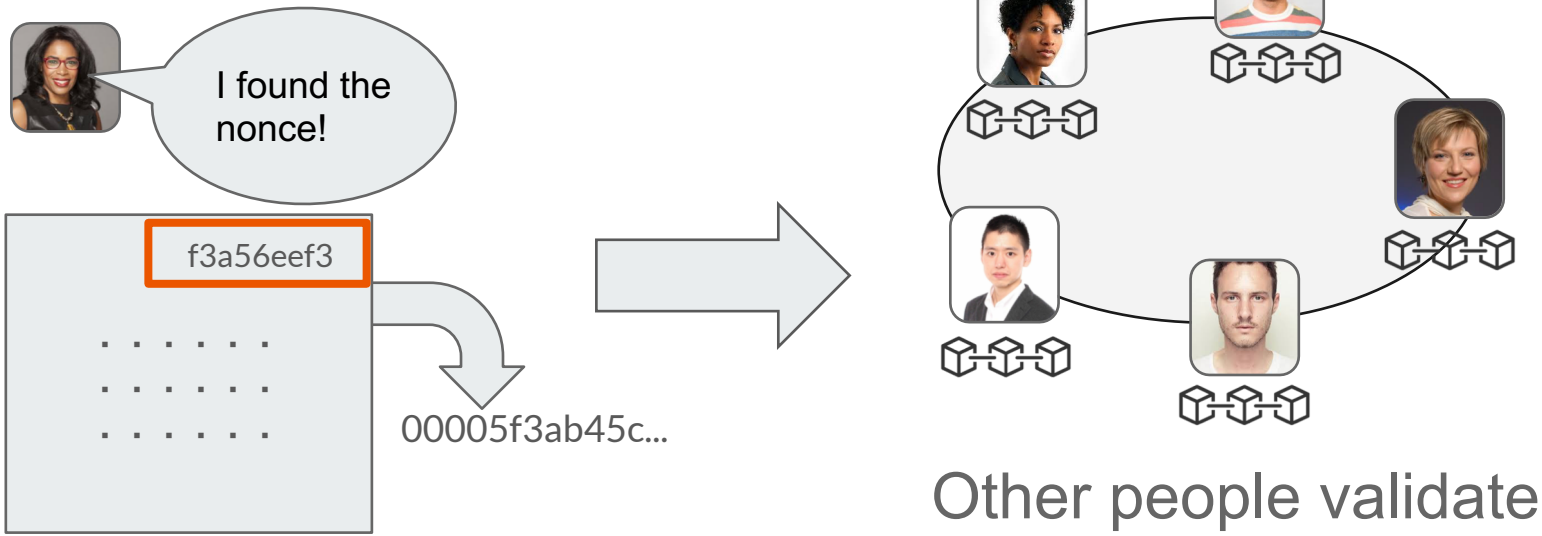
- Miner adds a random hexadecimal string called "Nonce."
- Miner hashes all the texts inside the block into one string.

Proof of Work

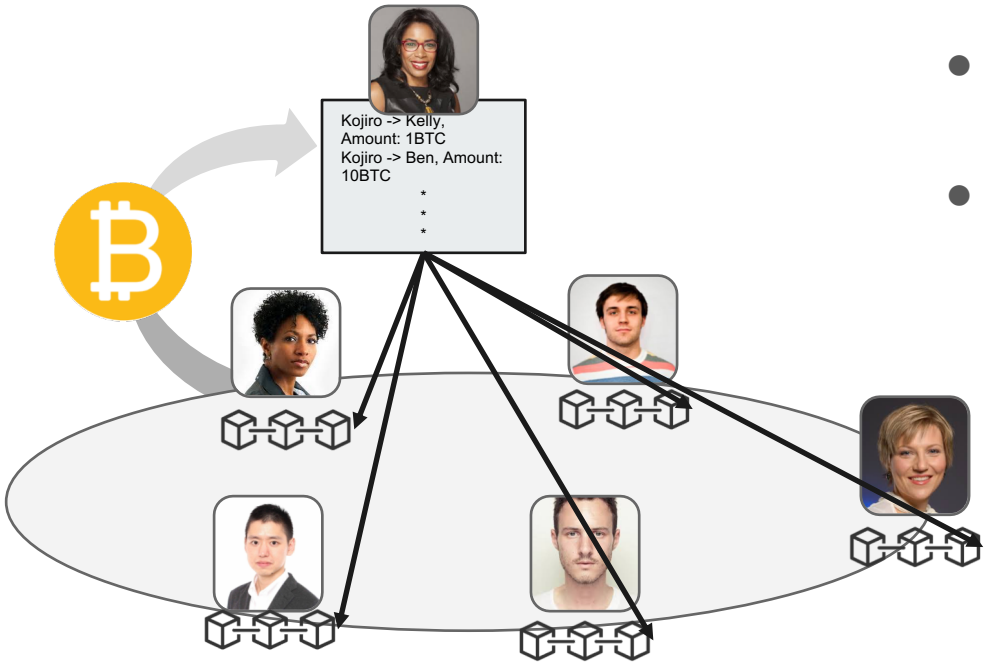
- There is a target for the output.
- Miner needs to find a nonce that will make the output smaller than the target.
- All the things miner can do is just keep changing the nonce.
- Miners will compete each other to find the nonce.



Proof of Work



Proof of Work



- The block will be added to other people's blockchain.
- The winner will receive BTC as reward.

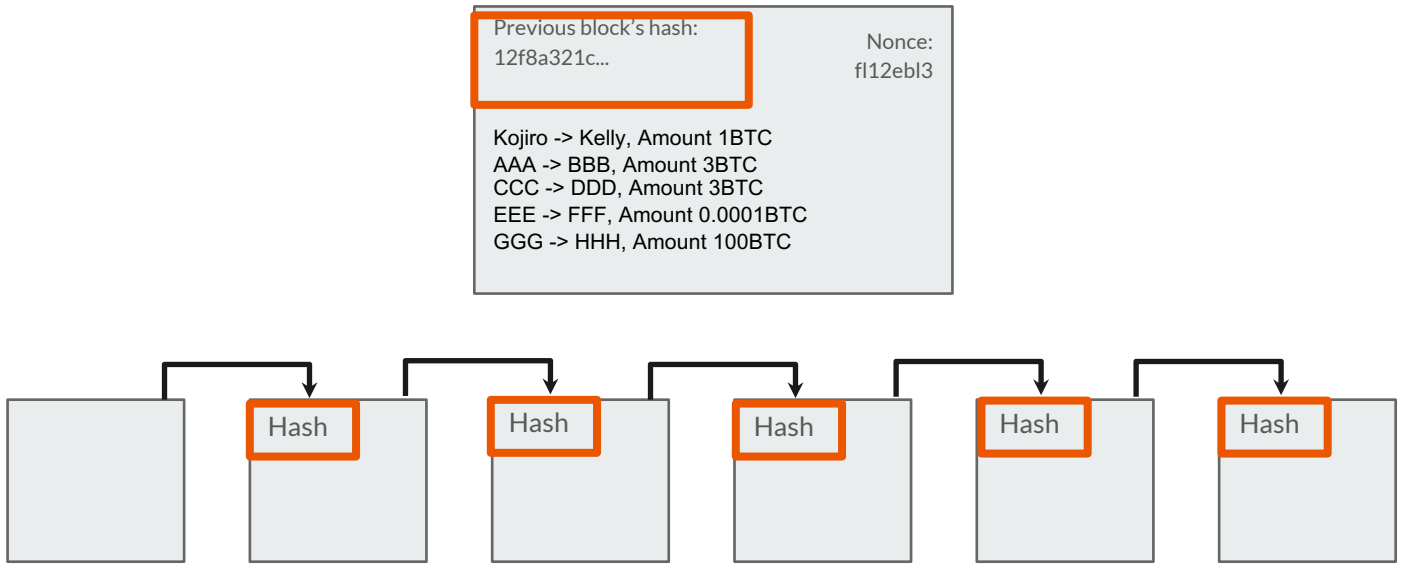
Proof of Work

https://www.youtube.com/watch?v=aYcvkTJKBNw

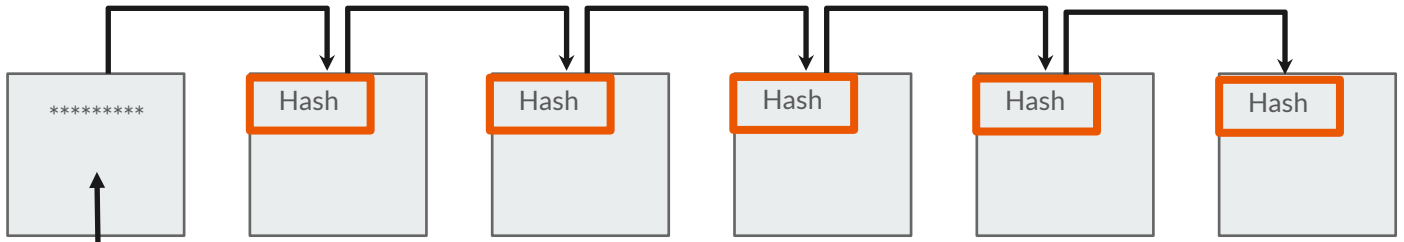
Proof of Work

- Miner gets two types of rewards
  - New Coins
  - Transaction fees

Proof of Work



Proof of Work



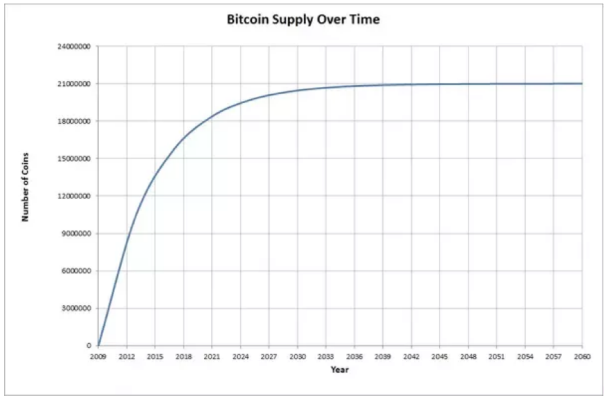
Bad Actor

- To change the data in previous blocks, all the hashes needs to be re-calculated.
- This is simply impossible.

Energy Usage

- Consume a lot of energy!!
- Bitcoin alone currently consumes 0.14% of global energy consumption.
- The consumption is almost the same as Switzerland’s entire country consumption.

Bitcoin Economy



- Total supply is 21 million coin.
- Not all of them have been issued yet.
- New coin is issued every time miner creates a block.
- Issuance rate decreases in about every 4 years
  - 25 BHC per block in Nov 2012
  - 12.5 BHC per block in July 2016
  - 6.25 BHC per block in 2020
  - 0 BHC in 2140.

Key, Address, Digital Signature

Private Key

- A randomly generated number.
- The root of controlling token.
- Create signatures for proving the ownership of fund.
- **Only you should know Private Key!**

Public Key

- Generated from private key.
- Used for proving the ownership of fund.
- Can be seen by anyone.

Key, Address, Digital Signature

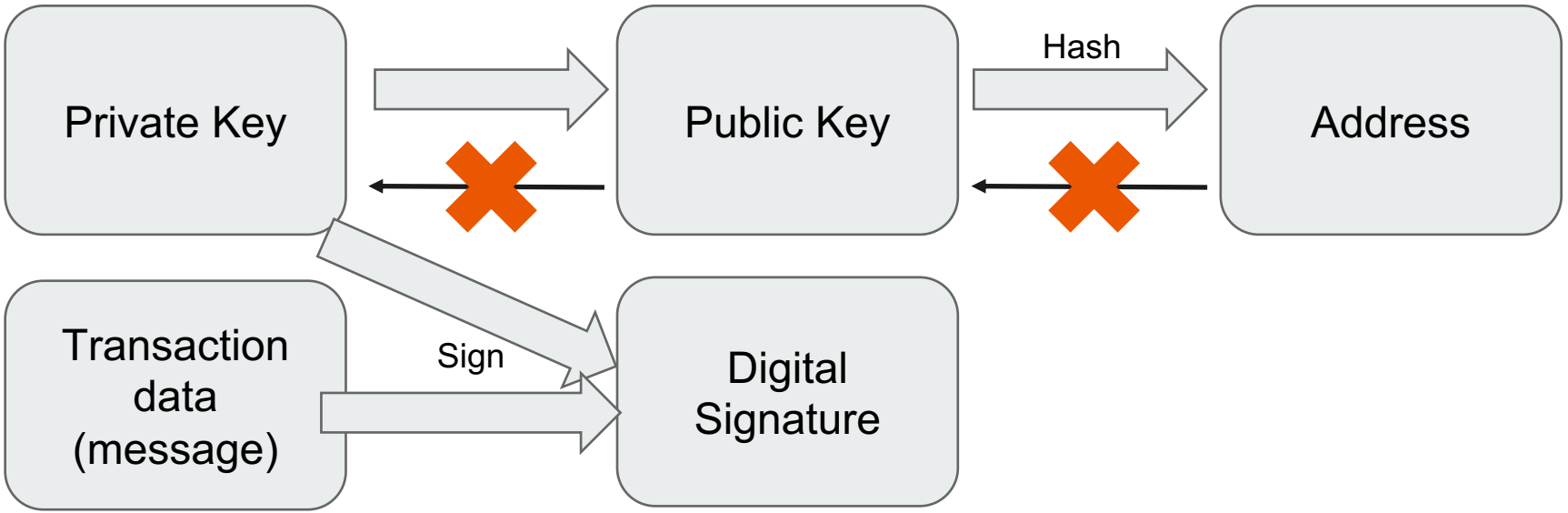
Address

- Hash of Public Key
- Used as a destination of transaction.
- Can be seen by anyone.

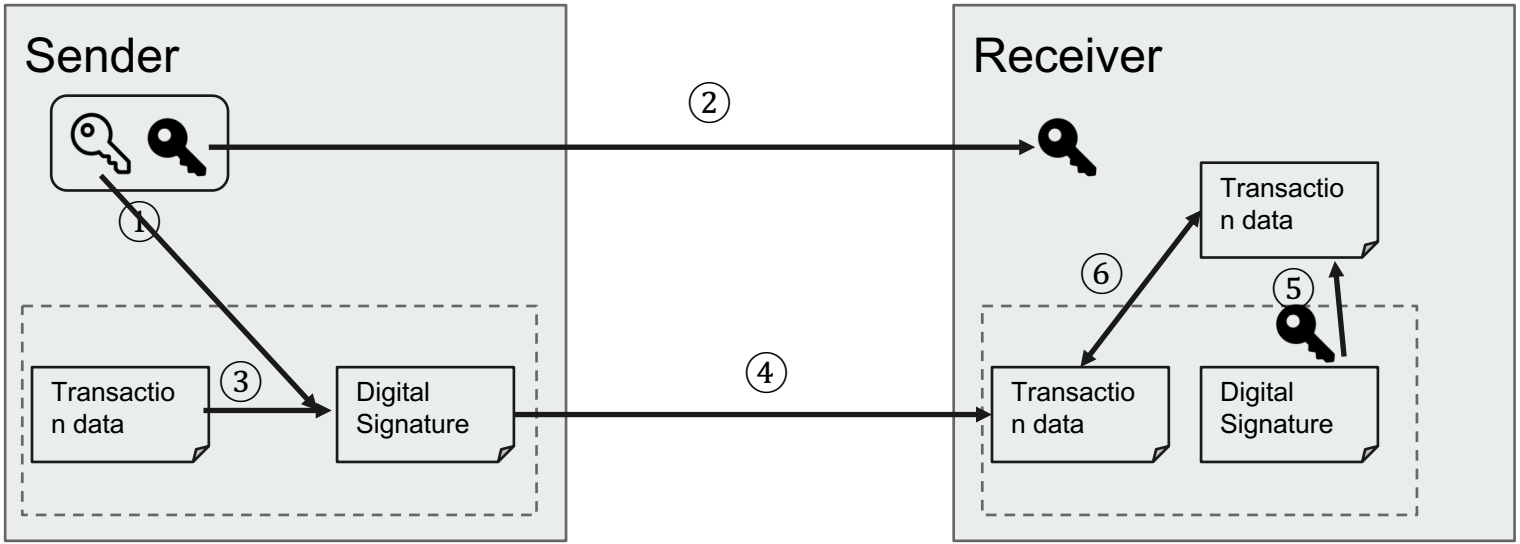
Digital Signature

- Generated by transaction data and private key.
- Only you can sign, but anyone can verify that it’s valid with public key.
- Signature is tied to a particular transaction.

Key, Address, Digital Signature



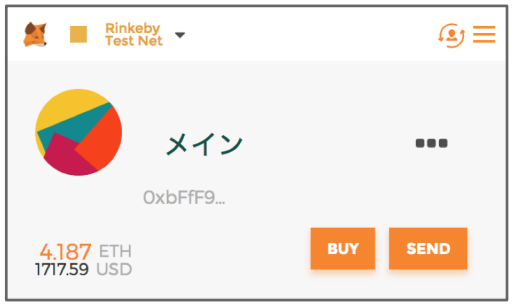
Key, Address, Digital Signature



Wallets

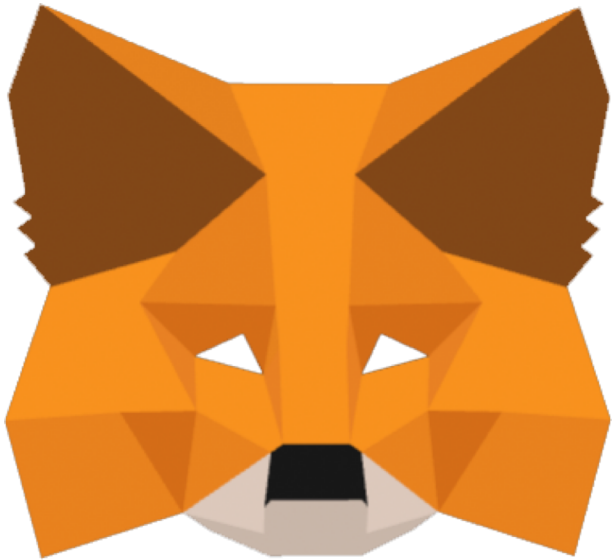
An application that controls access to user’s tokens, managing keys, and addresses.

Cold Storage



Hot Storage

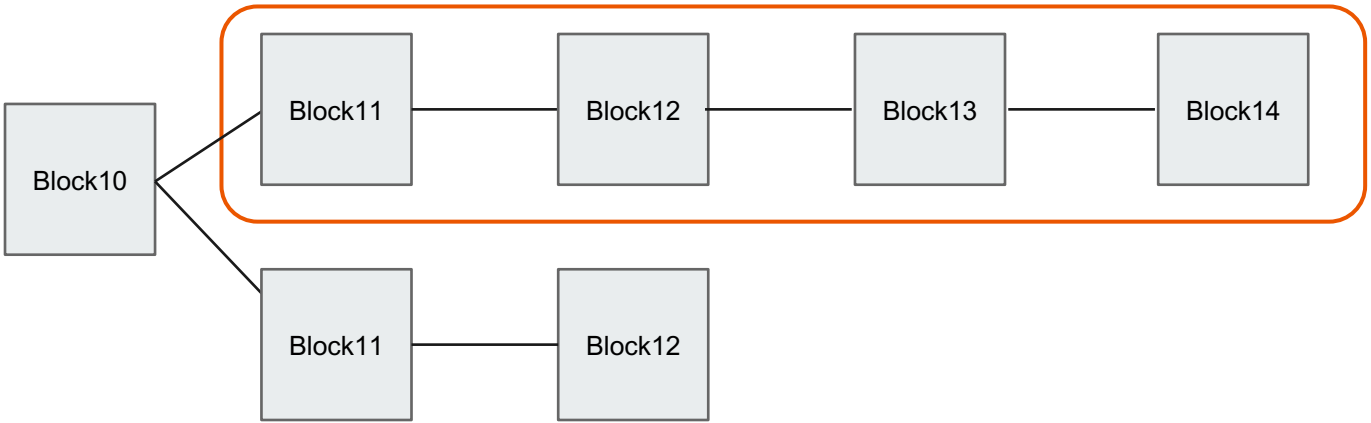
Let’s Try



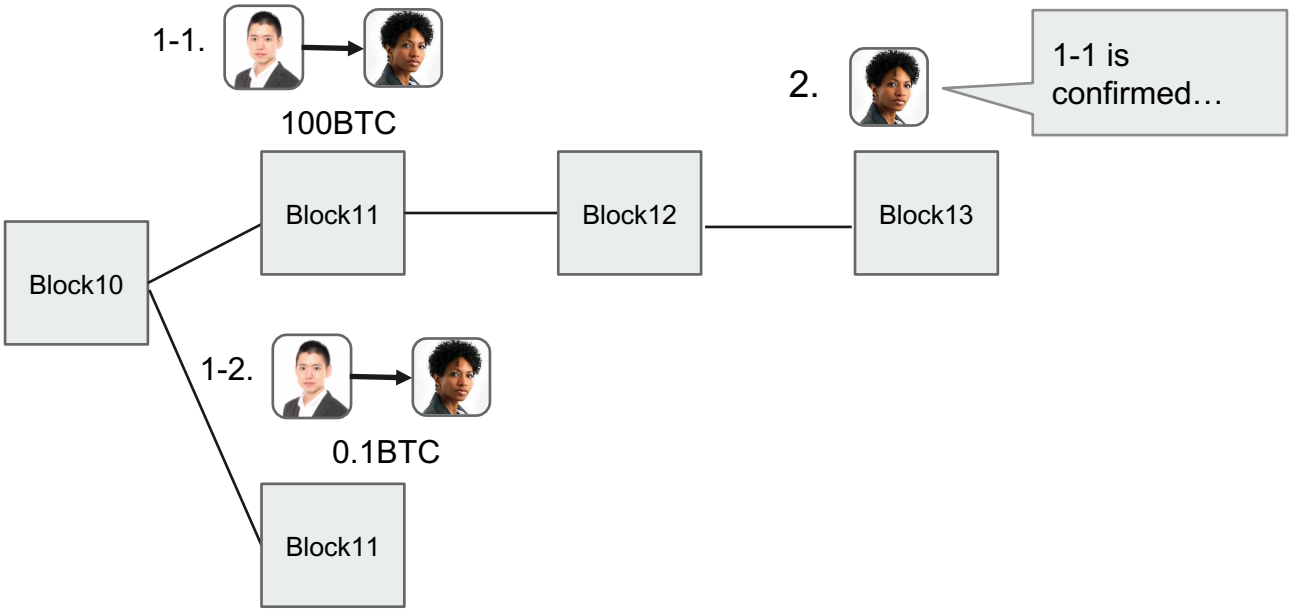
Remaining Problems

- 51% Attack
- Transaction Speed
- Environmental Damage

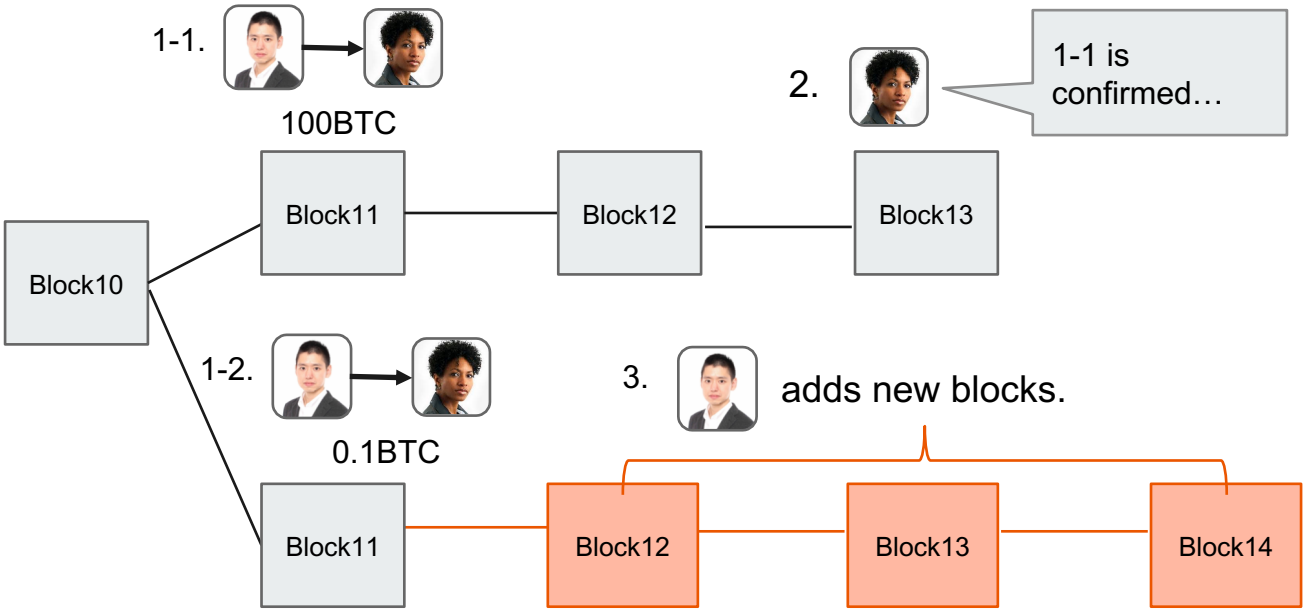
51% Attack



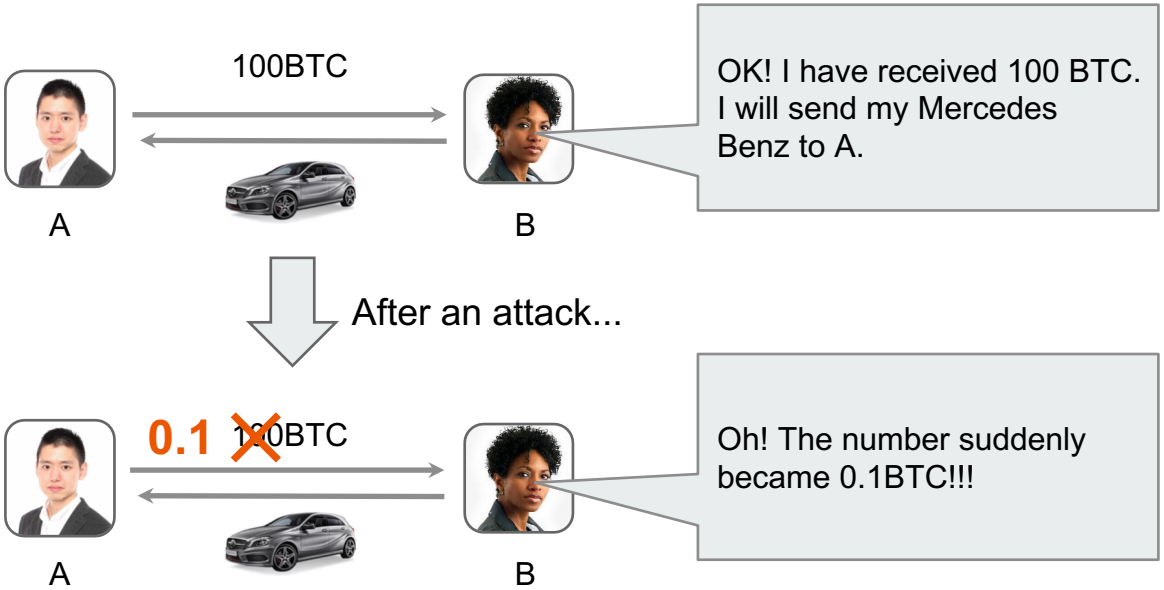
51% Attack


1-1.
100BTC
2.
1-1 is confirmed…
Block10
Block11
Block12
Block13
1-2.
0.1BTC
Block11

51% Attack


1-1.
100BTC
2.
1-1 is confirmed…
Block10
Block11
Block12
Block13
1-2.
0.1BTC
3.
adds new blocks.
Block11
Block12
Block13
Block14

51% Attack


100BTC
A
B
OK! I have received 100 BTC.
I will send my Mercedes
Benz to A.
After an attack...
0.1
100BTC
A
B
Oh! The number suddenly
became 0.1BTC!!!

PoS

100 coins

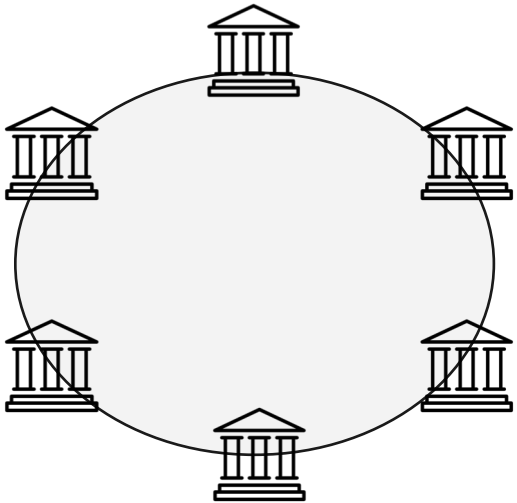
10,000 coins

50 coins

Bigger chance of winning

- Environmental-Friendly
- Faster Transaction Speed
- Reduce the chance of 51% attack

Consortium Blockchain



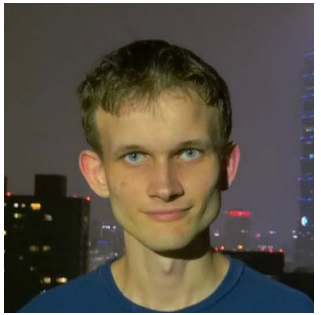
- Only the entities who are granted for permission can join.
- Voting-based validation mechanism.

Public vs Consortium Blockchain

Type	Public	Consortium
Participants	Anyone	Only people granted for permission
Consensus Mechanism	Proof of Work, Proof of Stake	Voting-based consensus algorithm
Transaction Speed	Slow	Fast
Pros	Distributed	Fast-transaction Speed
Cons	Slow	Partially Centralized
Use Cases	Currency, B2C	B2B

Ethereum

- The second largest blockchain led by a Canadian Programmer, Vitalik Buterin.
- The first version was released in 2015.
- The second largest blockchain after Bitcoin.



Ethereum

More programmer friendly than Bitcoin.

```
OP_DUP OP_HASH160 62e907b15cbf27d5425399ebf6f0fb50ebb88f18
OP_EQUALVERIFY OP_CHECKSIG
```

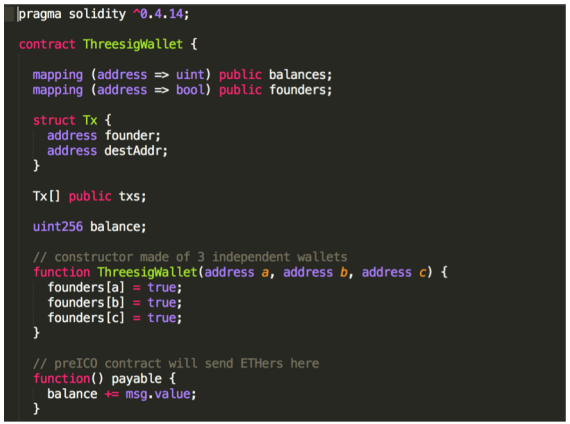
Bitcoin

```
if hasher.Verify(myBlock) {
        //fmt.Println("Share is valid")
        if hasher.Verify(myBlockRealDiff) {
                submitWork(paramsOrig)
                logInfo.Println("########################################
                logInfo.Println("############################Block fou
                logInfo.Println("########################################
        }
```

Ethereum

Smart Contract

- Can easily implement immutable logic called “Smart Contract.”
- The largest blockchain developer community. (30 times than the next blockchain community.)



```
pragma solidity ^0.4.14;

contract ThreesigWallet {

  mapping (address => uint) public balances;
  mapping (address => bool) public founders;

  struct Tx {
    address founder;
    address destAddr;
  }

  Tx[] public txs;

  uint256 balance;

  // constructor made of 3 independent wallets
  function ThreesigWallet(address a, address b, address c) {
    founders[a] = true;
    founders[b] = true;
    founders[c] = true;
  }

  // preICO contract will send ETHers here
  function() payable {
    balance += msg.value;
  }
```