Bitcoin: A Peer-to-Peer Electronic Cash System

Author: Satoshi Nakamoto
Presenter: Yiwen Yao
Date: Dec 1, 2021

Content

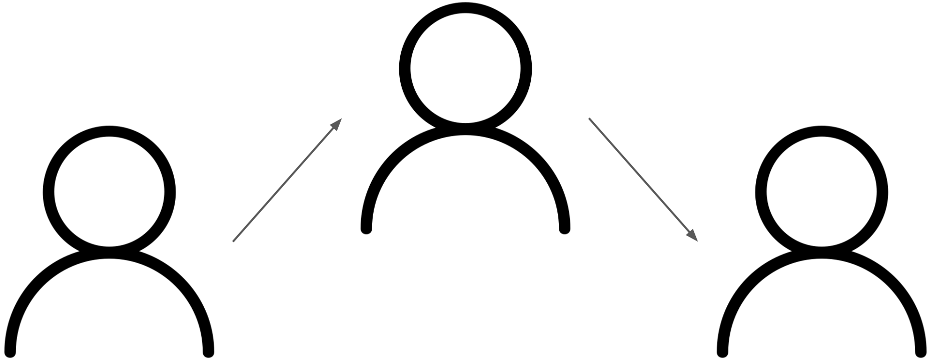
- Why Bitcoin?
    - Electronic payments background
    - Motivations of Bitcoin
- How it works?
    - Transaction and Times Server - payment records
    - Proof-of-work - to agree on a payment history
    - Network
- What if attacked?
    - Double spending problem
    - Incentive
- How to improve it?
    - Reclaiming Disk Space
    - Simplified Payment Verification
    - Combining and Splitting Value
- Conclusion
    - Pros of Bitcoin
    - Cons of Bitcoin

Why Bitcoin?

Electronic Payment Background

Online commerce rely on trusted third parties

- Reversible
- Have minimum transaction limits
- Not applicable for small amount transactions



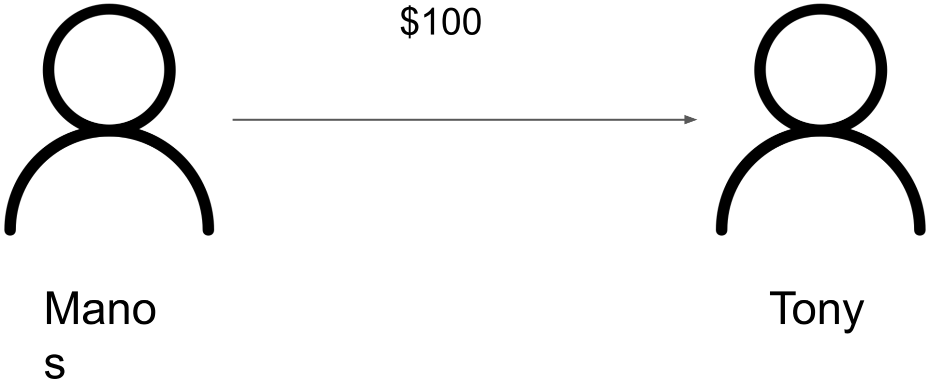
Motivation of Bitcoin

Electronic payment system based on cryptographic proof instead of trust

- Non-reversible
- Peer-to-peer
- Solves double-spending

How it works?

Transaction



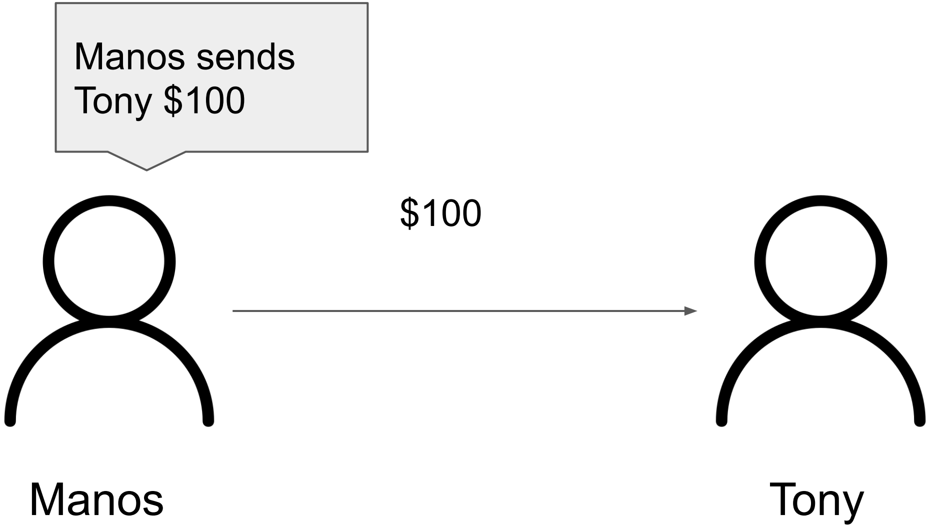
$100

Manos

Tony

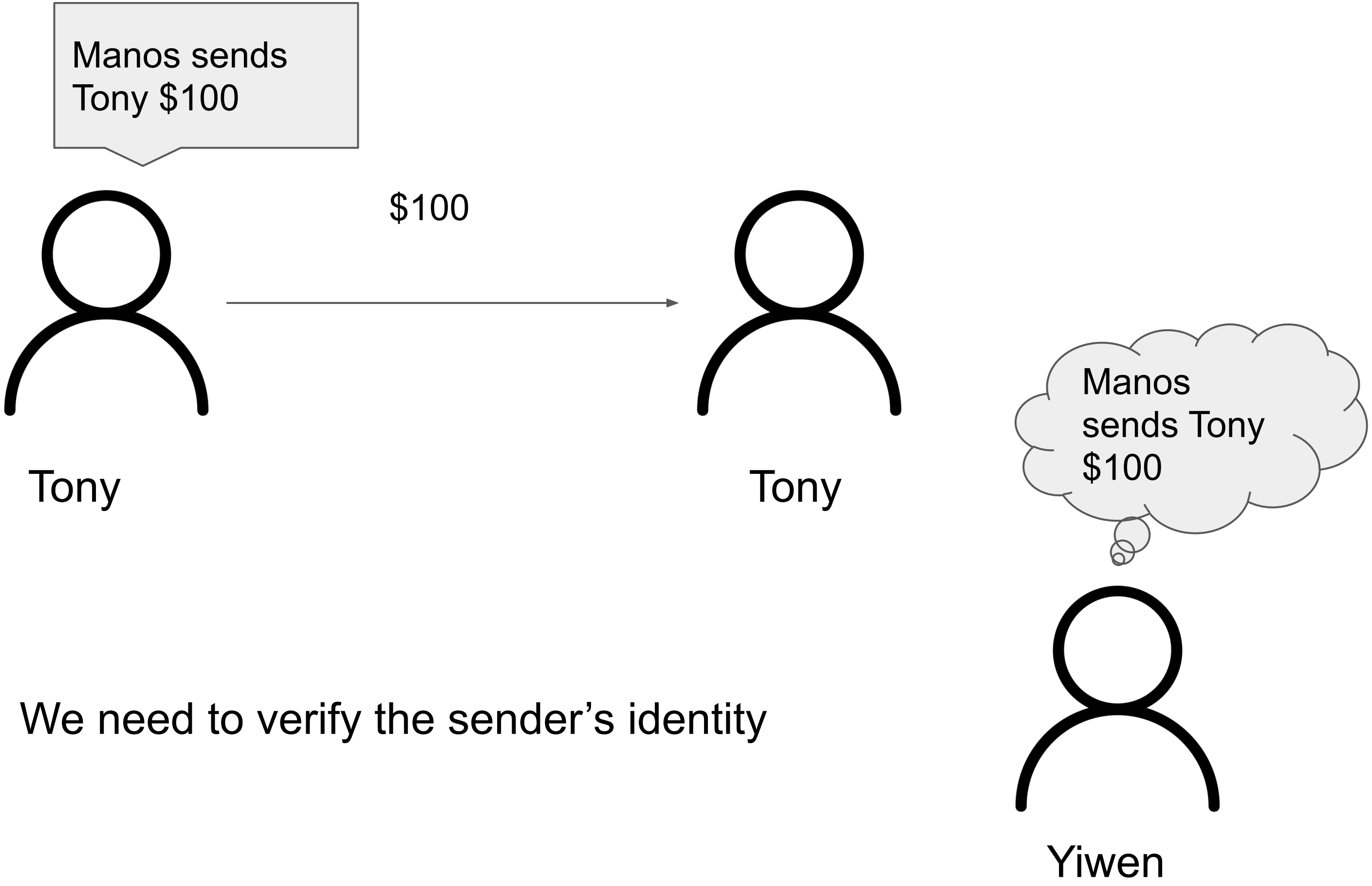
Who cares? No third Party! Only Manos and Tony knows!

Transaction



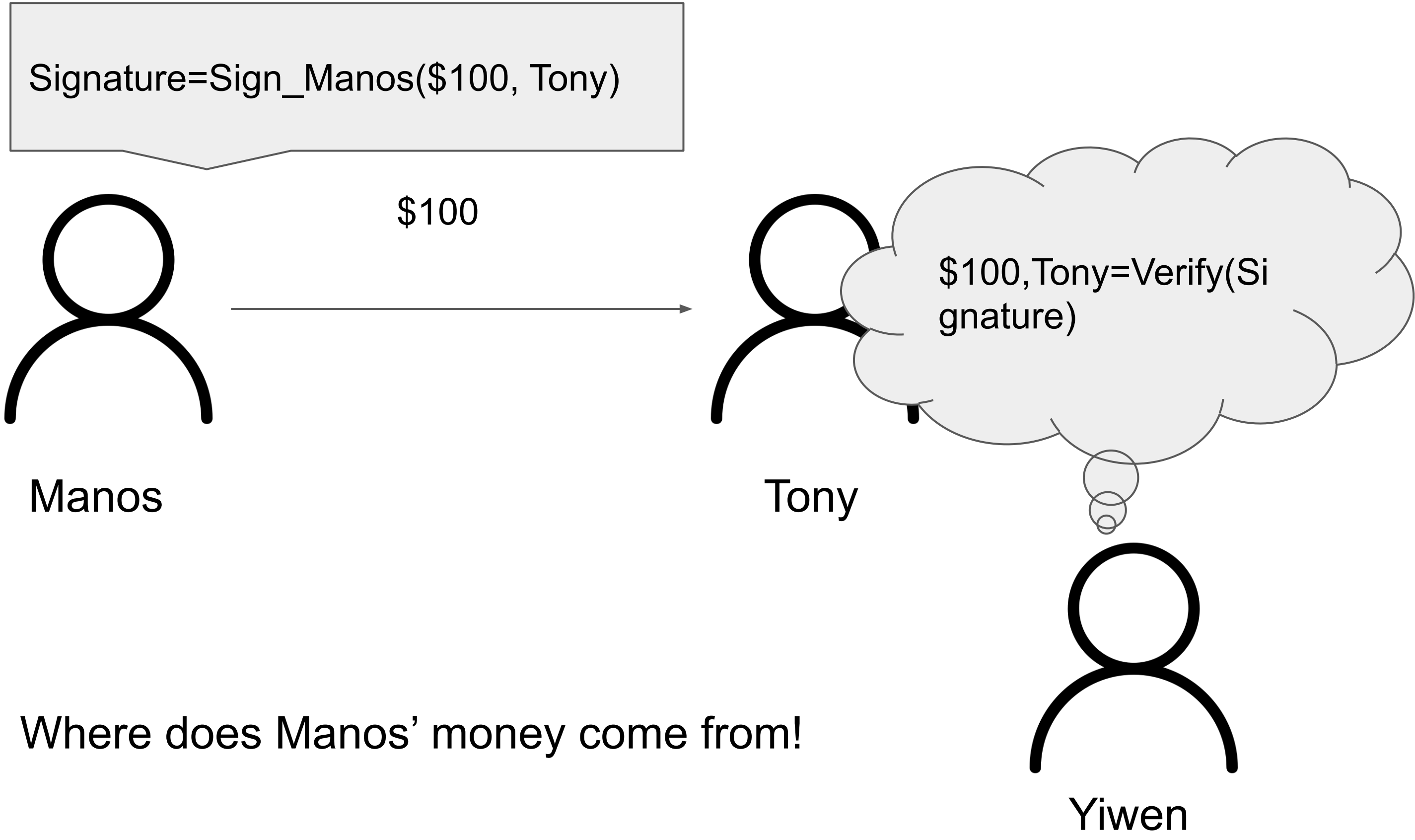
What if...

Transaction



We need to verify the sender’s identity

Transaction



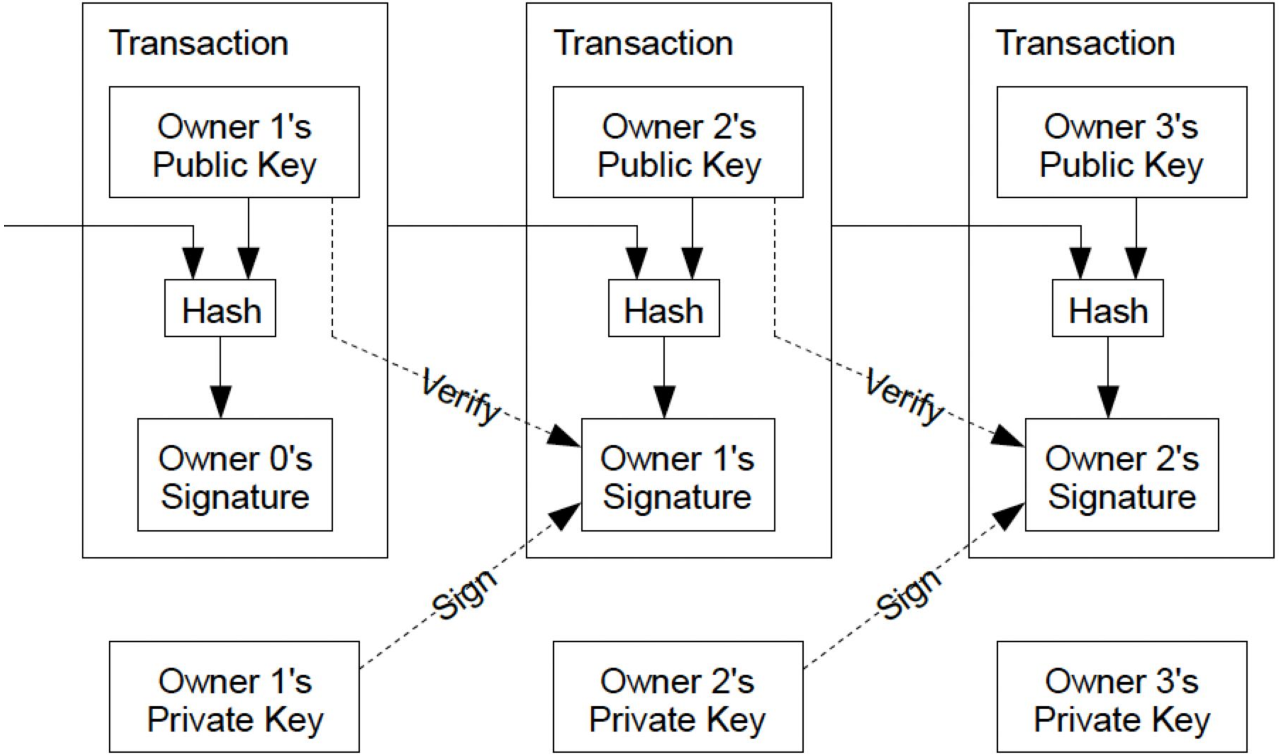
Signature=Sign_Manos($100, Tony)

$100

$100,Tony=Verify(Signature)

Where does Manos’ money come from!

Transaction

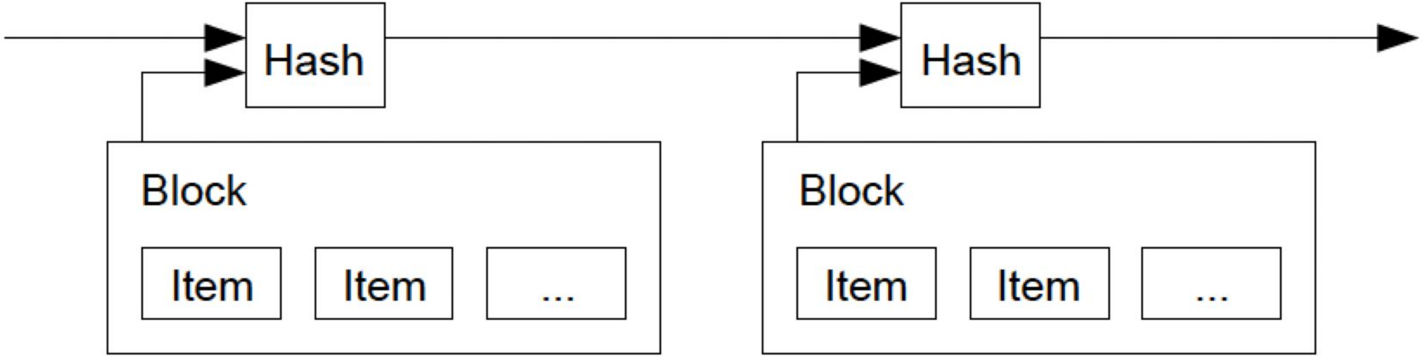


Requirements of Transactions

- Transactions must be publicly announced
- Need a system to agree on a single history of the order
- Need to prove that at the time of each translation, the majority nodes agreed

Timestamp Server

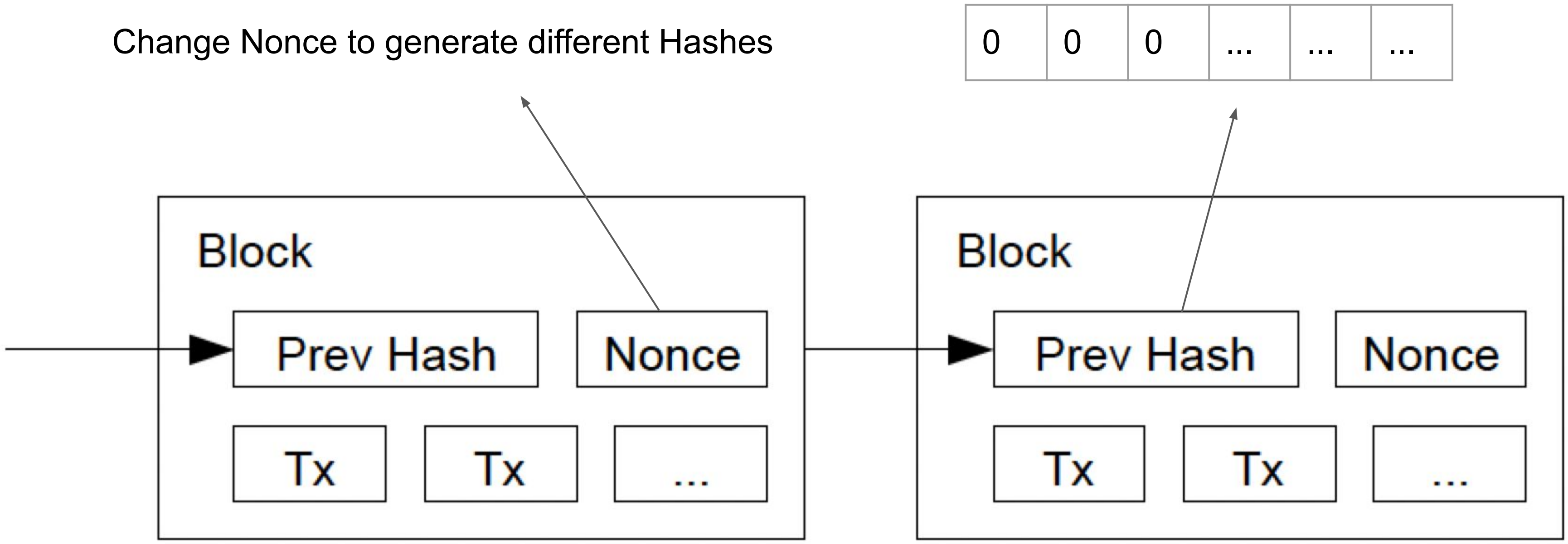
Takes a hash of a block of items to be timestamped and widely publishes the hash



Proof-of-work

N larger -> more efforts required

N 0 bits



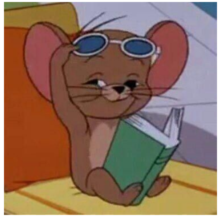
0	0	0	...	...	...

Change Nonce to generate different Hashes

Proof-of-work

- Makes the blockchain difficult to be changed
  - To change a block, need to recalculate the hash values of the target block and all blocks after
- Solves the problem of determining representation in majority decision making
  - If the majority is based on number of IP addresses -> Attacker with many IP addresses can break the system
  - CPU based majority makes the honest chain to grow fastest -> Immense efforts required to attack

Hardware speed develops fast, maybe attackers can catch up in future?



Nakamto

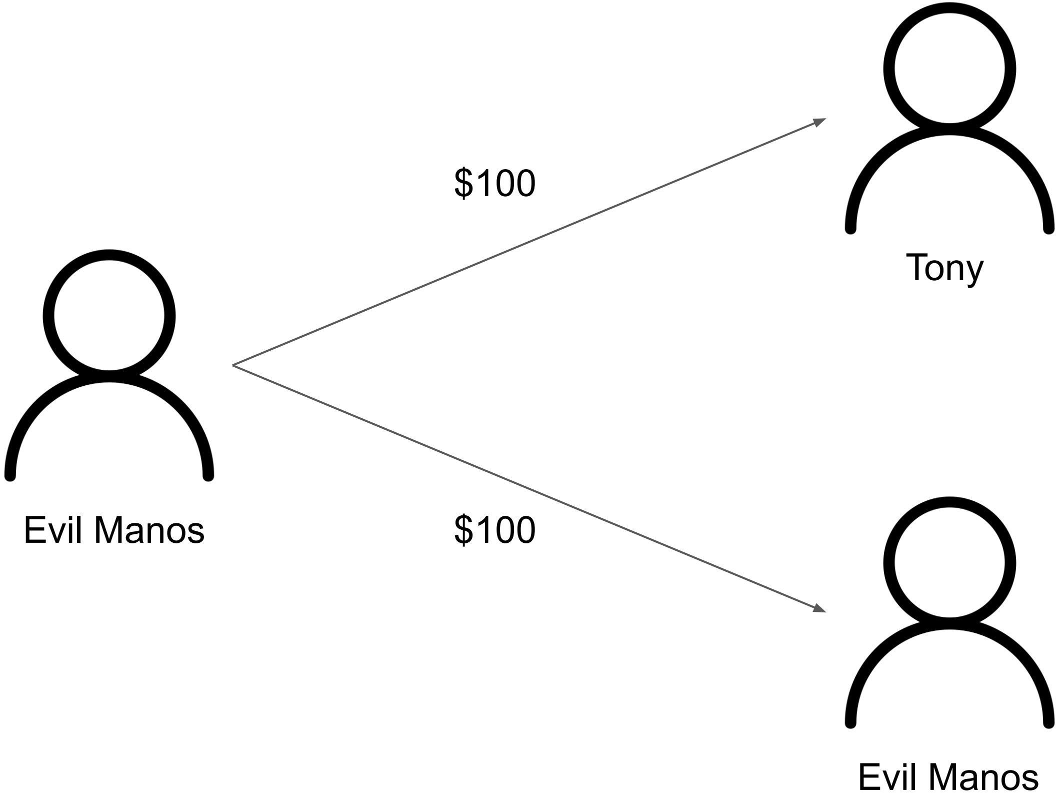
The proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. **If they are generated too fast, the difficulty increases.**

Network

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. When a node finds a proof-of-work, it broadcasts the block to all nodes.
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

What if attacked?

Double Spending



Double Spending

- Timestamp
  - Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.
- Consensus
  - The blockchain with largest proof-of-work is determined
- Proof-of-work
  - To validate an alternative chain, need to achieve a majority of CPU

Double Spending

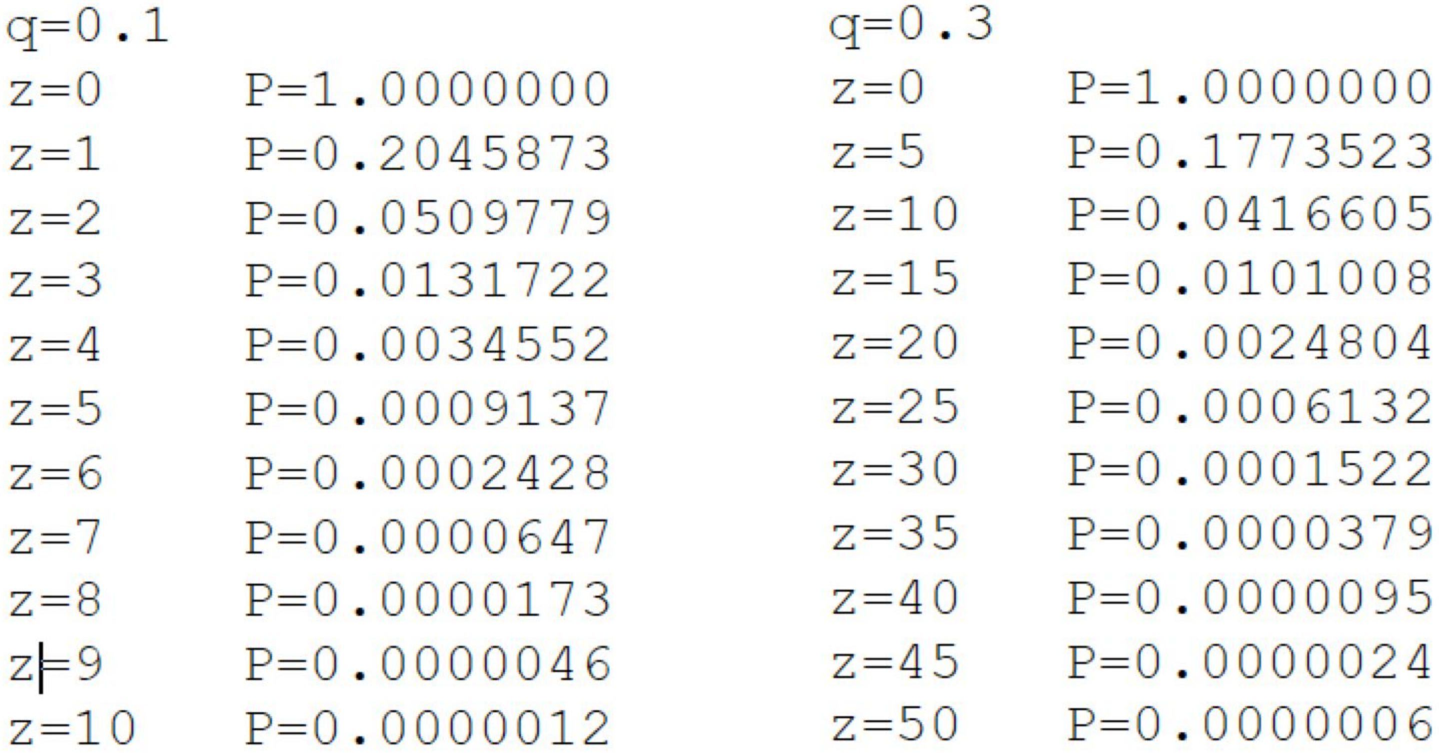
Assume there is an attacker chasing after the honest chain…

p: probability an honest node finds the next block

q: probability the attacker finds the next block

q_z: probability the attacker will ever catch up from z blocks behind

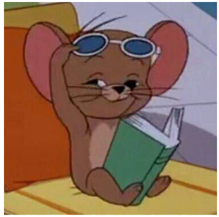
Double Spending



```
q=0.1                          q=0.3
z=0     P=1.0000000            z=0     P=1.0000000
z=1     P=0.2045873            z=5     P=0.1773523
z=2     P=0.0509779            z=10    P=0.0416605
z=3     P=0.0131722            z=15    P=0.0101008
z=4     P=0.0034552            z=20    P=0.0024804
z=5     P=0.0009137            z=25    P=0.0006132
z=6     P=0.0002428            z=30    P=0.0001522
z=7     P=0.0000647            z=35    P=0.0000379
z=8     P=0.0000173            z=40    P=0.0000095
z=9     P=0.0000046            z=45    P=0.0000024
z=10    P=0.0000012            z=50    P=0.0000006
```

Incentive

- The first translation in a block
  - A new coin
- Transaction fees
  - Difference of output value and input value
  - When the number of coins reaches some predetermined value, incentive transition entirely to transaction fees to prevent inflation



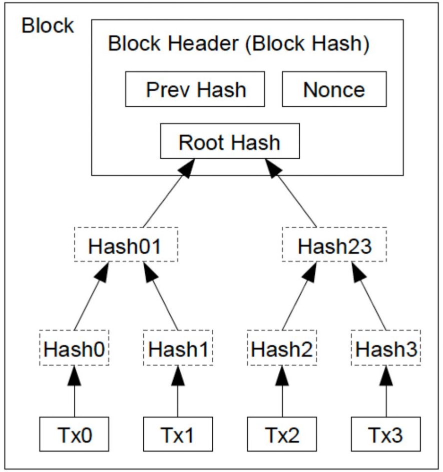
Nakamto

Since you have powerful, why not be honest? You will get rewards!

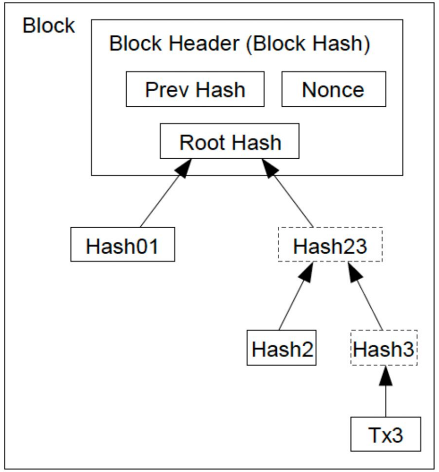
How to improve it?

Reclaiming Disk Space

- Merkle Tree - with only the root included in the block’s hash.
  - Suppose blocks generation speed 10 mins/block, block header 80 bytes
  - 80 * 6 * 24 * 365 = 4.2 MB/year



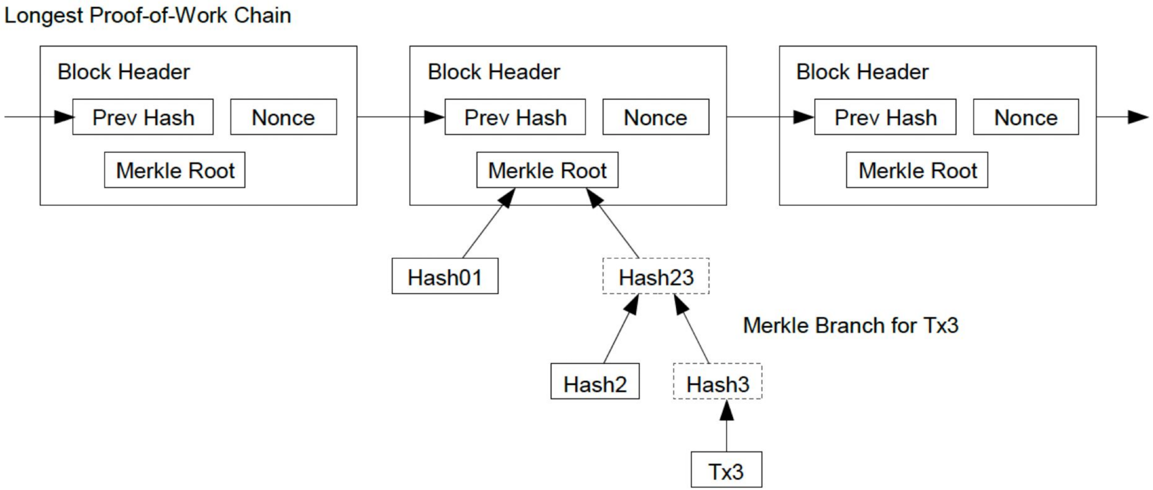
Transactions Hashed in a Merkle Tree



After Pruning Tx0-2 from the Block

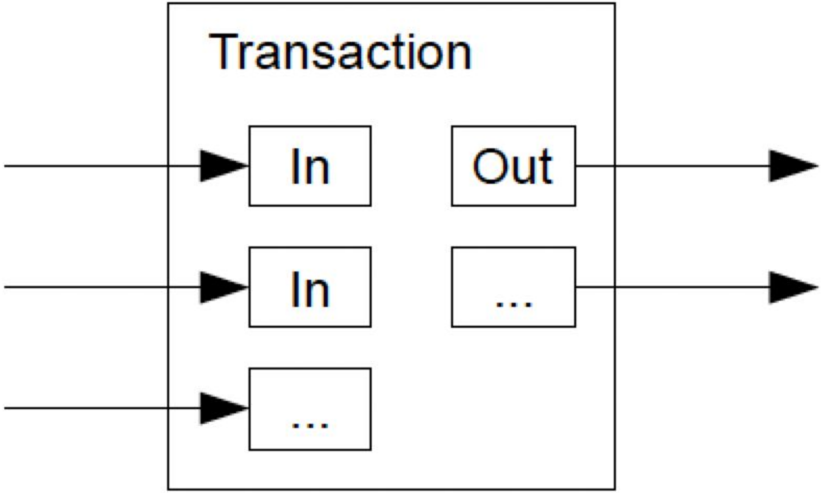
Simplified Payment Verification

- Keep a copy of the block headers of the longest proof-of-work chain
- Link the transaction to a space in the chain, if it is accepted by a network, then the transaction is valid
- Vulnerable if the network is overpowered by an attacker



Combining and Splitting Value

- A single input from a larger previous transaction or multiple inputs combining smaller amounts
- At most two outputs: payment -> payee and charge -> sender



Influence of Bitcoin

Pros of Bitcoin

- The first system of blockchain
  - A success on decentralization
- Privacy
- Hard to modify previous records
- Transparent
- Against inflation
  - Limited throughput

Cons of Bitcoin

- Long transaction time
  - Average of 10 mins
- Limited throughput
  - 21,000,000 in total
  - 3,000,000 remaining
- Large energy consumption
  - about 80 terawatt-hours
  - roughly equal to the annual output of 23 coal-fired power plants
- Graphics cards out of stock

Thanks for Listening!

Comments & Discussion