Distributed System : BitCoin & BlockChain

Houmin Wei

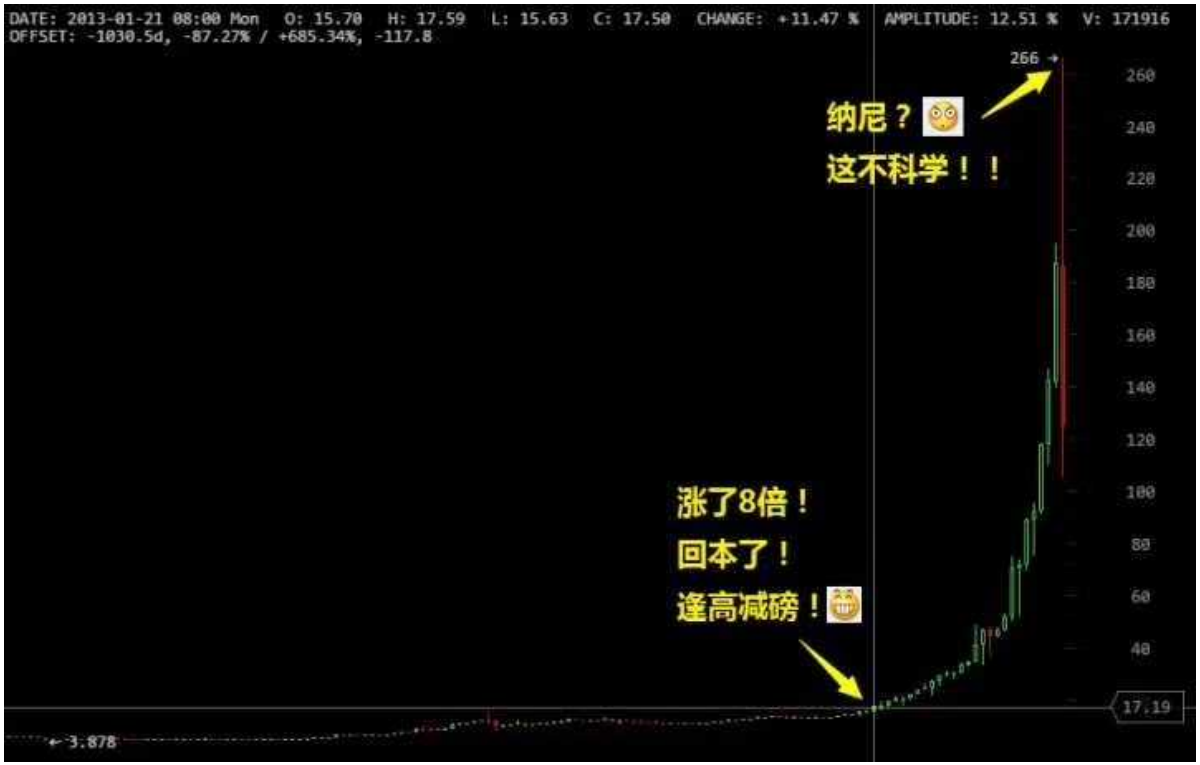
Electronics Engineering & Computer Science
Peking University

January 6, 2018

Bitcoin Mania

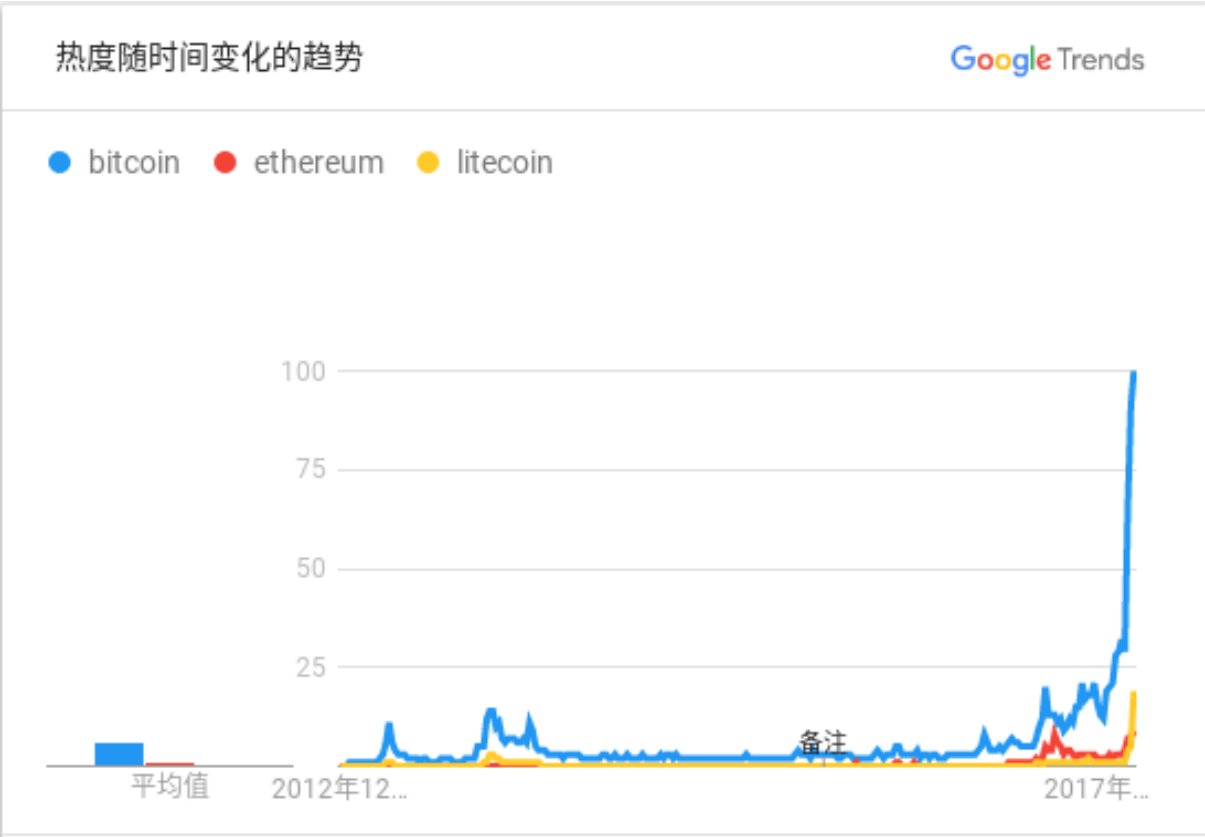
Bitcoin Mania

Bitcoin Mania

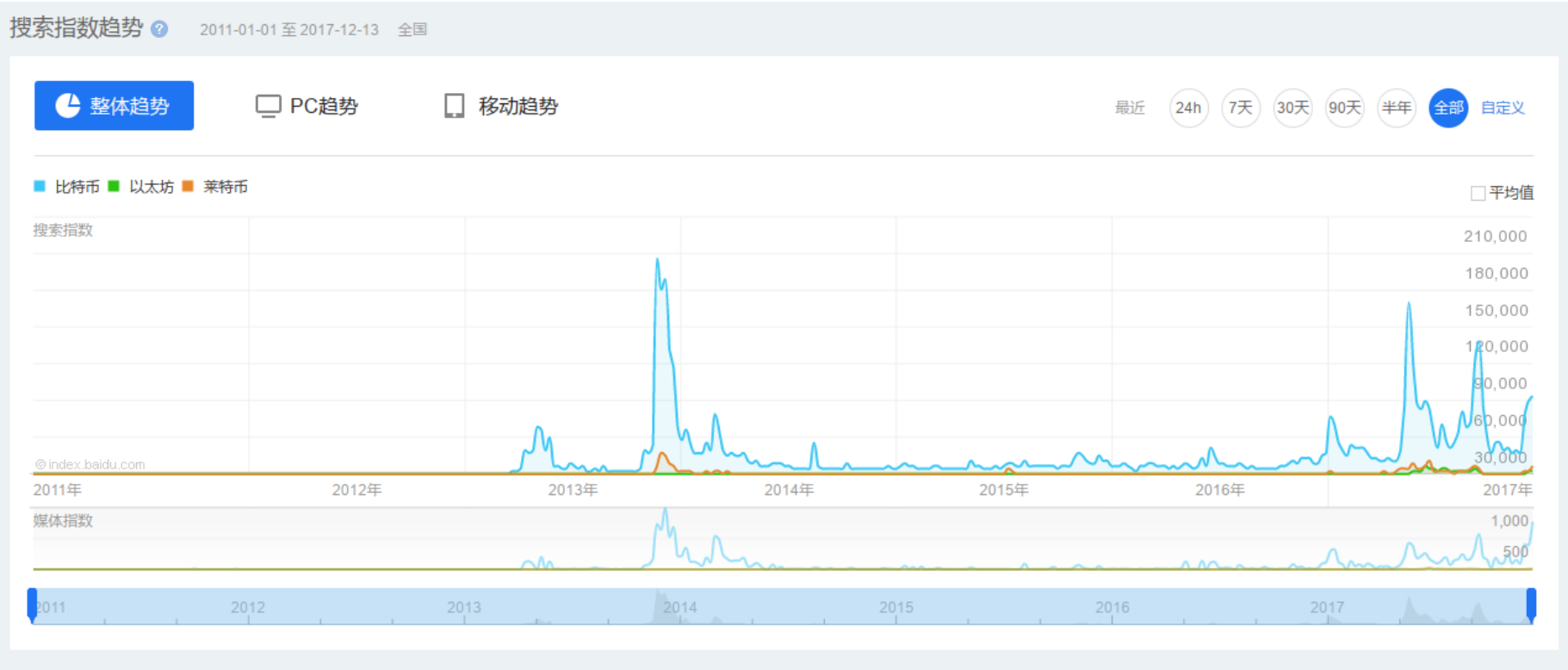


Bitcoin Mania

Bitcoin Mania



Bitcoin Mania

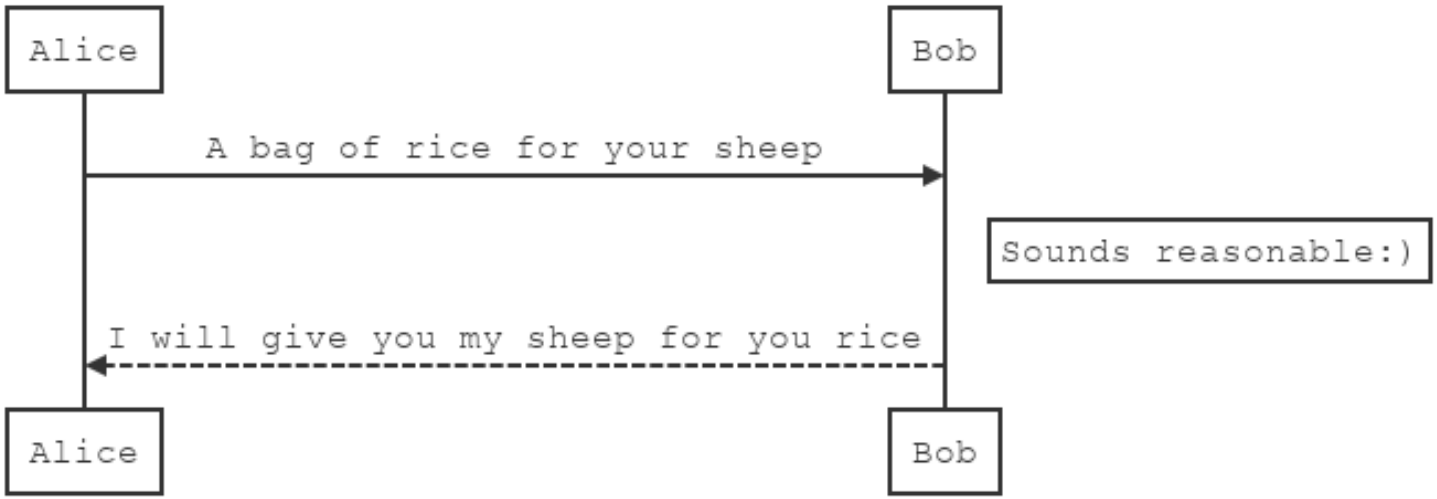


BlockChain

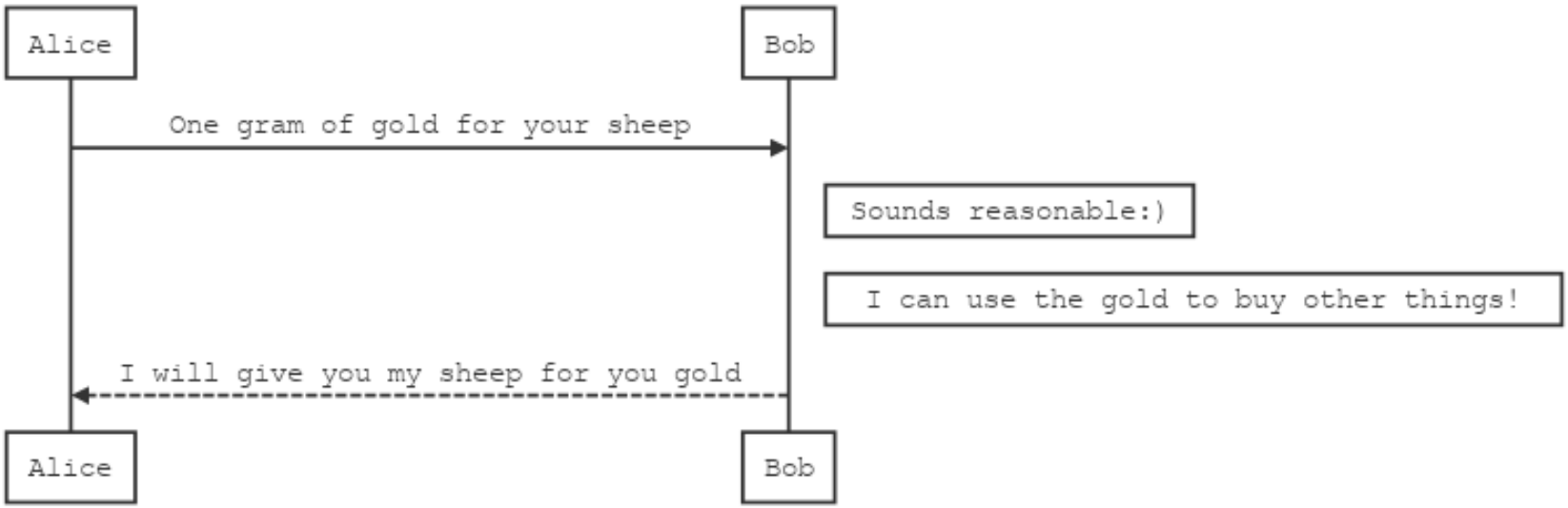
国务院 2016 年十三五报告

“十三五”时期，全球信息化发展面临的环境、条件和内涵正发生深刻变化 …信息技术创新代际周期大幅缩短，创新活力、集聚效应和应用潜能裂变式释放，更快速度、更广范围、更深程度地引发新一轮科技革命和产业变革。物联网、云计算、大数据、人工智能、机器深度学习、区块链、生物基因工程等新技术驱动网络空间从人人互联向万物互联演进，数字化、网络化、智能化服务将无处不在。现实世界和数字世界日益交汇融合，全球治理体系面临深刻变革。全球经济体普遍把加快信息技术创新、最大程度释放数字红利，作为应对“后金融危机”时代增长不稳定性和不确定性、深化结构性改革和推动可持续发展的关键引擎。

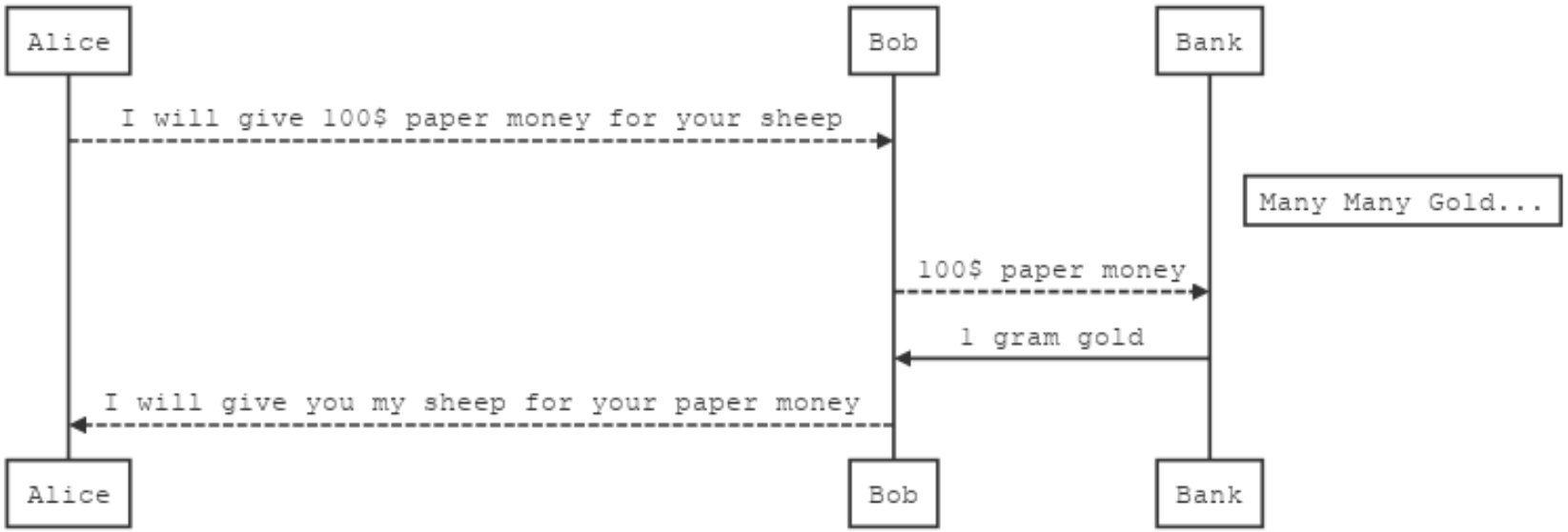
Barter System



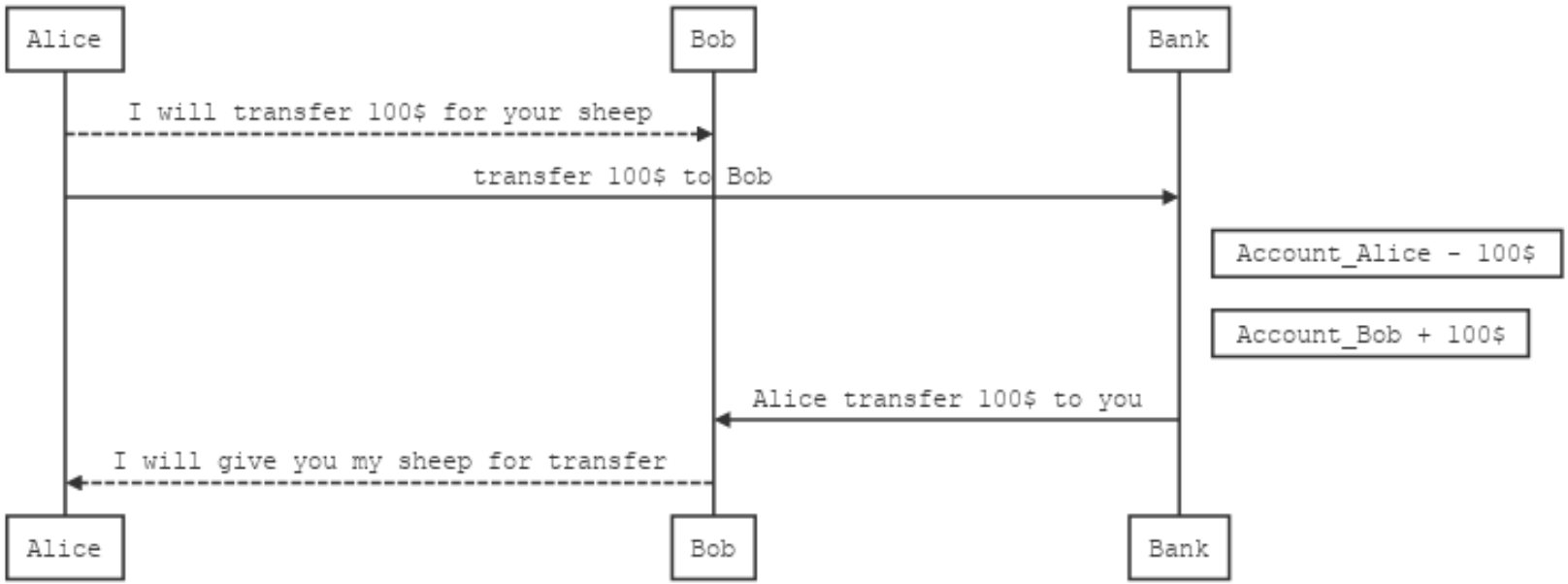
Gold Money


Alice
Bob
One gram of gold for your sheep
Sounds reasonable:)
I can use the gold to buy other things!
I will give you my sheep for you gold
Alice
Bob

Paper Money


Alice
Bob
Bank
I will give 100$ paper money for your sheep
Many Many Gold...
100$ paper money
1 gram gold
I will give you my sheep for your paper money
Alice
Bob
Bank

Central Banking



Nature of Money

What is Money?

- **Medium of exchange**
  - Standard object used in exchanging goods and services
- **Unit of account**
  - Standard unit used for quoting prices
- **Store of value**
  - Store wealth from one point in time to another

What is Bitcoin?

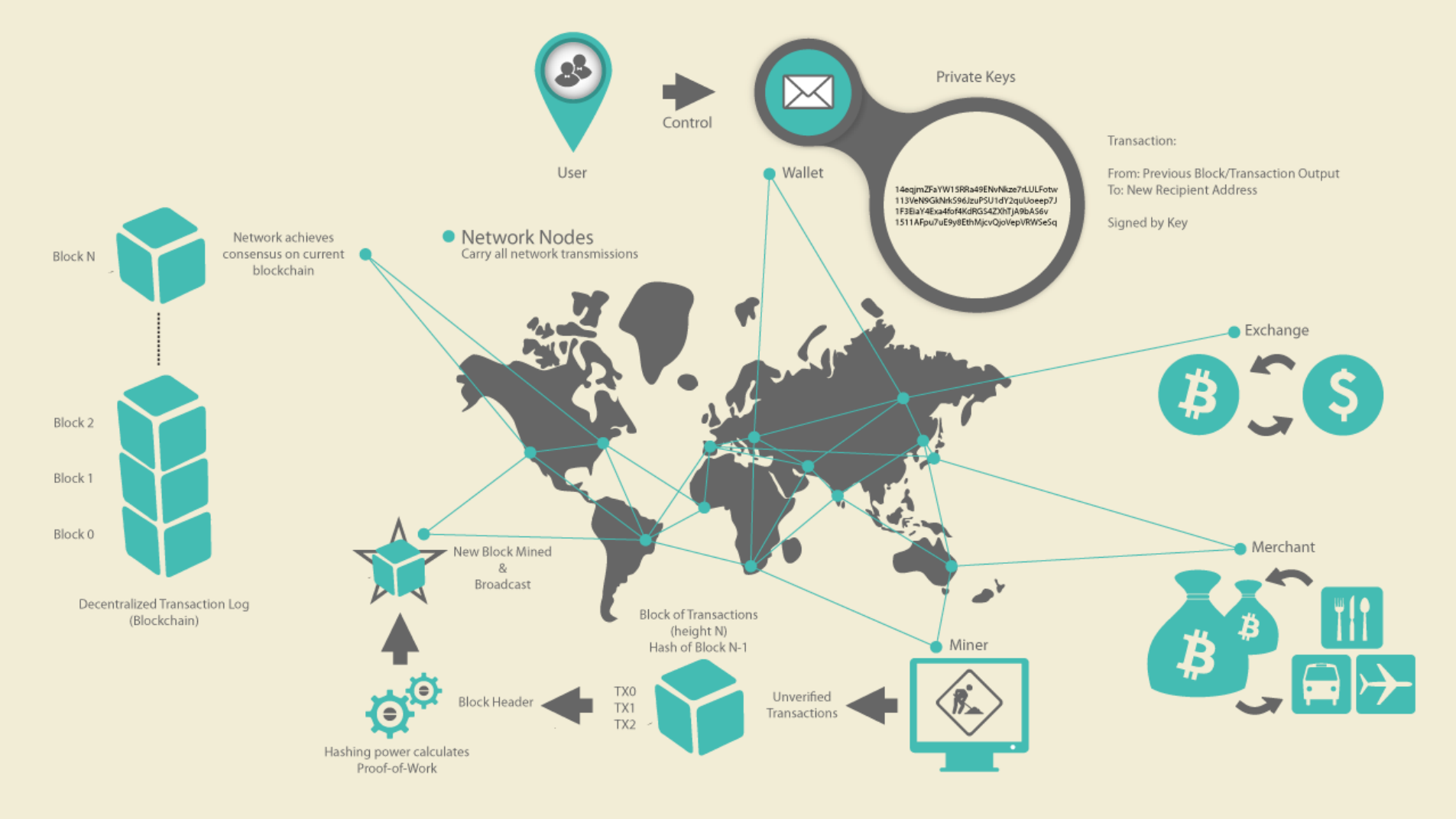
Wikipedia

Bitcoin is the first decentralized digital cryptocurrency, as the worldwide payment system works without a central bank or single administrator. The network is peer-to-peer and transactions take place between users directly through the use of cryptography, without an intermediary. These transactions are verified by network nodes, which called mining and recorded in a public distributed ledger called a blockchain. Bitcoin was invented by an unknown person or group of people under the name Satoshi Nakamoto and released as open-source software in 2009.

Video

Bitcoin, What is it?

Bitcoin Overview



Buying a cup of coffee

Alice buyes a cup of coffee at Bob's coffee shop, paying with BTC.

```
bitcoin:1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA?\
amount=0.015&\
label=Bob%27s%20Cafe&\
message=Purchase%20at%20Bob%27s%20Cafe

Components of the URL
A bitcoin address: "1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA"
The payment amount: "0.015"
A label for the recipient address: "Bob's␣Cafe"
A description for the payment: "Purchase␣at␣Bob's␣Cafe"

```

Q & A

Bitcoin: Challenges

- **Creation of a virtual coin**
  - How is it created in the first place?
  - How do you prevent inflation?
- **Validation**
  - Is the coin legit? (Proof-Of-Work)
  - How do you prevent a coin from double-spending?
- **Buyer and seller protection in online transactions**
  - Buyer pays, but the seller doesn't deliver
  - Seller delivers, buyer pays, but the buyer makes a claim
- **Trust on third party**
  - Rely on proof instead of trust
  - Verifiable by everyone
  - No central bank

CryptoCurrency

Bitcoin Overview

```
function mine()
{
    while (true)
    {
        longestChain = getLongestValidChain();

        // A number that changes every time, so that you don't waste time
        // trying to calculate a valid blockHash with the same input.
        nonce = getNewNonce();

        currentTXs = getUnconfirmedTransactionsFromNetwork();
        newBlock = getNewBlock(longestChain, currentTX, nonce);

        // Hash function, and this is what all the *mining machines* are doing
        blockHash = sha256(newBlock)

        if (meetRequirements(blockHash))
        {
            broadcast(newBlock)
            // now the height of the block chain is incremented by 1
            // if the new block is accepted by other peers
            // and all the TXs in the new block are "confirmed"
        }
    }
}

```

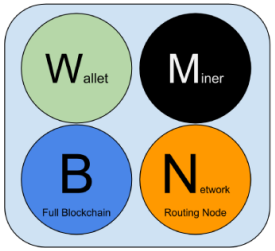
Bitcoin Overview

```
1 function sendBTC(amount)
2 {
3     sourceTXs = pickConfirmedTransactionsToBeSpent(amount);
4     tx = generateTX(sourceTx, targetAddrs, amount, fee);
5     signedTx = sign(tx, privateKeysOfAllInputAddress);
6     broadcast(signedTx);
7 }
8
```

P2P Network

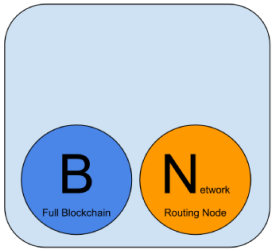
The steps to run the network are as follows:

- New transactions are broadcast to all nodes.
- Each node collects new transactions into a block.
- When node finds a proof-of-work, it broadcast the block to all nodes.
- Nodes accept the block only if all transactions in it are valid and not already spent.
- 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 previous hash.



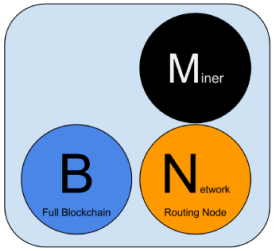
Reference Client (Bitcoin Core)

Contains a Wallet, Miner, full Blockchain database, and Network routing node on the bitcoin P2P network.



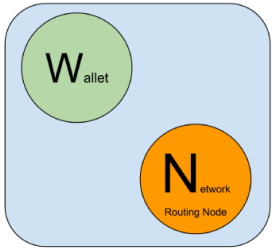
Full Block Chain Node

Contains a full Blockchain database, and Network routing node on the bitcoin P2P network.



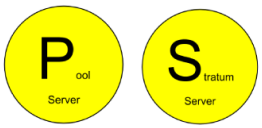
Solo Miner

Contains a mining function with a full copy of the blockchain and a bitcoin P2P network routing node.



Lightweight (SPV) wallet

Contains a Wallet and a Network node on the bitcoin P2P protocol, without a blockchain.

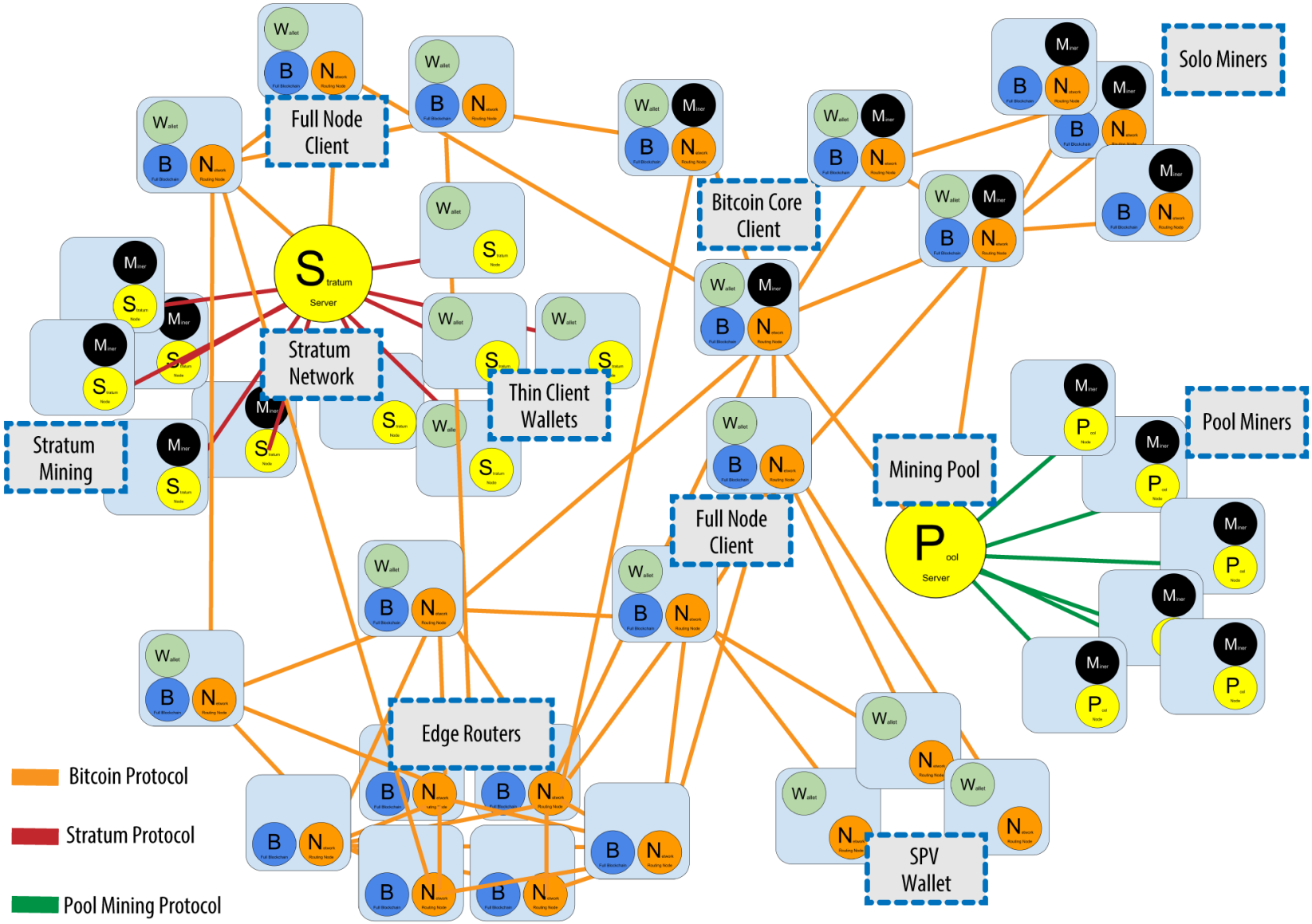


Pool Protocol Servers

Gateway routers connecting the bitcoin P2P network to nodes running other protocols such as pool mining nodes or Stratum nodes.

Mining Nodes

Contain a mining function, without a blockchain, with the Stratum protocol node (S) or other pool (P) mining protocol node.


Full Node Client
Solo Miners
Bitcoin Core Client
Stratum Server
Stratum Network
Thin Client Wallets
Stratum Mining
Full Node Client
Mining Pool
Pool Server
Pool Miners
Edge Routers
SPV Wallet
Bitcoin Protocol
Stratum Protocol
Pool Mining Protocol

Security in Bitcoin

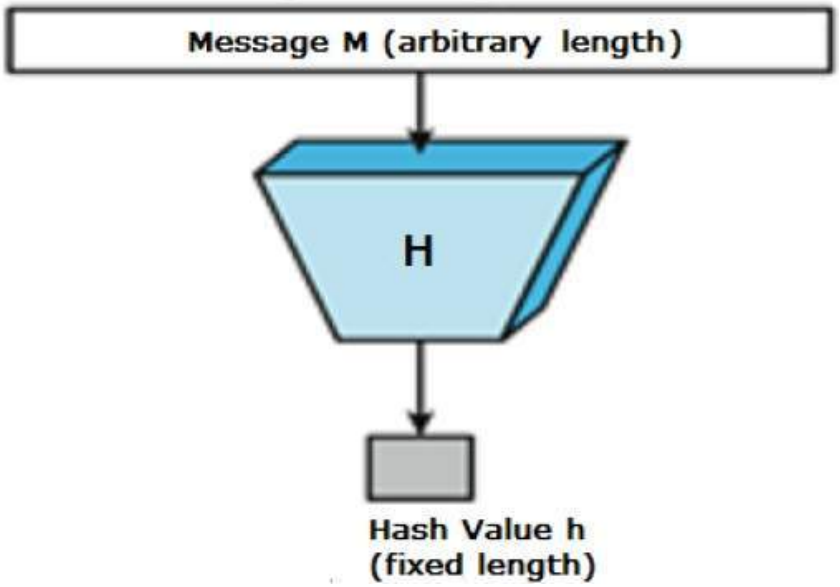
- **Authentication**
  - Am I paying the right person?
- **Integrity**
  - Is the coin double-spent?
  - Can an attacker reverse or change transactions?
- **Availability**
  - Can I make a transactions anytime I want?
- **Confidentiality**
  - Are my transactions private? Anonymous?

Security in Bitcoin

- **Authentication** -> **Public Key Crypto: Digital Signatures**
  - Am I paying the right person?
- **Integrity** -> **Digital Signatures and Cryptographic Hash**
  - Is the coin double-spent?
  - Can an attacker reverse or change transactions?
- **Availability** -> **Broadcast messages to the P2P network**
  - Can I make a transactions anytime I want?
- **Confidentiality** -> **Pesudonymity**
  - Are my transactions private? Anonymous?

Cryptographic Hash Function

- **Computationally efficient**
- **Consistent**
  hash(x) always yields same result.
- **Collision Resistant**
  Given $hash(W) = Z$, hard to find X such that $hash(X) = Z$
- **One-way**
  Given Y, hard to find X s.t. $hash(X) = Y$



Common Hash Functions:

- **MD5**
- **SHA-1**
- **SHA-2**
  - SHA-256
  - SHA-384
  - SHA-512

Hash Function Example

```
1    SHA224("")
2    0x d14a028c2a3a2bc9476102bb288234c415a2b01f828ea62ac5b3e42f
3    SHA256("")
4    0x e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
5
```

Even a small change in the message wil result in a mostly different hash.

```
1    SHA224("The␣quick␣brown␣fox␣jumps␣over␣the␣lazy␣dog")
2    0x 730e109bd7a8a32b1cb9d9a09aa2325d2430587ddbc0c38bad911525
3    SHA224("The␣quick␣brown␣fox␣jumps␣over␣the␣lazy␣dog.")
4    0x 619cba8e8e05826e9b8c519c0a5c68f4fb653e8a3d8aa04bb2c8cd4c
5
```

Proof of work first sight:

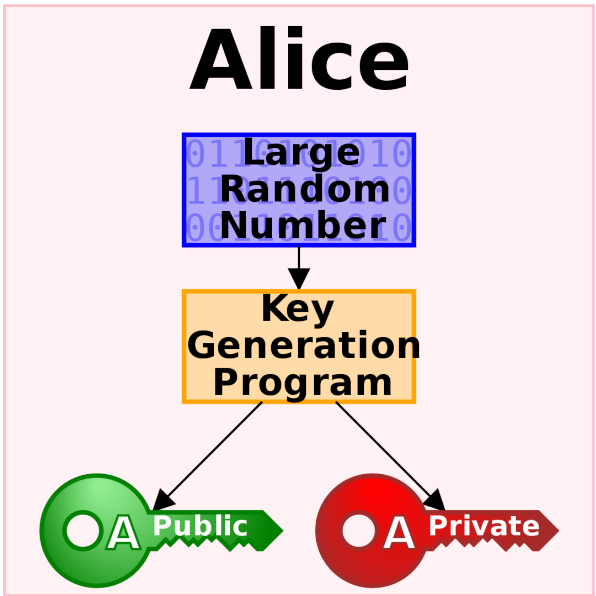
Given a basic string hello world! + random number nonce

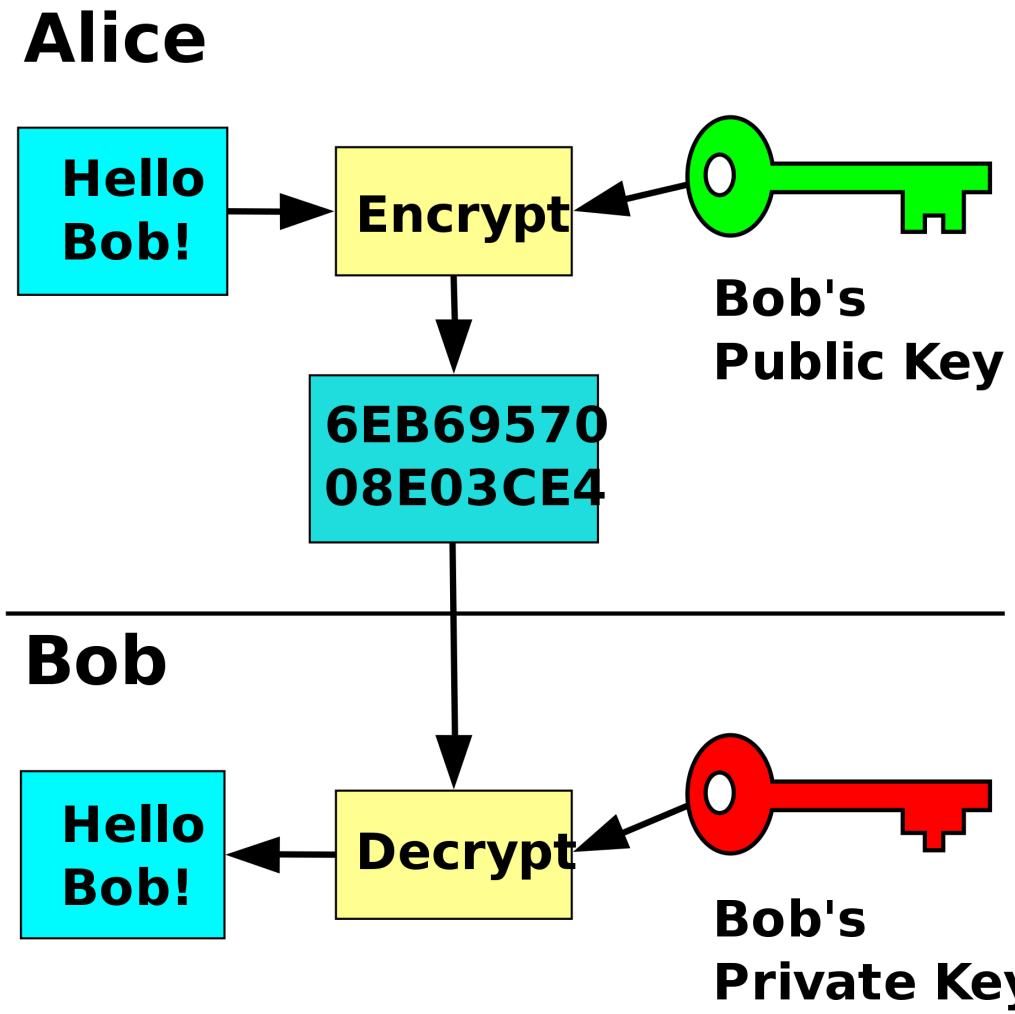
We need the digest have 4 leading 0.

```
1    "Hello,␣world!0" => 1312af178c253f84028d480a6adc1e25e81caa44c749ec81976192e2ec934c64
2    "Hello,␣world!1" => e9afc424b79e4f6ab42d99c81156d3a17228d6e1eef4139be78e948a9332a7d8
3    "Hello,␣world!2" => ae37343a357a8297591625e7134cbea22f5928be8ca2a32aa475cf05fd4266b7
4    ...
5    "Hello,␣world!4248" => 6e110d98b388e77e9c6f042ac6b497cec46660deef75a55ebc7cfdf65cc0b965
6    "Hello,␣world!4249" => c004190b822f1669cac8dc37e761cb73652e7832fb814565702245cf26ebb9e6
7    "Hello,␣world!4250" => 0000c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464e12dcd4e9
8
```

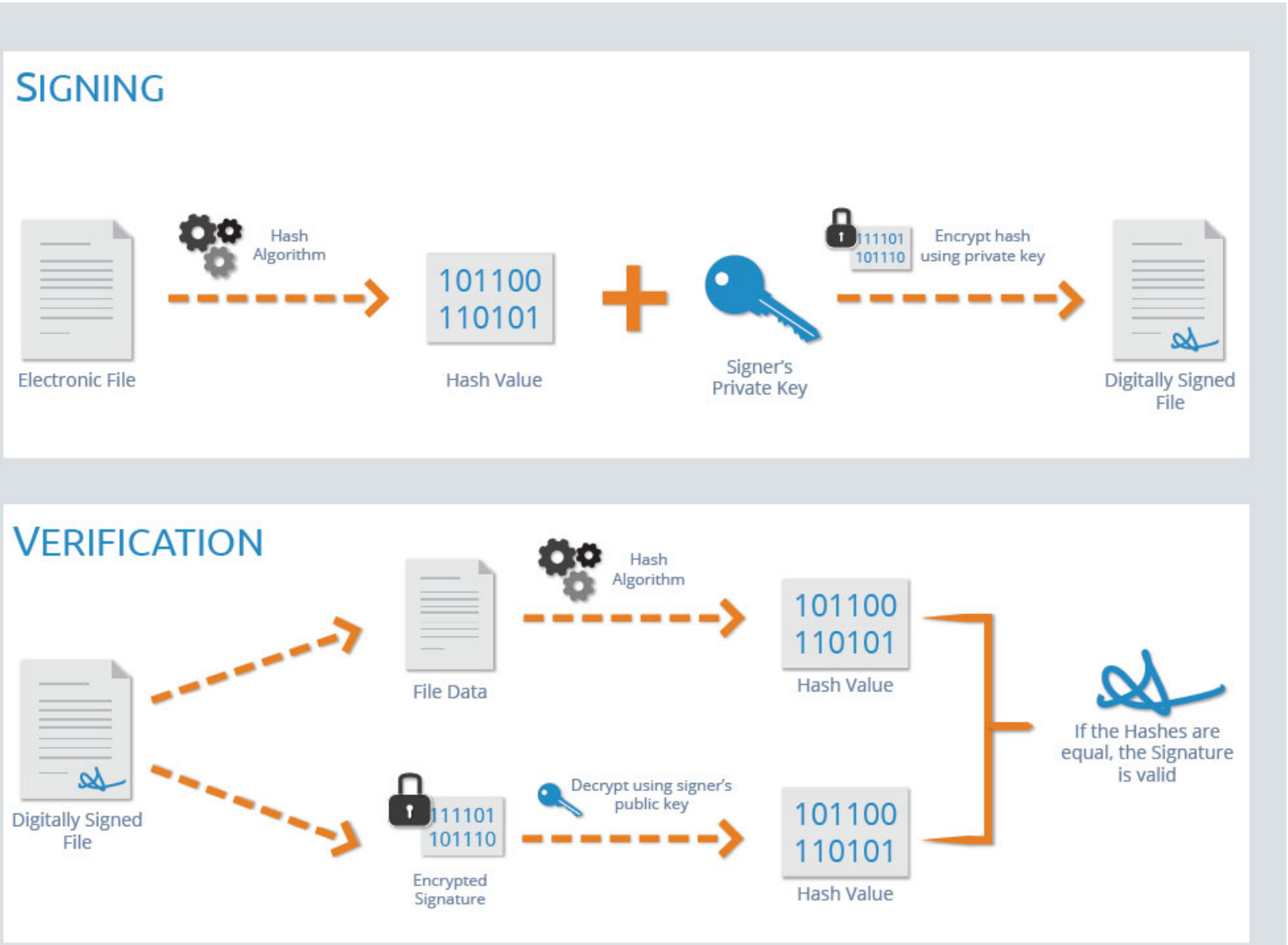
Public Key Crypto: Encryption

Key pair: Public Key and Private Key

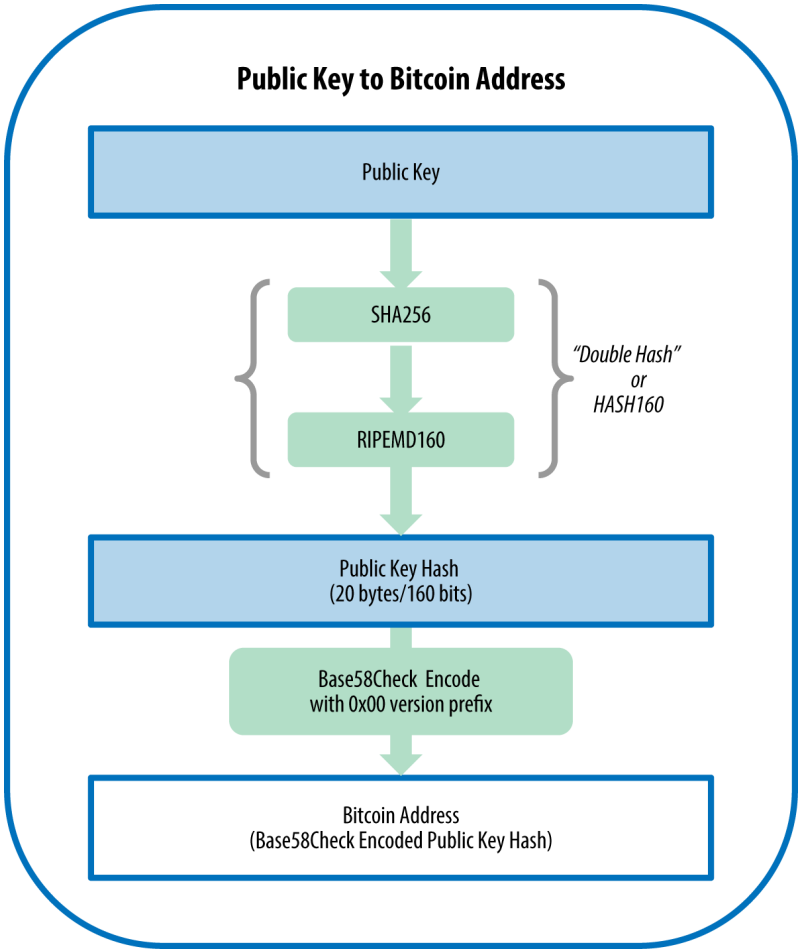




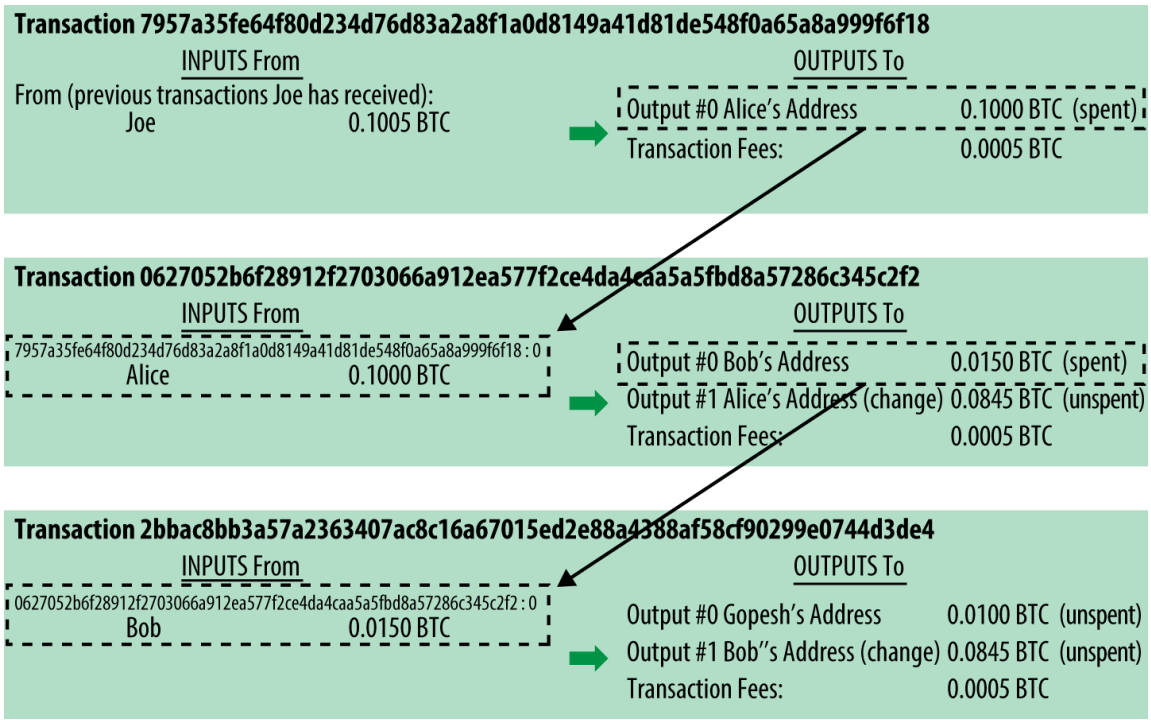
Public Key Crypto: Digital Signature



Public Key to Bitcoin Address



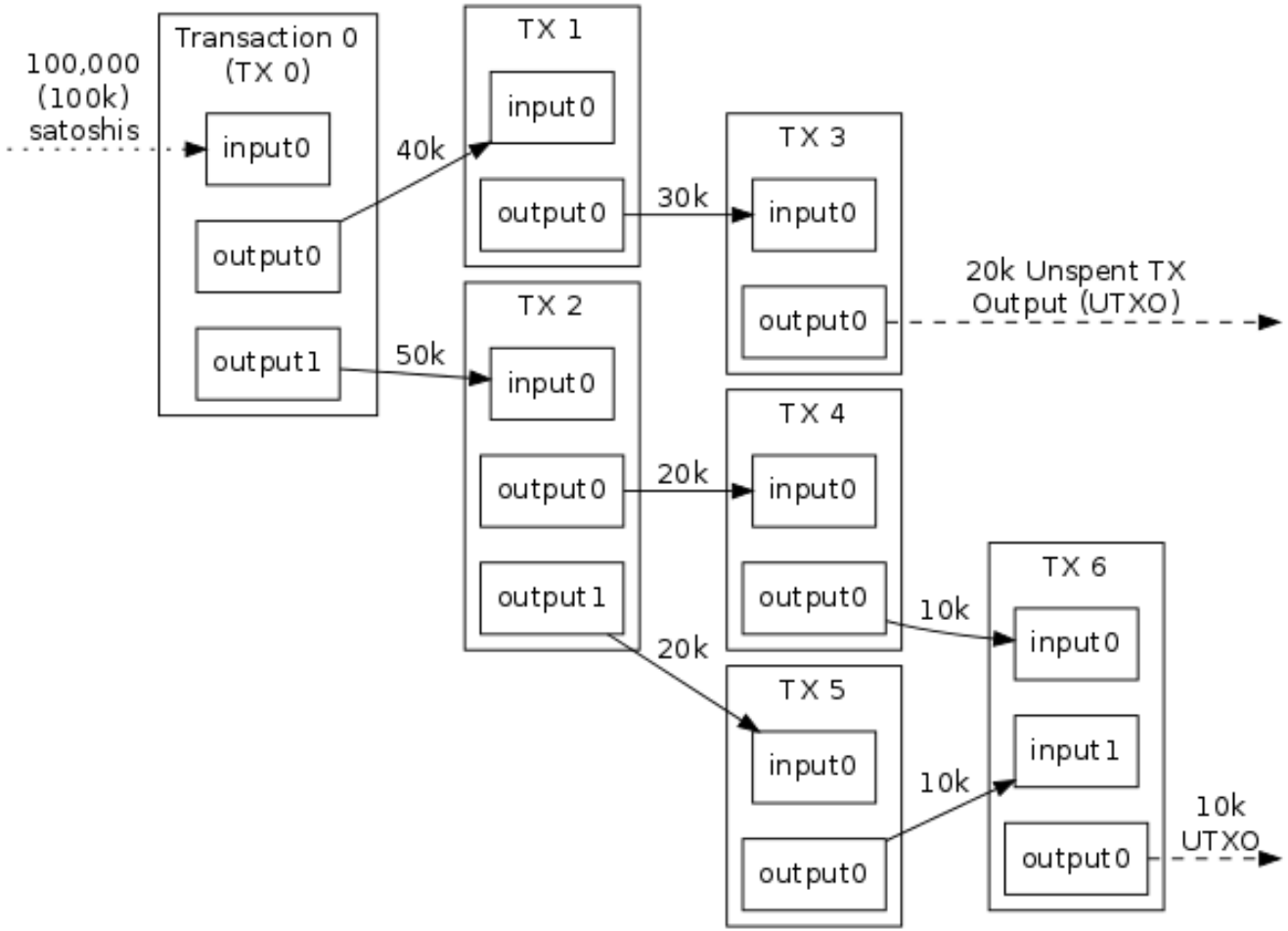
Transactions



- **transaction input**: Where the coin is coming from, usually a previous transaction's output.
- **transaction output**: Assigns a new owner to the value by associating it with a key.
- **transaction fees**: Small payment collected by the miner.

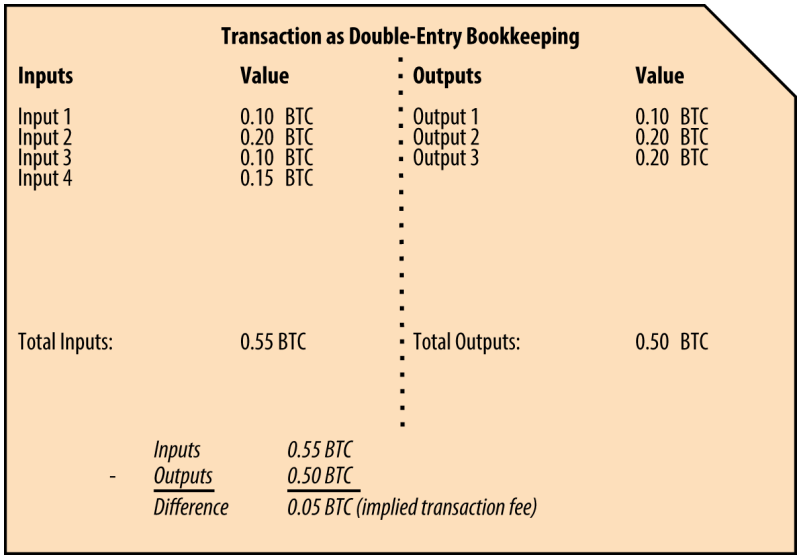
UTXO(Unspent Transaction Output)

- UXTO is the fundamental building block of a bitcoin transaction.
- There are NO Accounts in bitcoin, there are ONLY UTXO.



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin

Transactions



Transaction as Double-Entry Bookkeeping

Inputs	Value	Outputs	Value
Input 1	0.10 BTC	Output 1	0.10 BTC
Input 2	0.20 BTC	Output 2	0.20 BTC
Input 3	0.10 BTC	Output 3	0.20 BTC
Input 4	0.15 BTC		
Total Inputs:	0.55 BTC	Total Outputs:	0.50 BTC

	Inputs	*0.55 BTC*
-	*Outputs*	*0.50 BTC*
	Difference	*0.05 BTC (implied transaction fee)*

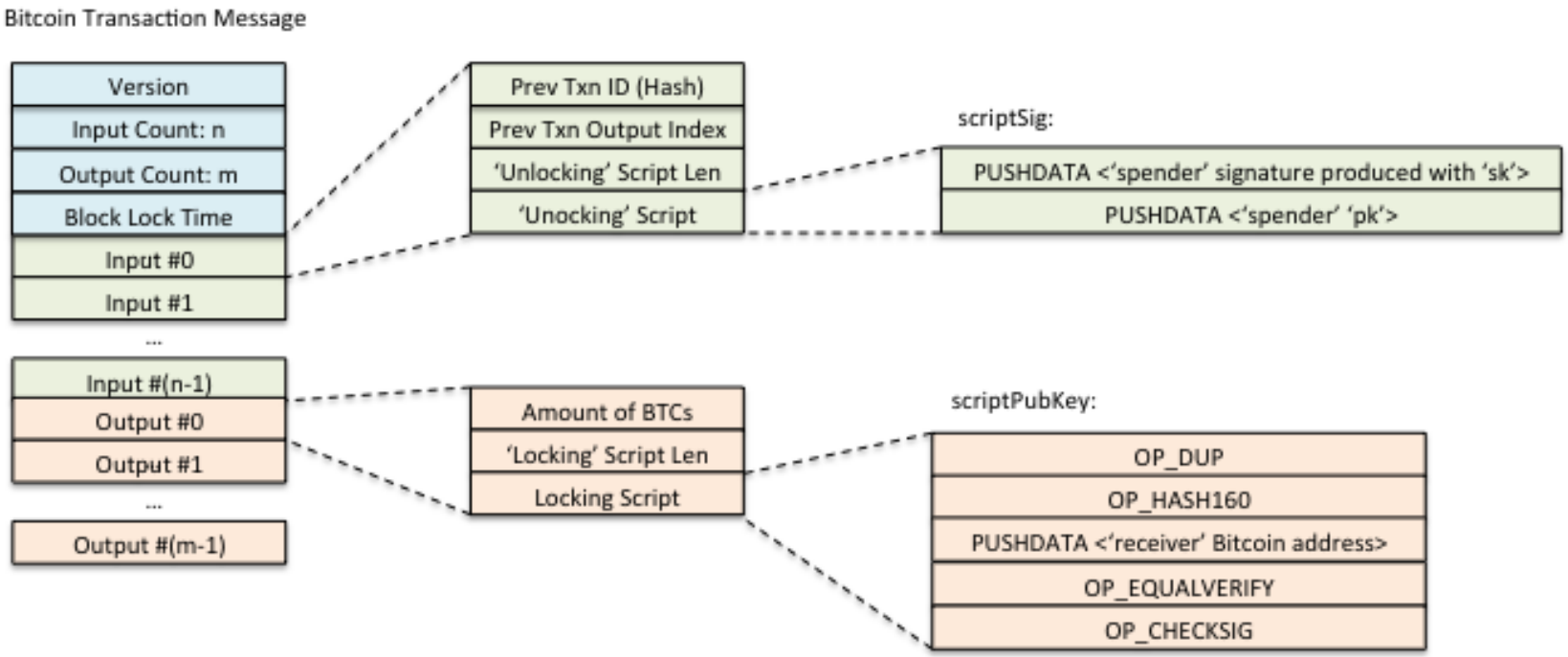
Table 5-1. The structure of a transaction

Size	Field	Description
4 bytes	Version	Specifies which rules this transaction follows
1-9 bytes (VarInt)	Input Counter	How many inputs are included
Variable	Inputs	One or more Transaction Inputs
1-9 bytes (VarInt)	Output Counter	How many outputs are included
Variable	Outputs	One or more Transaction Outputs
4 bytes	Locktime	A unix timestamp or block number

- A transaction is a transfer of bitcoin value that is broadcast to the network and collected into blocks.
- A transaction typically references previous transaction outputs as new transaction inputs and dedicates all input Bitcoin values to new outputs.
- Once transactions are buried under enough confirmations they can be considered irreversible.

Regular Bitcoin Transaction

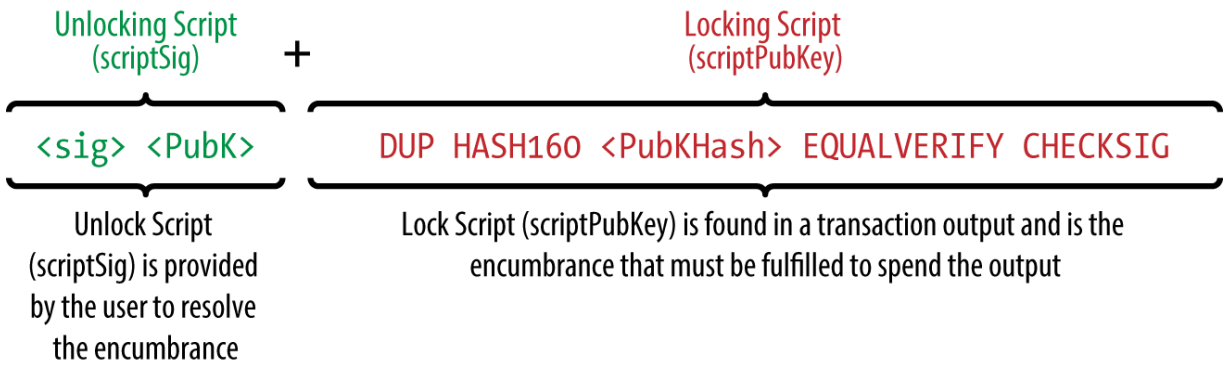
Transaction Data Structure:



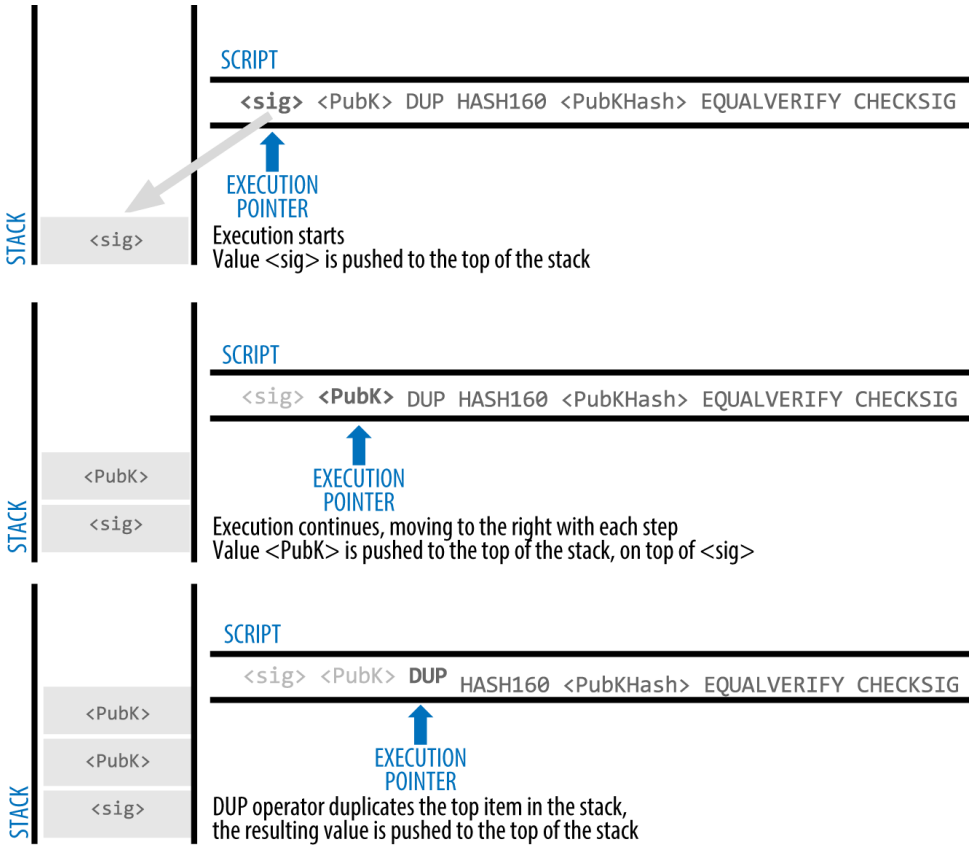
Transaction Example: The Famous Pizza Transaction

Transaction Script: Lock + Unlock

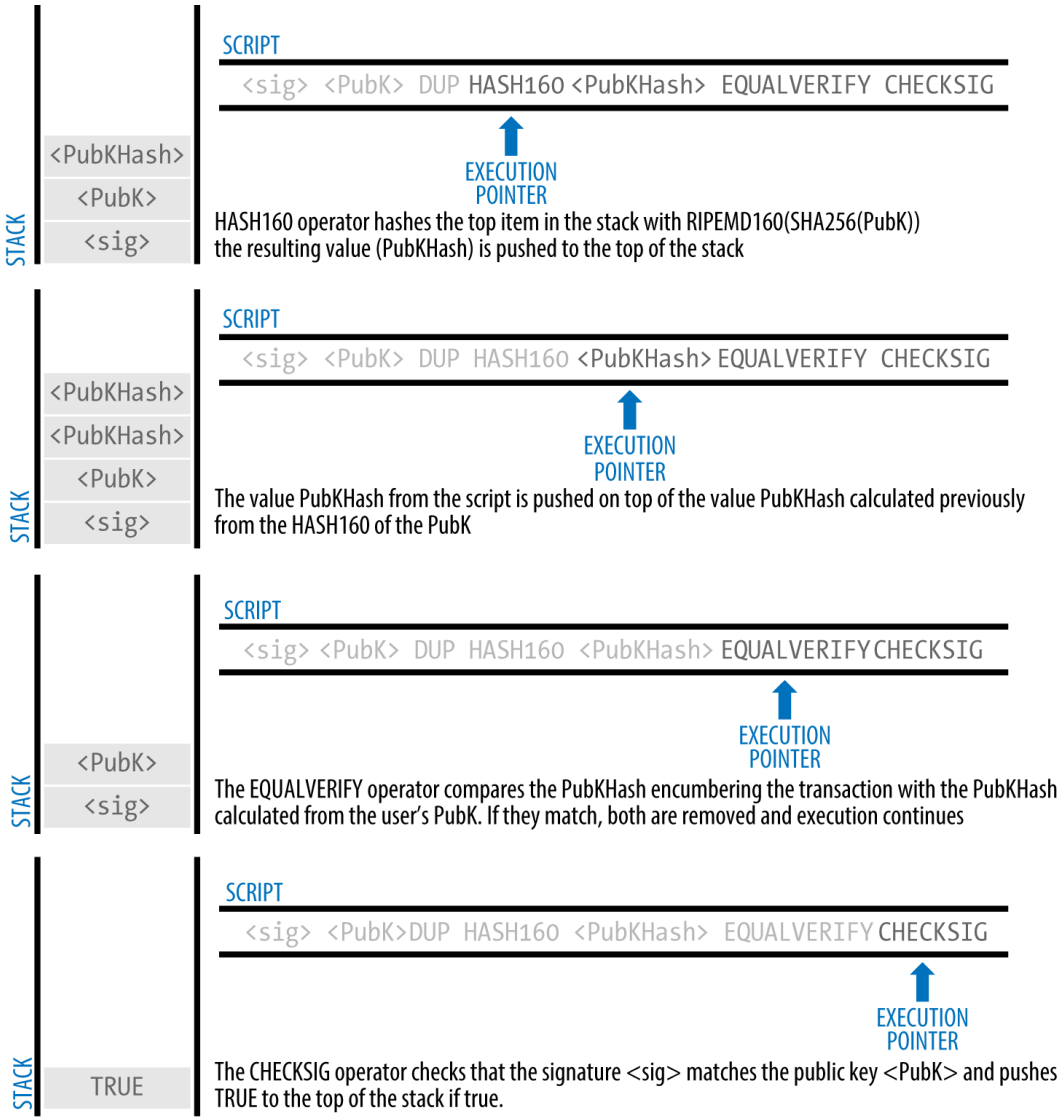
- Transaction Verification relies on Locking Script and Unlocking Script
- A Locking Script(scriptPubkey) specifies the conditions that must be met to spend the output in the future.
- An Unlocking Script(scriptSig) solves the conditions placed on an output by a Locking Script and allows the output to be spent.
- Every bitcoin client will validate transactions by executing the locking and unlocking scripts together.



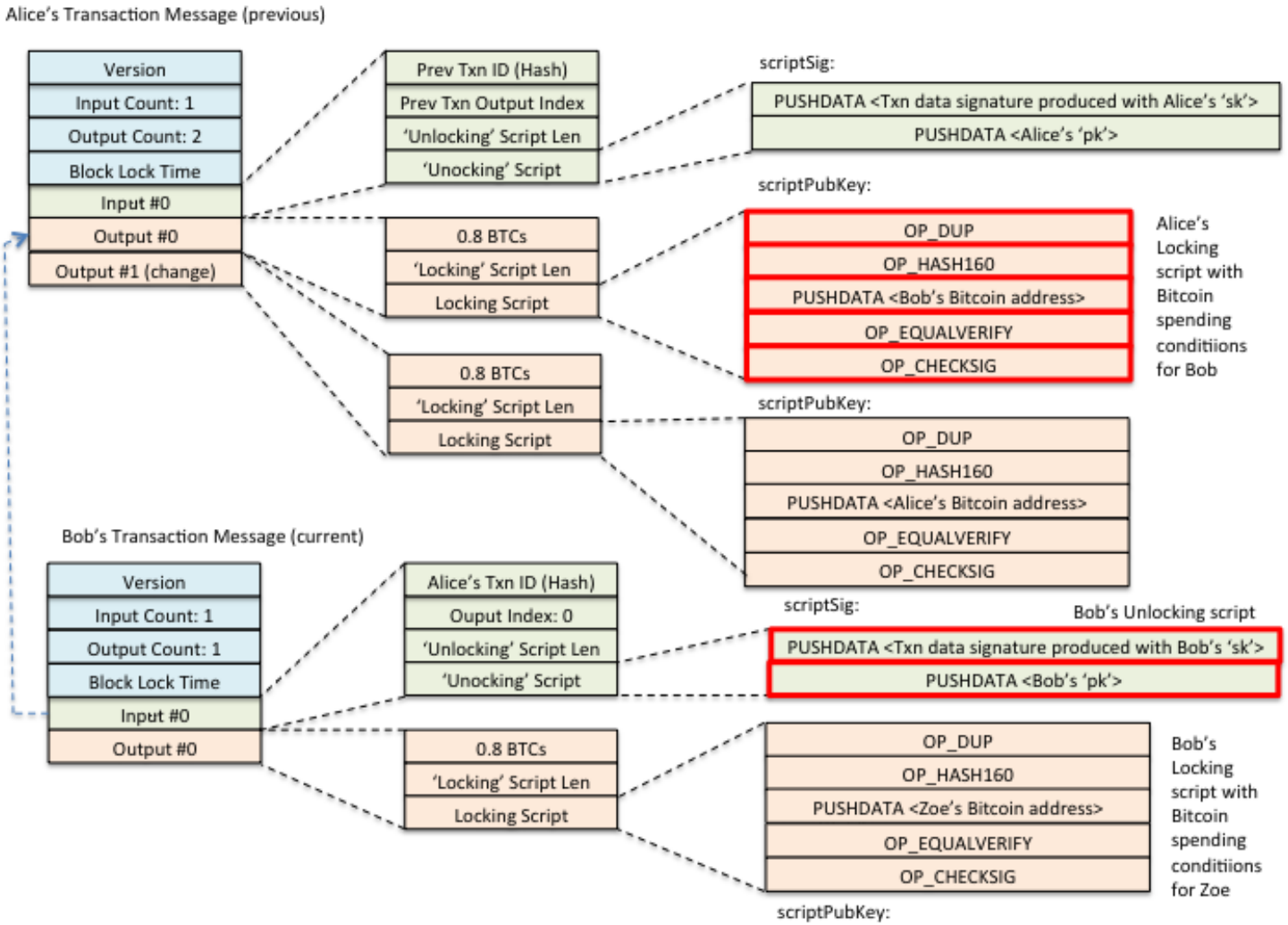
Transaction Script



Transaction Script

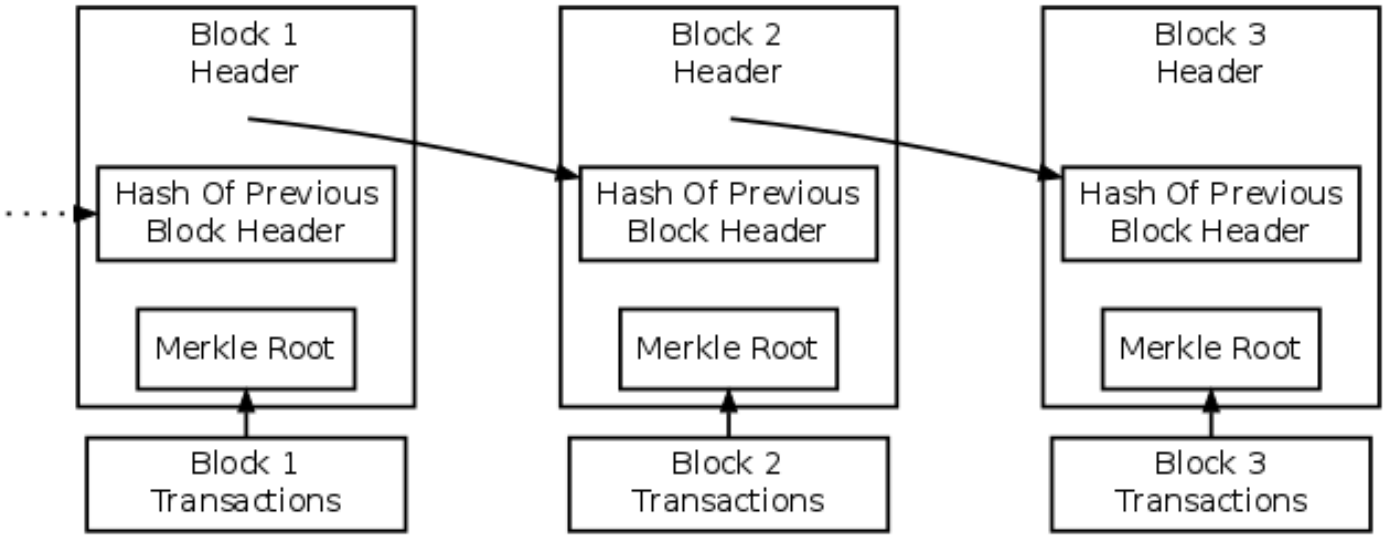


Bitcoin Transaction Verification



BlockChain Overview

- The block chain provides Bitcoin's public ledger, an ordered and timestamped record of transctions.
- This system is used to protect against double spending and modification of previous transactions records.
- Each full node in the Bitcoin network independently stores a block chain containing only blocks validated by that node.



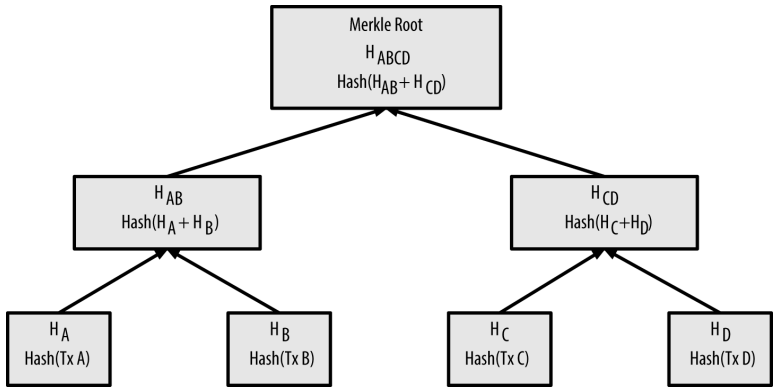
Simplified Bitcoin Block Chain

BlockChain Data Structure

version	02000000
previous block hash (reversed)	17975b97c18ed1f7e255adf297599b55 330edab87803c817010000000000000000
Merkle root (reversed)	8a97295a2747b4f1a0b3948df3990344 c0e19fa6b2b92b3a19c8e6badc141787
timestamp	358b0553
bits	535f0119
nonce	48750833
transaction count	63
coinbase transaction	
transaction	
...	

Block hash

0000000000000000 e067a478024addfe cdc93628978aa52d 91fabd4292982a50



- **Block Header**: 80 bytes, whereas transctions is at least 250 bytes.
- **Nonce**: A counter used for proof-of-work algorithm.
- **Difficulty**: How difficult it is to find a hash below a given target.
- **Coinbase**: The content of the input of a generation transction.

```
1 The Times 03/Jan/2009 Chancellor on brink of second bailout for banks.
```

The Genesis Block

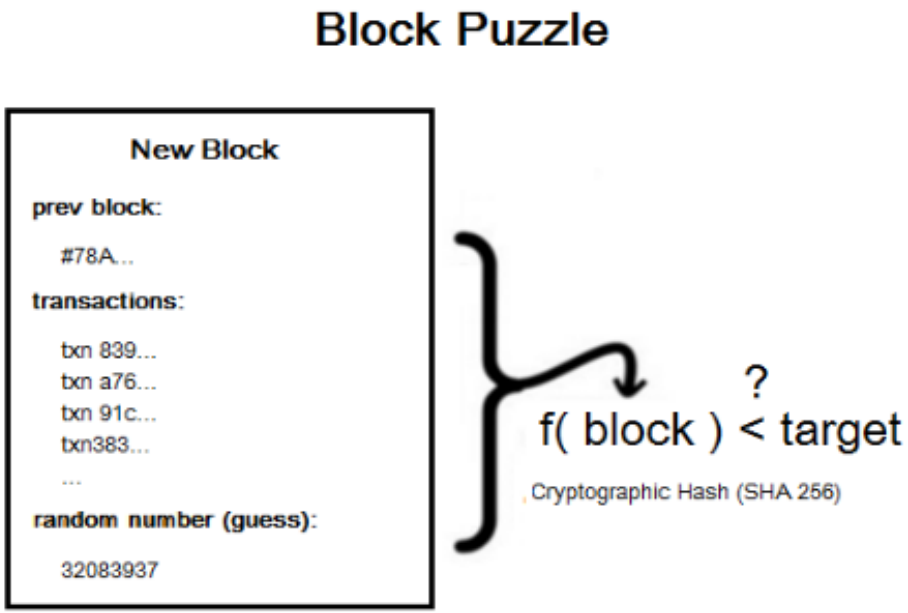
```
bitcoind getblock 000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
{
    "hash" : "000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f",
    "confirmations" : 308321,
    "size" : 285,
    "height" : 0,
    "version" : 1,
    "merkleroot" : "4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b",
    "tx" : [
        "4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b"
    ],
    "time" : 1231006505,
    "nonce" : 2083236893,
    "bits" : "1d00ffff",
    "difficulty" : 1.00000000,
    "nextblockhash" : "00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048"
}

```

The difficulty value updates every 2 weeks to ensure that it takes 10 minutes(on average) to add a new block to the BlockChain.

```
0x00ffff * 2**(8*(0x1d-3))=0x00000000FFFF0000000000000000000000000000000000000000000000000000
```

Proof Of Work

- Block contains transctions to be validated and previous hash value.
- Pick a nonce such that **H(prev hash, nonce, Tx)** $<$ **E**.
- Verification is easy, But proof-of-work is hard.



Difficulty Target and Re-Target

Every 2016 blocks, all node re-target the Proof-of-Work difficulty.

```
New Difficulty = Old Difficulty * (20160 minutes / Actual Time of Last 2016 Blocks)
The Target = The maximum target / current difficulty.
The maximum target is:
    0x00000000FFFF0000000000000000000000000000000000000000000000000000
```

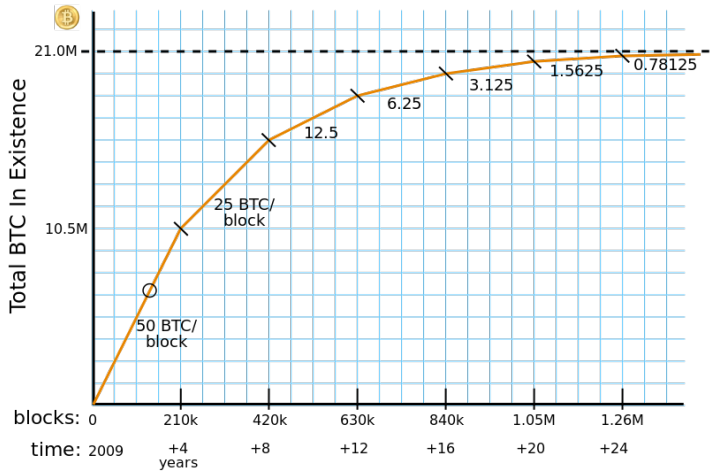
Re-Target Code in bitcoind

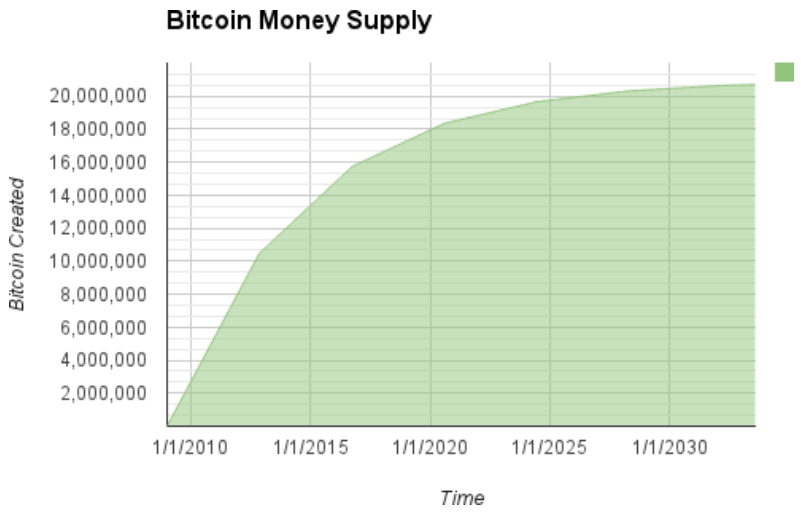
```
    // Limit adjustment step
    int64_t nActualTimespan = pindexLast->GetBlockTime() - nFirstBlockTime;
    LogPrintf("  nActualTimespan = %d  before bounds\n", nActualTimespan);
    if (nActualTimespan < params.nPowTargetTimespan/4)
        nActualTimespan = params.nPowTargetTimespan/4;
    if (nActualTimespan > params.nPowTargetTimespan*4)
        nActualTimespan = params.nPowTargetTimespan*4;

    // Retarget
    const arith_uint256 bnPowLimit = UintToArith256(params.powLimit);
    arith_uint256 bnNew;
    arith_uint256 bnOld;
    bnNew.SetCompact(pindexLast->nBits);
    bnOld = bnNew;
    bnNew *= nActualTimespan;
    bnNew /= params.nPowTargetTimespan;

    if (bnNew > bnPowLimit)
        bnNew = bnPowLimit;
```

Mining and Consensus

Definition

Mining secures the bitcoin system and enables the emergence of network-wide consensus without a central authority. The reward of newly mined coins and transaction fees is an incentive scheme that aligns the actions of miners with the security of the network, while simultaneously implementing the monetary supply.





Decentralize Consensus

When there is no central authority, how can everyone in the network realize consensus?

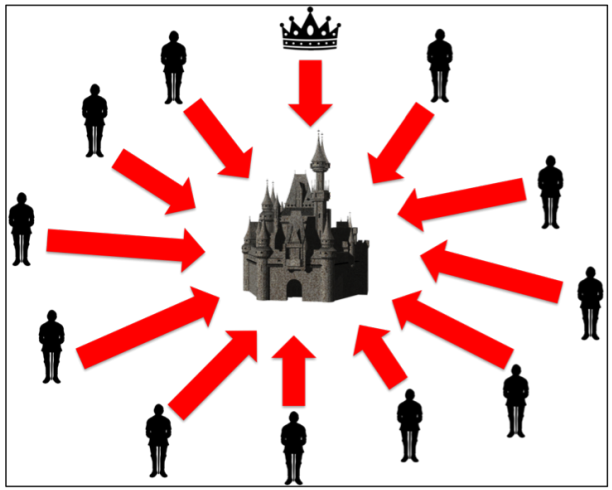
- **tradition consensus algorithm**
  - Paxos, Raft, ...
  - Failure model: Not Byzantine(Node may fail, But not malicious)
  - Leader election.
  - Focusing on log(database), more universe
- **blockchain consensus algorithm**
  - PoW, PoS(Proof of Stake), ...
  - Failure model: Byzantine(Node may be malicious)
  - No election.
  - Focusing on transaction.

Anyway, all the consensus algorithm comply with the majority rule.

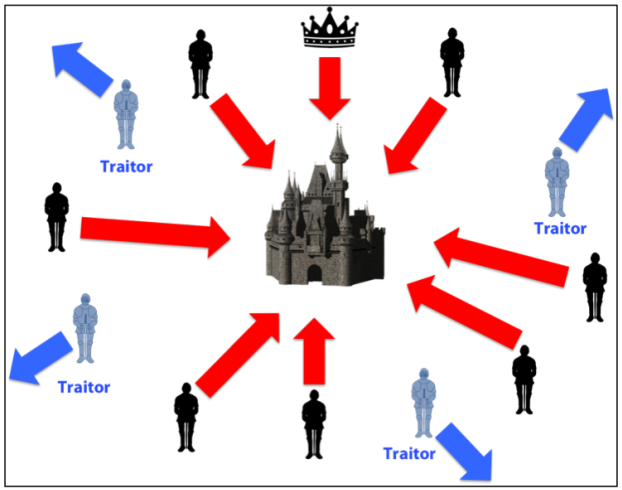
Byzantine Generals' Problem

Story background:

- Generals of army surround enemy city
- Actions in union required to win
- Some generals may be traitors
- Messengers are reliable



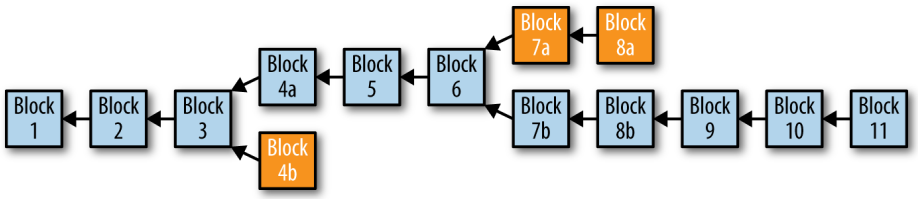
Coordinated Attack Leading to Victory



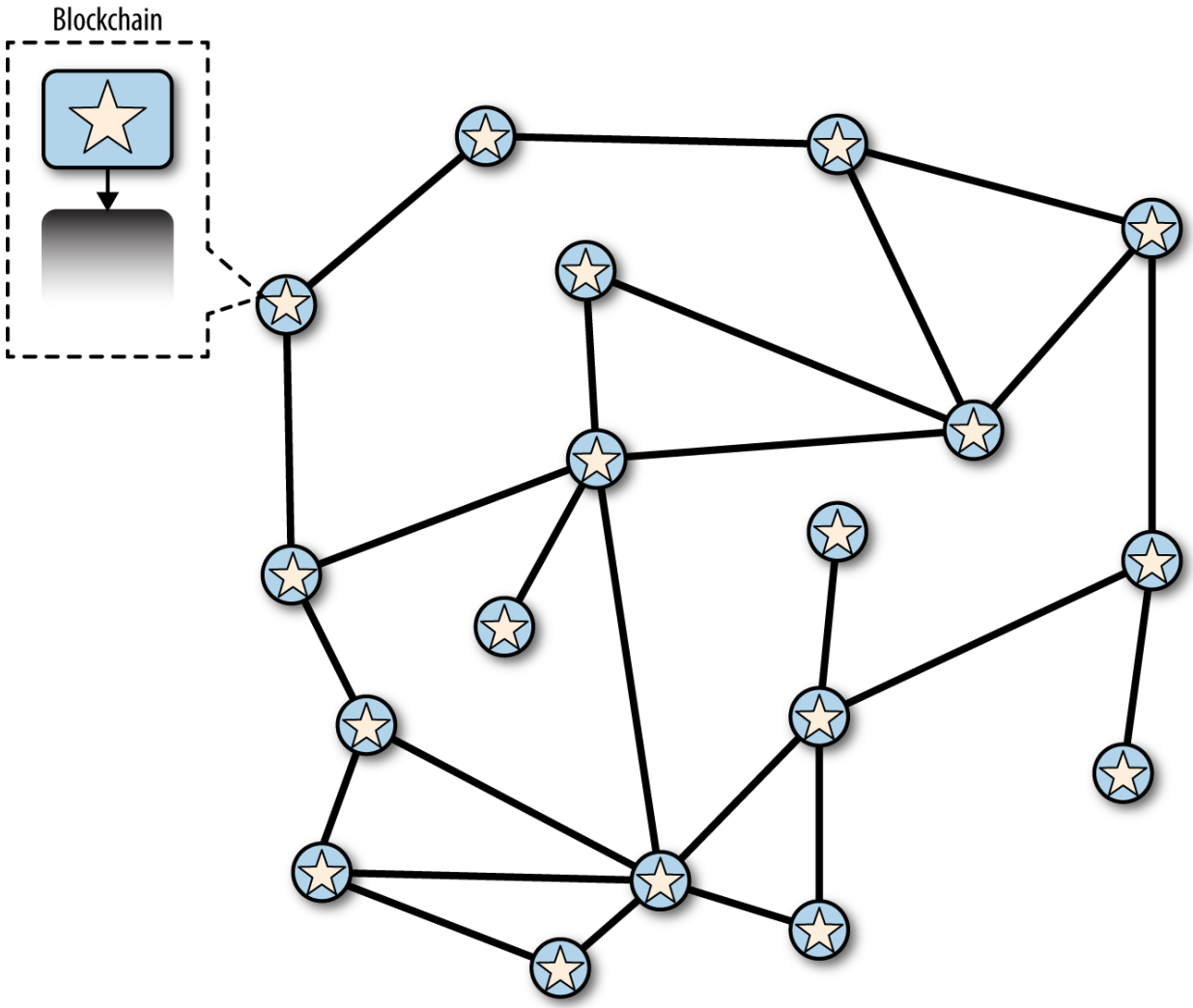
Uncoordinated Attack Leading to Defeat

Proof of Work

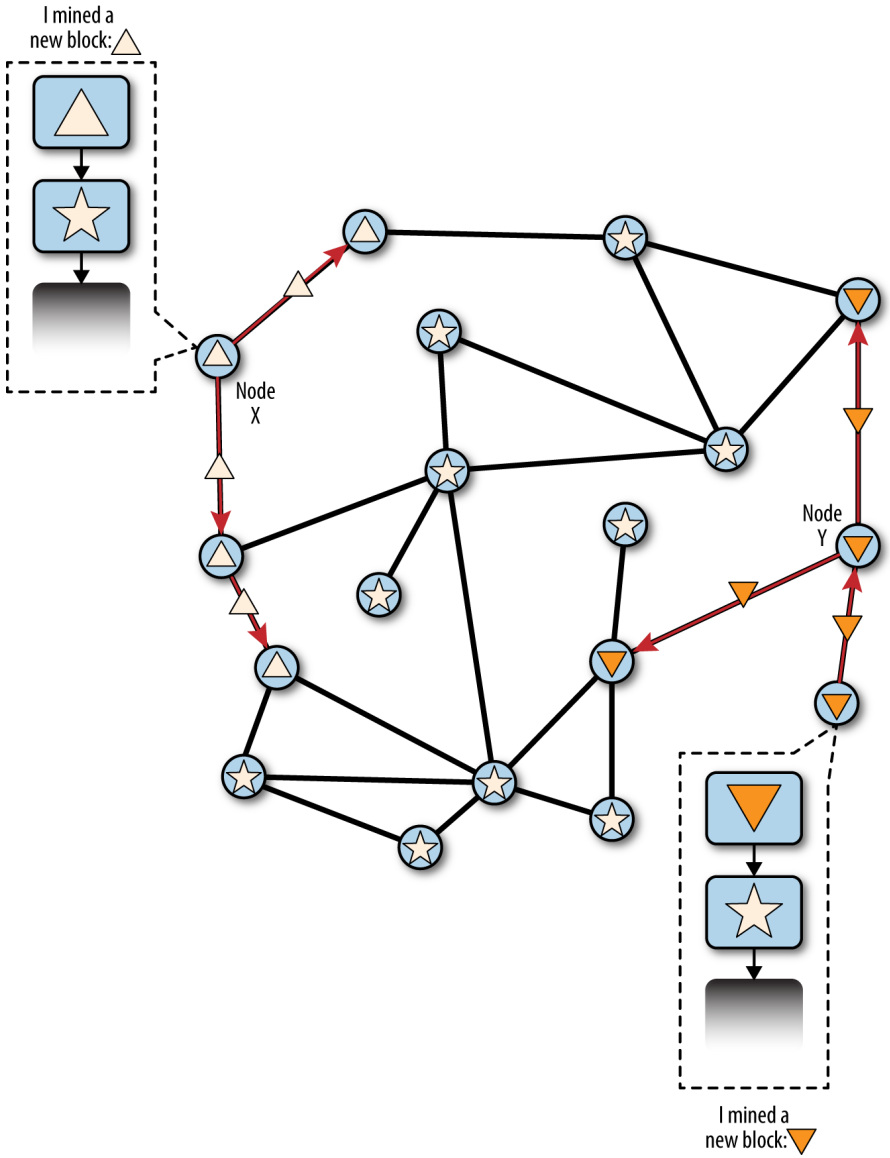
- **Fork**
  - when 2 miners mine a block at the same time.
- **Orphan**
  - Only one can be in the chain, the other is called Orphan
- **Fork/Dispute Resolution**
  - The longest block chain is valid.



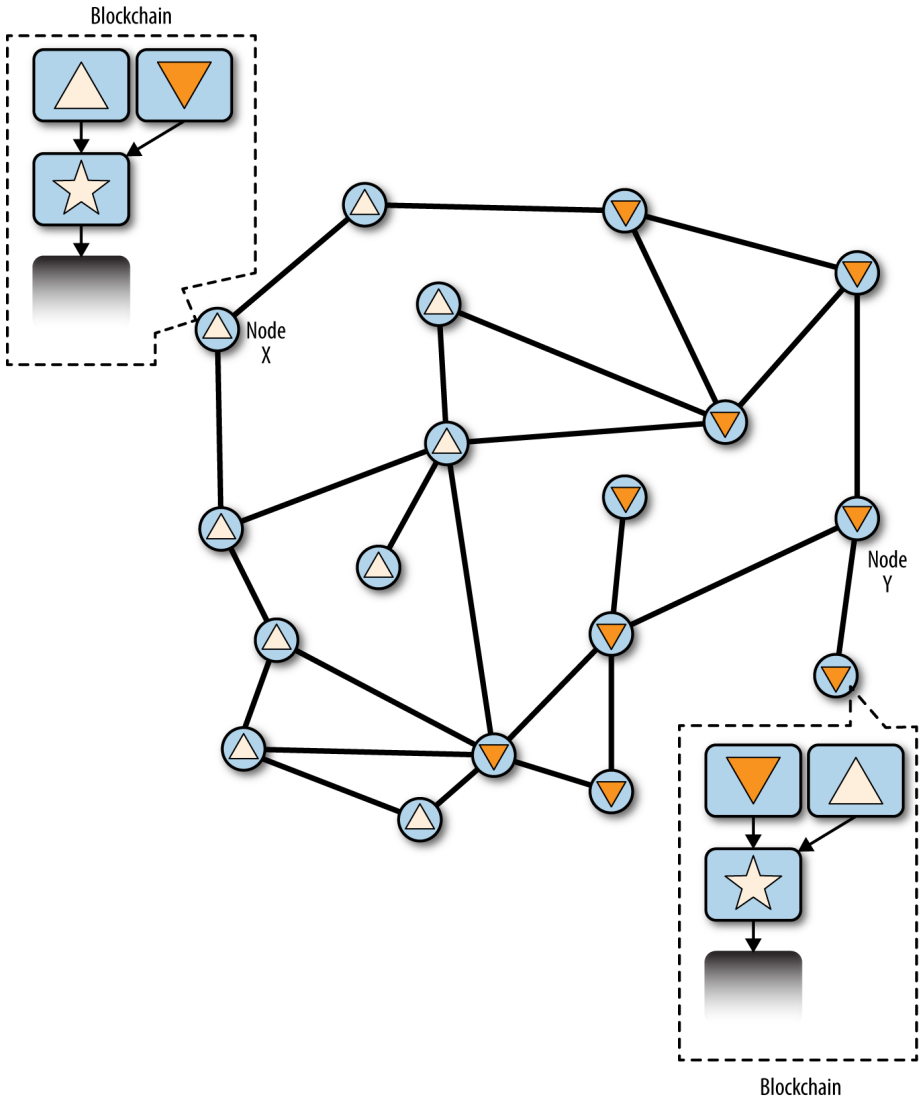
Block Chain Fork



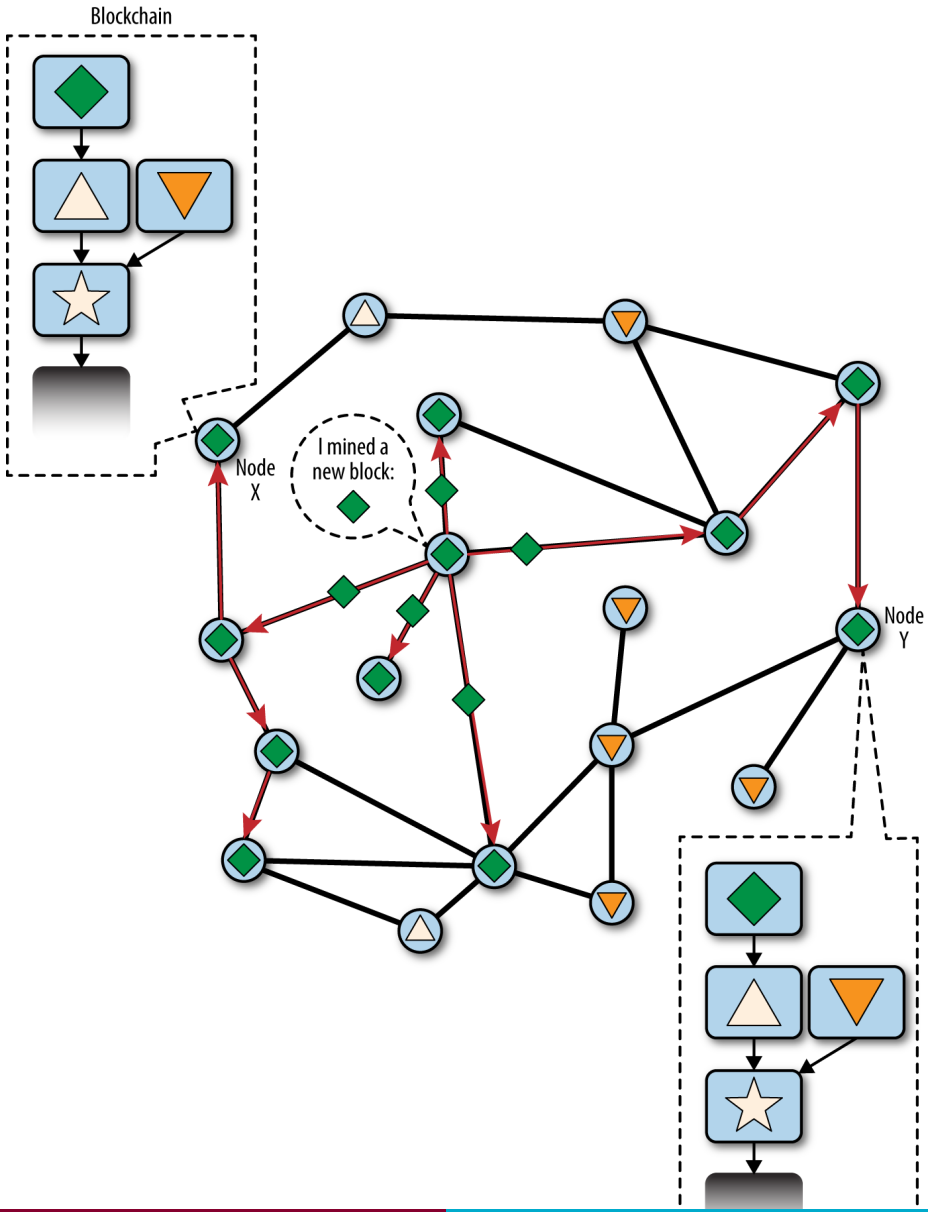
Block Chain Fork



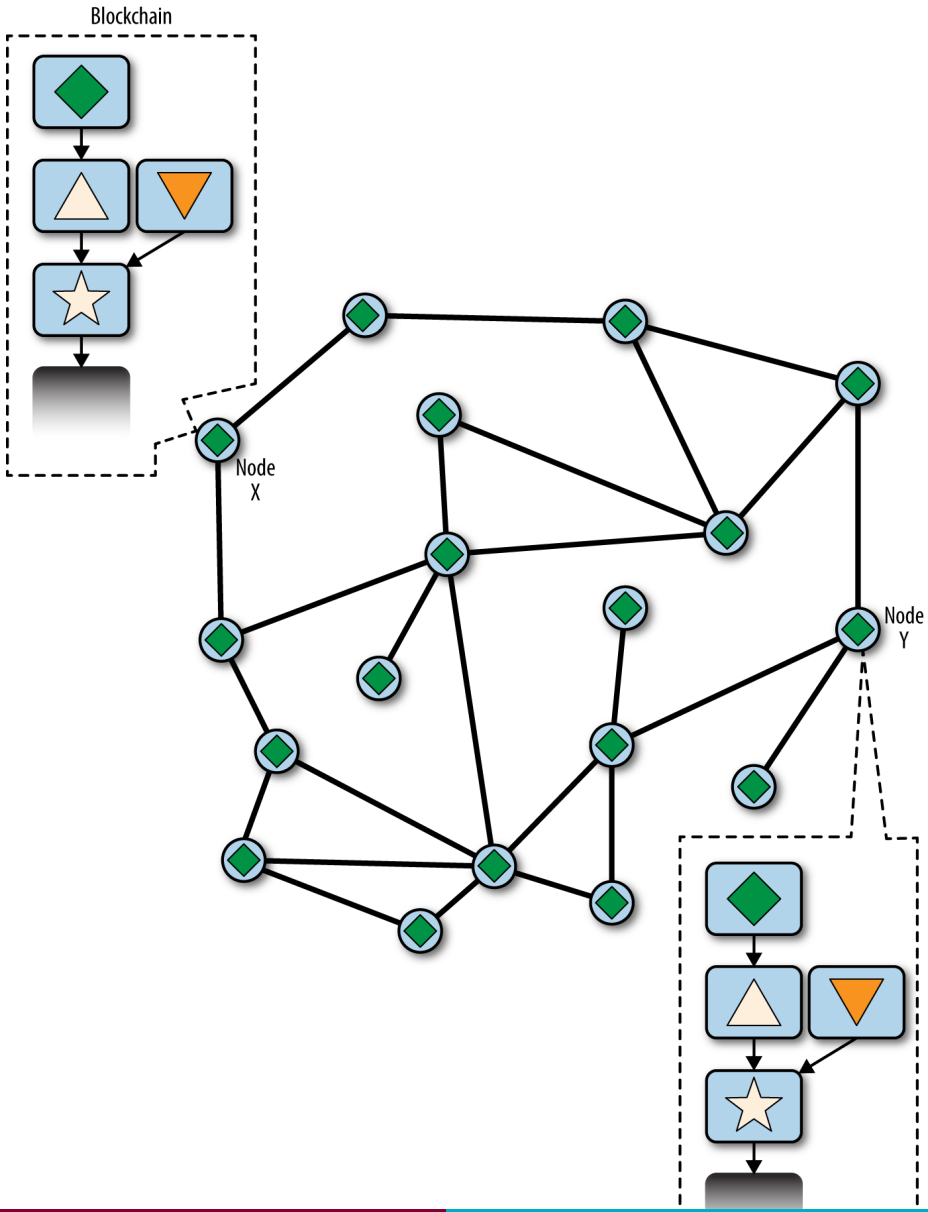
Block Chain Fork



Block Chain Fork

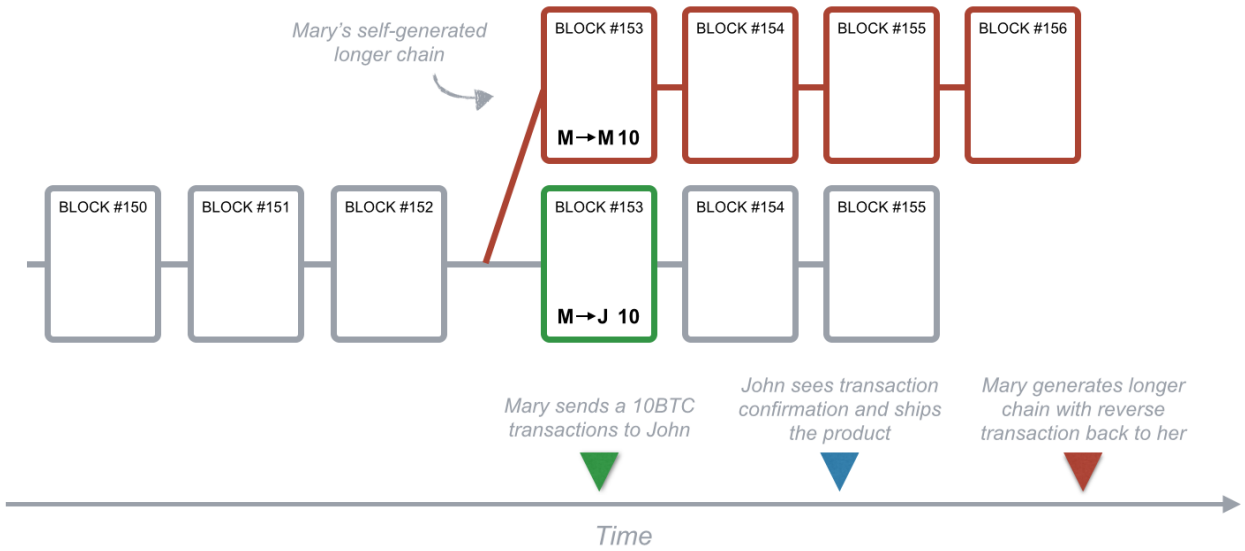


Block Chain Fork



Double Spending Attack

Any attacker is competing against the whole network.

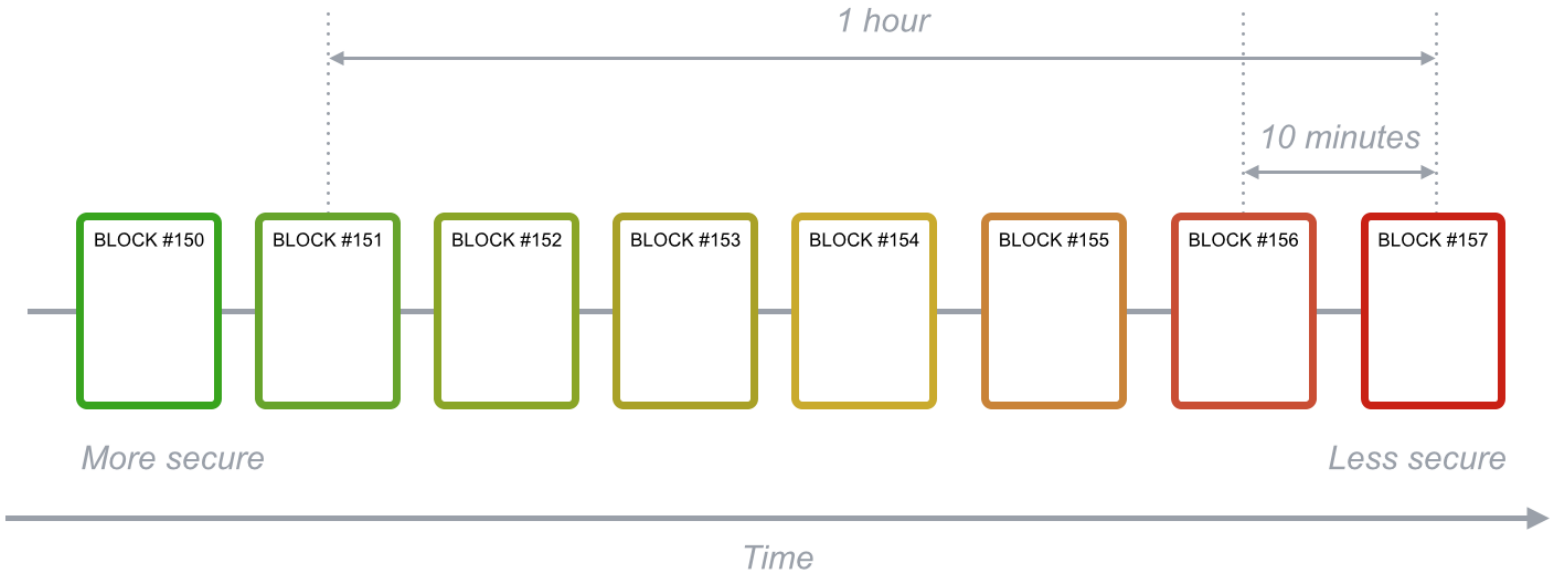


- Each block contains a reference to the previous block.
- 51% computing power of the whole network have a 50% chance to solve a block before some other node does.
- Create 2, 3 or more blocks in a row is even harder.

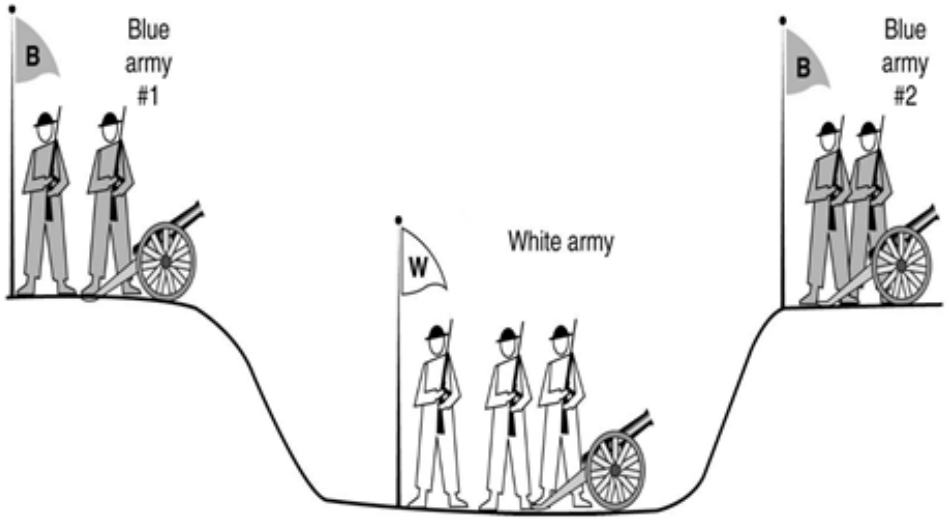
Blockchain Transactions Security

Transactions get more and more secure with time.

- A block is add to the chain every 10 minutes on average.
- Waiting for 6 block(1 hour) gives a quite high probability that the transaction has been processed and is non reversible.

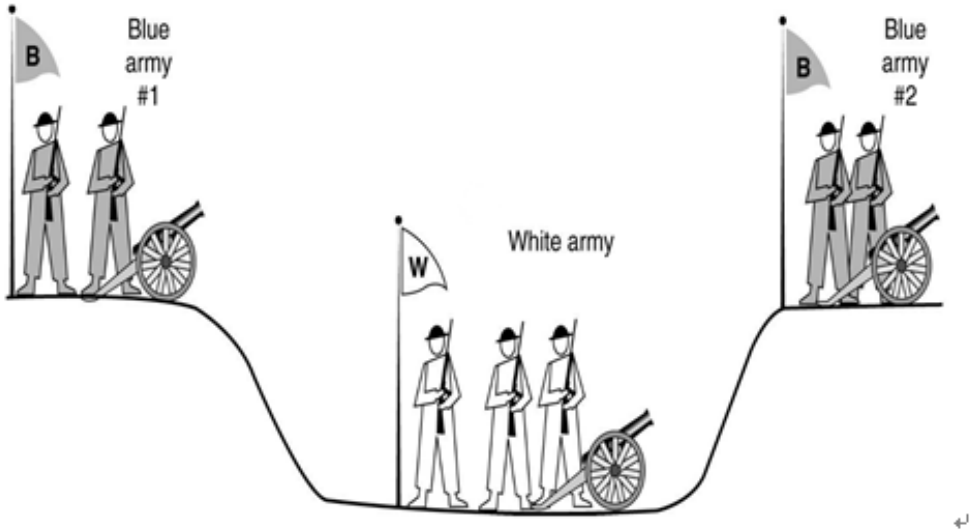


Two Generals' Problem



- Two generals need to coordinate an attack.
  - Must agree on time to attack.
  - They will win only if they attack simultaneously.
  - Communicate through messengers.
  - Messengers may be killed on their way.

Two Generals' Problem

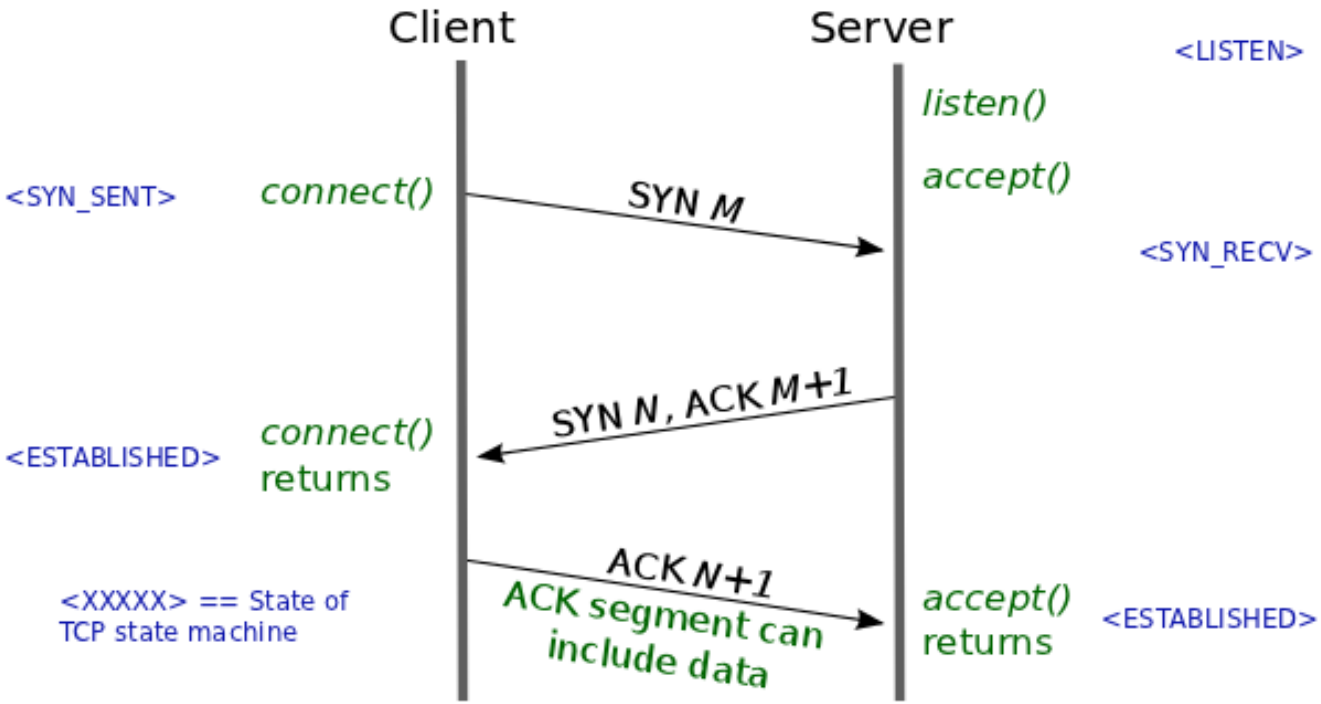


- Let's try to solve it for general g1 and g2.
- g1 sends time of attack to g2.
  - Problem: how to ensure g2 received message?
  - Solution: let g2 ack receipt of message.
  - Problem: how to ensure g1 received ack?
  - Solution: let g1 ack the receipt of the ack.
  - ...
- This problem is impossible to solve!

Engineering approaches: TCP handshake

A progmatic approach to dealing with the Two Generals's Problem

- Accept the uncertainty of the Communication channel.
- Not attempt to eliminate it, but mitigate it to an acceptable degree.
- Classic Example is TCP handshake protocol.



Q & A