



Høgskolen
på Vestlandet

DAT159

Module3 – Blockchain technology

L15 - Distributed Consensus

Lars-Petter Helland, 09.10.2018



Today

- › Distributed consensus
- › Proof-of-work (+mining)
- › Other consensus protocols

- › Reading material:
 - › **[NA Ch2]** - How Bitcoin Achieves Decentralization (some of the text / examples in this presentation is a direct copy from this book)
 - › **[SN]** - Bitcoin: A Peer-to-Peer Electronic Cash System
 - › **[BCP]** - Basic Primer: Blockchain Consensus Protocol, <https://blockgeeks.com/guides/blockchain-consensus/>





Before we start...

Was anything unclear last time?

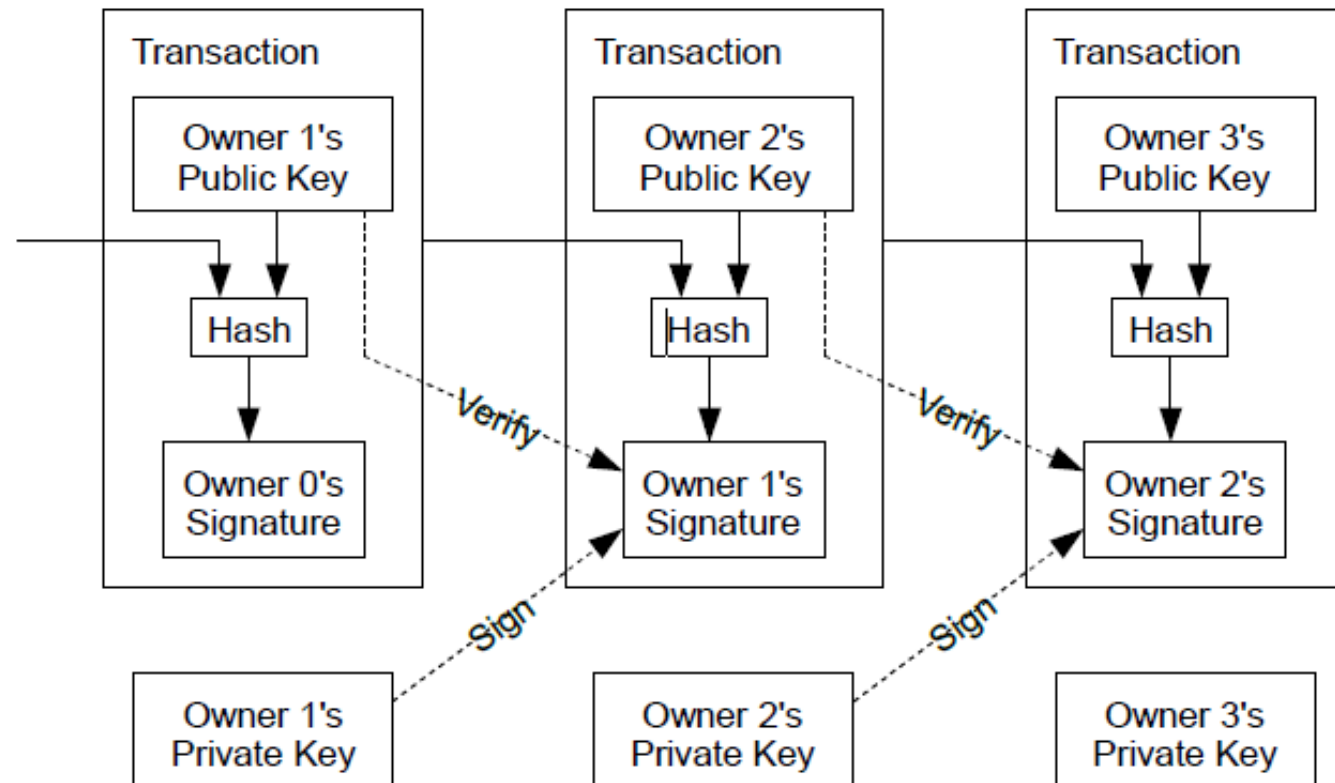
- › Did I leave out something important?
 - › What is...? Why...? How...? Tell me about...?
- › Did some of the things not make sense? Should we repeat something?
 - › Private key, public key, address, hash, signature, bitcoin
 - › The abundance of addresses (2^{160}) / keys (2^{256})
 - › How the Bitcoin-network handles a transaction
 - › Nodes, transaction pool, miner(node)s, mining pools
 - › The purpose of mining (and what it is)
 - › Confirmations and the longest chain
 - › Block, nonce, prev, hashpointer, mining, valid/signed block,
 - › Blockchain, tamper resistance
 - › Transactions, coinbase transaction



Bitcoin white paper - Transactions - Let's try again

The explanation below by Nakamoto is quite simplified, so I got a little bit confused, but ...

"Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership."



Help with exercise

- › How to create a SHA256 hash of a String
- › How to represent a byte[] as a binary String
- › How to represent a byte[] as a hexadecimal String
- › How to represent a byte[] as a base64 String

- › Mining difficulty and the number of leading zeros. What representation should we look at?



Question about the exam

- › I got a question about the exam yesterday.
- › After the lecture, the teachers discussed how to do it.
- › Probably, we will have two "stations" with two topics/teachers in each station.
- › You will probably get ~25 min per station (=> 10+ per topic), ~50 min in total for the exam.
- › That way, it will not be as hectic for any of us.
- › At the end of each day, the teachers will have a meeting and discuss the grades for that day.
- › We will probably be finished in three days, Tuesday-Thursday



Spending the money from last lecture

› <https://live.blockcypher.com/btc/address/1NBdMFdpnd65ACKghrDq9bgRXDeLaHexdD/>

Bitcoin Address



SHARE

1NBdMFdpnd65ACKghrDq9bgRXDeLaHexdD

Private Key

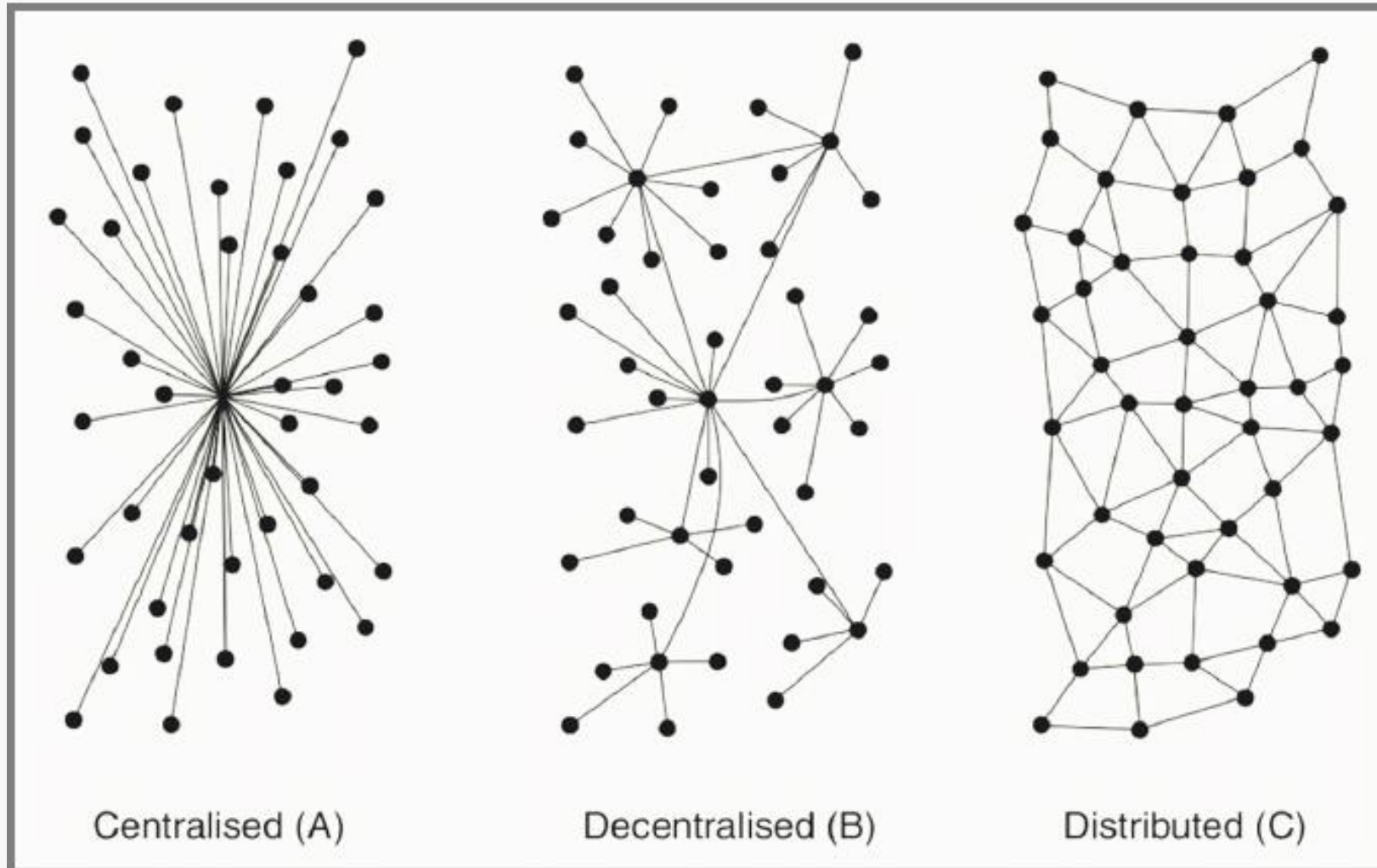


SECRET

L4BrM5MBhzCqGdhAY98EBAQwNJdY42P5hivVUCsqkryHURcWHqs2



Centralization versus Decentralization



Why do we want decentralized / **distributed systems**?

- › Less dependence of middle-men (like banks)
- › Less power to centralized authorities (like banks, governments)
- › Democracy / Anarchy (taking control over ones own life)
- › Owning ones own data (in centralized models, your data is not yours. Facebook, Google, Advertising, Health Info, ...)
- › Privacy (doing business without Little/Big Brother watching)
- › Transacting freely peer-to-peer (not needing permission or paying fees)
- › Efficiency (open global standards)
- › More robust (reliable) systems
- › ...



The problem with distributed systems

- › Reaching consensus!
- › We have copies of data on many different places, and we have to agree on what are the "true" data.
- › **Distributed consensus** has various applications, and it has been studied for decades in computer science. The traditional motivating application is reliability in distributed systems.
- › Let's start with a classical problem.



The Byzantine General's Problem

<https://youtu.be/SF362xxcfdk?t=252>



Properties of a Distributed Consensus Protocol

- › There are n nodes that each have an input value. Some of these nodes are faulty or malicious. A distributed consensus protocol has the following two properties:
 - › **It must terminate with all honest nodes in agreement on the value.**
 - › **The value must have been generated by an honest node.**



Are there solutions to this problem?

Much research has been done on distributed consensus,
... and many **impossibility results** have been proven (see [https://en.wikipedia.org/wiki/Consensus_\(computer_science\)](https://en.wikipedia.org/wiki/Consensus_(computer_science)) for an overview)

One famous impossibility result concerns the **Byzantine General's Problem**. It has been proven that consensus is impossible to achieve if **one-third** or more of the generals are traitors.

A much more subtle impossibility result, known by the names of the authors who first proved it, is **the Fischer-Lynch-Paterson impossibility result**. Under some conditions, which include the nodes acting in a deterministic manner, they proved that consensus is impossible with even **a single** faulty process.



Bitcoin consensus

Ironically, with the current state of research, consensus in **Bitcoin works better in practice than in theory.**

What are the assumptions in traditional models for consensus that Bitcoin violates?

First, it introduces the idea of **incentives**, which is novel for a distributed consensus protocol. **This is only possible in Bitcoin because it is a currency** and therefore has a natural mechanism to incentivize participants to act honestly.

Second, Bitcoin embraces the notion of **randomness** (lottery/mining). .. / probability / convergence.



Pause



Implicit Consensus

- › Bitcoin uses implicit consensus. There is no "consensus algorithm" as such for selecting the block, and no voting of any kind.
- › The algorithm is simplified in that it **assumes the ability to select a random node** in a manner that is not vulnerable to Sybil attacks.
 1. New transactions are broadcast to all nodes.
 2. Each node collects new transactions into a block.
 3. In each round, a random node gets to broadcast its block.
 4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures).
 5. Nodes express their acceptance of the block by including its hash in the next block they create.



Can this work?

Let's now analyze why this consensus algorithm works. To do this, consider how a malicious adversary—call her Alice—may be able to subvert this process.



Stealing bitcoins

Can Alice simply steal bitcoins belonging to another user at an address she doesn't control? **No.**

Even if it is Alice's turn to propose the next block in the chain, she cannot steal other users' bitcoins.

Doing so would require Alice to create a valid transaction that spends that coin. This would require Alice to forge the owners' signatures, which she cannot do if a secure digital signature scheme is used.



Denial-of-Service Attack

Let's consider another attack.

Suppose that Alice really dislikes some other user Bob. Alice can then decide that she will not include any transactions originating from Bob's address in any block that she proposes to put in the block chain.

Even though this is a valid attack that Alice can try to mount, luckily it's nothing more than a minor annoyance.

If Bob's transaction doesn't make it into the next block that Alice proposes, he will just wait until an honest node has the chance to propose a block, and then his transaction will get into that block.



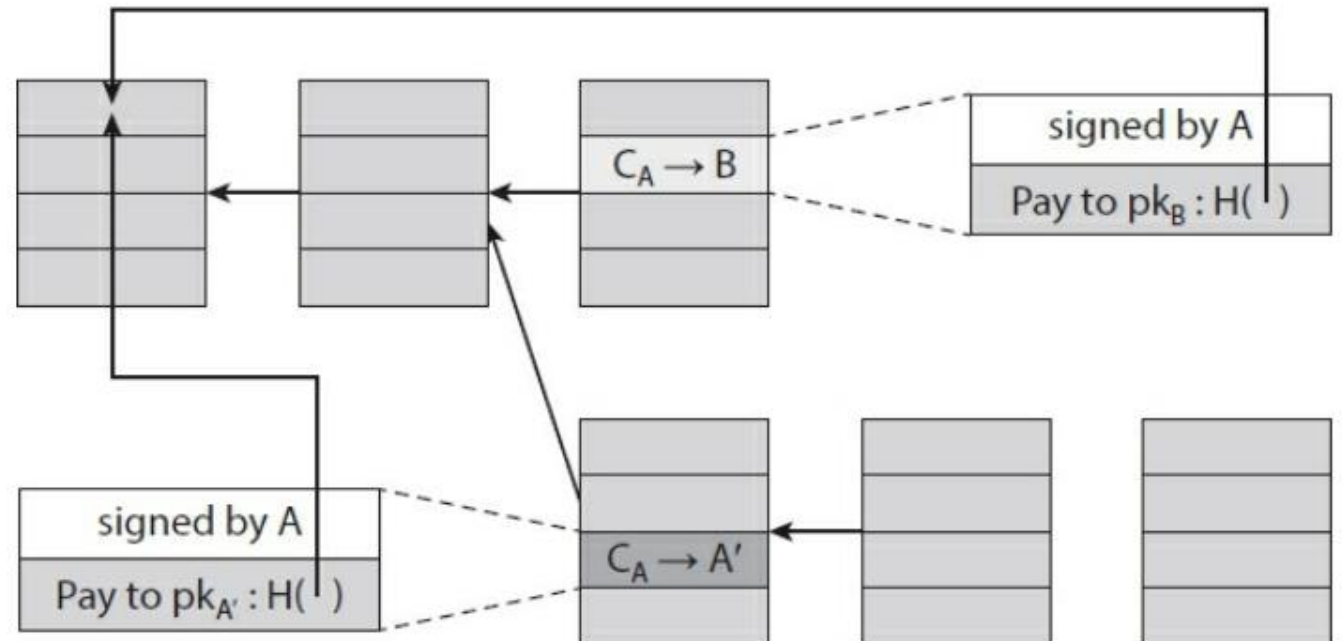
Double-spend Attack

Alice may try to launch a double-spend attack.

A double-spend attack is when you make two transactions that spend the same money, where one of the transactions sends the money back to yourself (at possibly another address you control).

By tricking the recipient to believe that he has received the money, you can get the stuff that you bought.

If Alice gets to propose the next block, she can actively "overwrite" the previous block.



Is the Double-spend Attack going to succeed?

What determines which block will be included? Honest nodes follow the policy of extending the longest valid branch, so which branch will they extend? There is no right answer!

At this point, the two branches are the same length—they only differ in the last block, and both of these blocks are valid.

The node that chooses the next block then may decide to build on either one of them, and this choice will largely determine whether the double-spend attack succeeds.

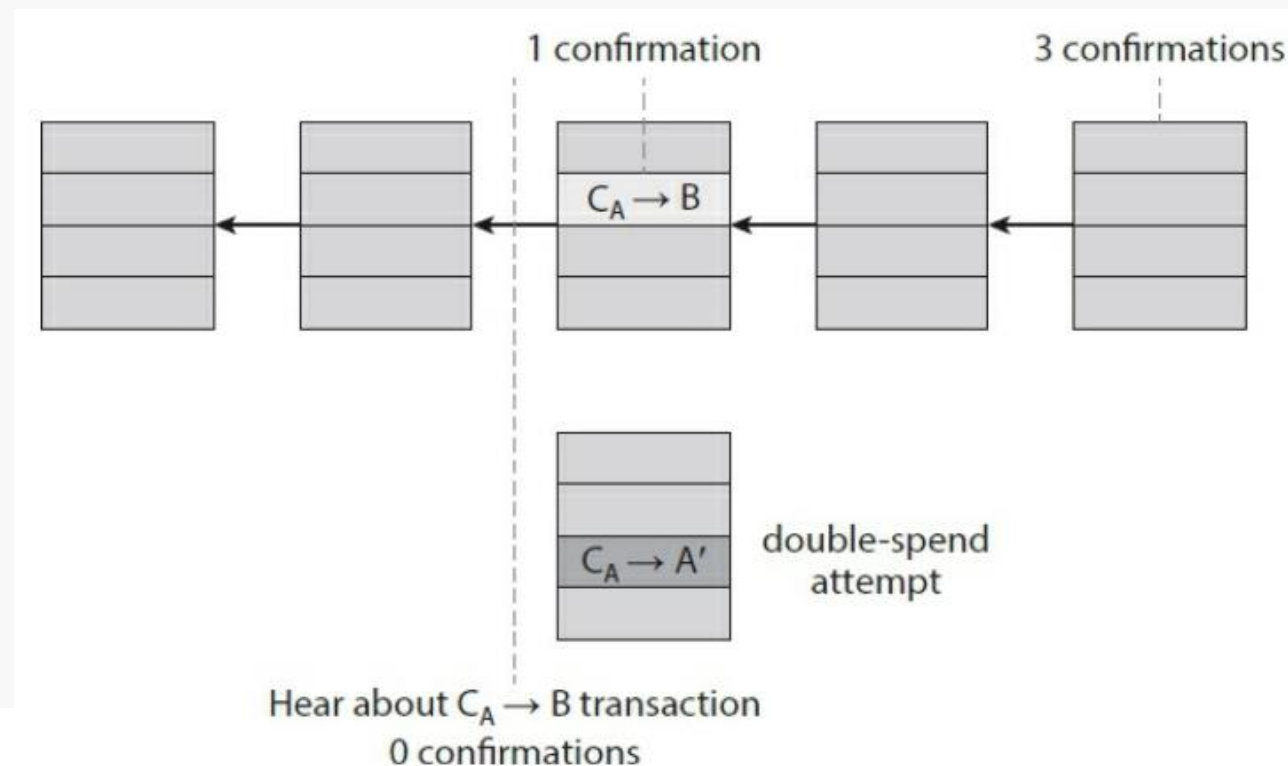


Is the Double-spend Attack going to succeed?

Could the merchant have done anything?

Yes. Waited for more confirmations.

In fact, the double-spend probability decreases exponentially with the number of confirmations.



Incentives and Proof-of-Work

Can we pick a random node and, perhaps more problematically, that at least 50 percent of the time, this process will pick an honest node. This assumption of honesty is particularly problematic if there are financial incentives for participants to subvert the process, in which case we can't really assume that a node will be honest. The question then becomes: **Can we give nodes an incentive for behaving honestly?**

Can we reward each of the nodes that created the blocks that did end up on the long-term consensus chain?

We will look at incentive mechanisms used in Bitcoin:

1. **Block Reward**
2. **Transaction Fees**



Block Reward

According to the rules of Bitcoin, the node that creates a block gets to include a special transaction in that block.

This transaction is a coin-creation transaction (called a coinbase transaction) and the node can also choose the recipient address of this transaction.

The block reward is now 12.5 bitcoin. This means that every 10 minutes,

600.000 NOK is paid to a miner.

Every day

100 mill NOK is paid to miners.

The reward is paid ONLY if the block ends up on the consensus chain.

Year	Blocks	Bitcoins per Block
2009	-	-
2012	210000	50
2016	210000	25
2019	210000	12.5
2023	210000	6.25
2026	210000	3.125
2030	210000	1.5625
2033	210000	0.78125
2037	210000	0.390625
2040	210000	0.1953125



Transaction Fees

The second incentive mechanism is the transaction fee.

The creator of any transaction can choose to make the total value of the transaction outputs less than the total value of its inputs.

Whoever creates the block that first puts that transaction into the block chain gets to collect the difference, which acts as a transaction fee.

The transaction fee is purely voluntary, but it is customary to include a small fee. The miner can choose to omit transactions without fees, and zero-fee transactions also can open for spamming.

We expect the transaction fee to be more important as the block reward decreases.



Mining and Proof of Work

So, how do we "choose" which node that can propose the next block?

The solution is called **Proof-of-work**.

The key idea behind proof of work is that we approximate the selection of a random node by instead selecting nodes in proportion to their computing power.

This can be thought of as **allowing nodes to compete with one another** by using their computing power, which will result in nodes automatically being picked in proportion to that capacity.



Mining and Proof of Work

Bitcoin achieves proof of work using **hash puzzles**. To create a block, the node that proposes that block is required to **find a number (a nonce)** matching a relatively small target space. This is called **mining**.

By this means, the system is completely decentralized. Nobody is deciding which node gets to propose the next block.



Properties of hash puzzles

- › Difficult to Compute
 - › Bitcoin is using a puzzle called HashCash
<https://en.wikipedia.org/wiki/Hashcash>
 - › As of August 2017 (block #478608) the bitcoin network has responded to deployments of ever faster hashing hardware by miners by hardening the requirement to first 72 of 256 hash **bits** must be zero.
 - › The global hashrate in the Bitcoin network is now $\sim 10^{18}$ Hashes/second
 $\sim 10^{22}$ Hashes/block.
- › Parameterizable Cost
 - › We want an average time of 10 minutes between blocks.
 - › Recalculation of the target happens roughly every two weeks.
- › Trivial to Verify



Game-theory

In the game-theoretic view, the big question is whether the default miner behavior is a **Nash equilibrium**, that is, whether it represents a stable situation in which no miner can realize a higher payoff by deviating from honest behavior. This question is still contentious and is an active area of research.

On the practical side, you need much resources to carry out a successful attack. An attack may affect the value of the token negatively. What is the motivation of attacking "your own business"?



The 51 Percent Attack

- › To successfully control what blocks that are added to the chain, you need to control the majority of the hashrate.
- › That is why we call it the 51% attack.
- › We have already looked at this, but what can a 51% attacker do?
 - › Steal coins from an existing address? **No**
 - › Suppress transactions? **Well**
 - › Change the block reward? **No**
 - › Double-spend? **Maybe**
 - › Destroy the confidence in Bitcoin? **Yes**



Problems with Mining and Proof of Work (PoW)

- › Cost / energy consumption / environmental impact
 - › **Fact:** Bitcoin mining consumes the same amount of electricity as a small country like **Denmark**
- › ASICs / Centralization / Monopoly
 - › **Fact:** More than **90%** of the overall Bitcoin hash power is owned by less than **20 companies**. One company (**Bitmain**) controls over **50%** of the mining power. (51% attacks?)
- › Vulnerability to attacks
 - › **Fact:** Coins with smaller networks using the same hashing algorithm as Bitcoin can easily be **attacked** by Bitcoin-miners.



Other consensus mechanisms

- › Proof of Stake (PoS)
- › Delegated Proof of Stake (DPoS)
- › Delegated Byzantine Fault Tolerance (dBFT)

- › ... there are some more (Proof Of Activity, Proof Of Burn, Proof Of Elapsed Time, Proof Of Capacity, ...)
- › ... there are also hybrids

- › <https://blockgeeks.com/guides/blockchain-consensus/>


















There are ~2000
other cryptocurrencies

See

<https://coinmarketcap.com/all/views/all/>

, some of these use other consensus
mechanisms.

Cryptocurrencies ▾		Exchanges ▾	Watchlist
#	Name	Symbol	Market Cap
1	 Bitcoin	BTC	\$114 980 323 998
2	 Ethereum	ETH	\$23 453 788 270
3	 XRP	XRP	\$19 337 376 972
4	 Bitcoin Cash	BCH	\$9 052 325 139
5	 EOS	EOS	\$5 354 951 278
6	 Stellar	XLM	\$4 655 758 678
7	 Litecoin	LTC	\$3 461 361 226
8	 Tether	USDT	\$2 791 088 935
9	 Cardano	ADA	\$2 236 548 173
10	 Monero	XMR	\$1 878 778 533
11	 TRON	TRX	\$1 699 603 923
12	 IOTA	MIOTA	\$1 645 704 595
13	 Dash	DASH	\$1 517 267 301
14	 Binance Coin	BNB	\$1 235 067 961
15	 NEO	NEO	\$1 190 010 278



Proof of Stake (PoS)

- › Soon to be used by **Ethereum (#2)**
- › Miners are replaced with **validators**.
 - › The validators will have to lock up some of their coins as **stake**.
 - › After that, they will start validating the blocks. Meaning, when they discover a block which they think can be added to the chain, they will validate it by placing a bet on it.
 - › If the block gets appended, then the validators will get a reward proportionate to their bets.
- › Pros: More resource-friendly
- › Cons: The "Nothing at Stake" problem. You are **not punished** if you are dishonest. (Ethereum is addressing this problem in Casper)



Delegated Proof of Stake (DPoS)

- › Used by **EOS (#5)**
- › The problem with highly distributed consensus like PoW and PoS is that they are slow and don't scale well.
- › A solution to this is Delegated Proof of Stake (DPoS). You **delegate** the job of producing blocks to a small number of nodes through a voting system.
- › Pros: A DPOS blockchain typically has 100% **block producer** participation. A transaction is usually confirmed within 1.5 seconds from the time of broadcast by a 99.9% certainty. (EOS)
- › Cons: A degree of centralization. Game-theoretic issues.



Delegated Byzantine Fault Tolerance (dBFT)

- › Used by **NEO (#15)**
- › ... (read article) <https://blockgeeks.com/guides/blockchain-consensus/>
- › Pros:
- › Cons:



Next

- › Work on the programming assignment
- › Supervision Friday from 8+ - 10.
- › Next week: The mechanics of Bitcoin, inkl. programming

