

The Blockchain Tech



Mihail Mihaylov
AI lab, VUB

Technological innovation

No new technology - a clever combination of existing technologies:

- Distributed Systems
- Peer-to-peer networks
- Hashing function
- Public-Private key **cryptography**
- **Cryptographic** signatures
- Elliptic curve **cryptography**

Which problems does blockchain solve?

- Preventing double spend of digital currency
- Achieving consensus in a decentralized network
- Byzantine Generals' problem:
 - It is not sufficient that everyone knows X.
 - We need everyone to know that everyone knows X,
 - and that everyone knows that everyone knows that everyone knows X...

Byzantine Generals' problem

- Several armies surround an enemy city.
- To win, generals must agree on a common attack time.
- Doesn't matter when, as long as enough troops attack together.
- Communication is insecure.
- One or more generals may be traitors.

Find an algorithm to ensure that the loyal generals will reach agreement. (assuming more than the half are loyal generals)

How to reach decentralized agreement?

Anyone can announce a “plan”. (E.g. *Attack next Monday@9am!*)

1. If you receive a plan, set your attack time to it.
2. Start solving a difficult problem involving that plan.
 - each army's problem is comparable in difficulty to the rest
 - solving depends on army size
 - 10k troops solve it for 1 day (on average).
3. When your army finds a solution, announce it to others.
 - others can instantly verify the solution.
4. When you receive a valid solution, start solving a new problem, involving the (new) plan and the previous solution.

When to attack?

- After several days the problem has been solved multiple times. All solutions form a chain.
- Looking at the chain, each general:
 - can estimate the size of armies busy solving the problems
(10k troops = 1 day)
 - knows that the majority generals must have seen the plan
(chain) \Rightarrow everyone knows that everyone knows that everyone knows X...
- If the amount of armies is enough to win the battle, they will attack according to the plan.

Key concepts

Bitcoin software - open-source implementation of the bitcoin protocol

Peer-to-peer network - decentralized network of nodes running the software

Blockchain - decentralized database of transactions generated by miners

Bitcoin token - decentralized digital currency

Bitcoin address - cryptographic key pair

Bitcoin wallet - app that generates/keeps your keys

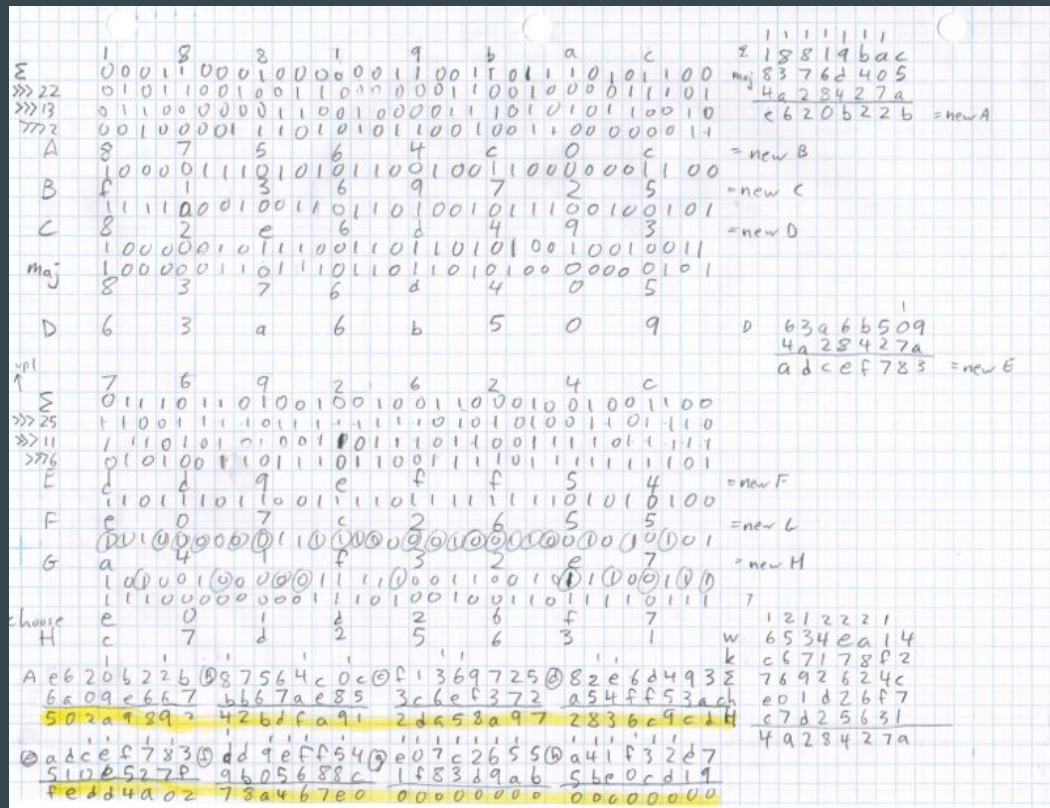
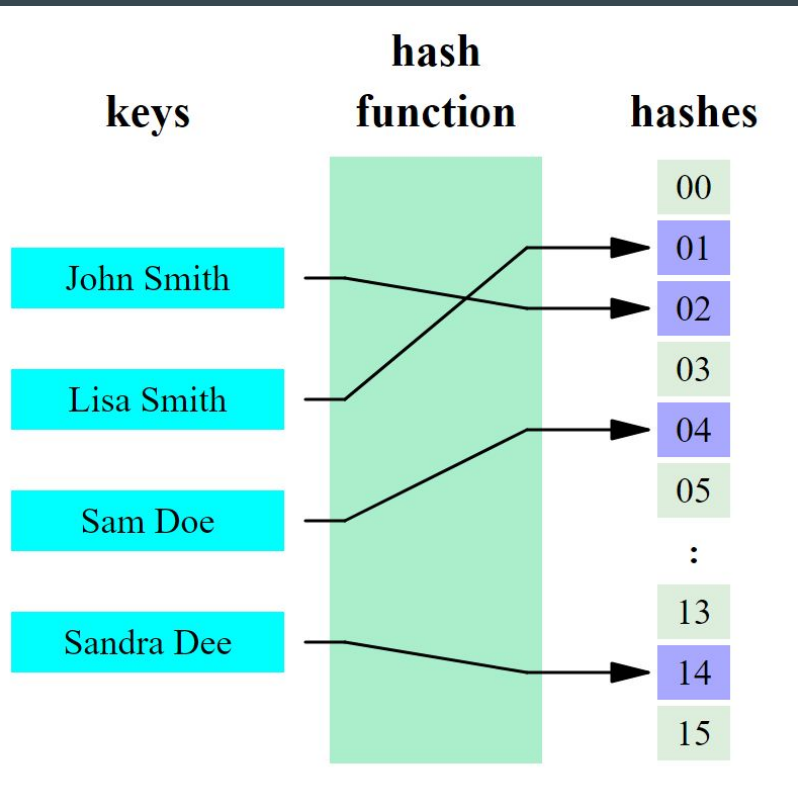
Concepts

- Hash
- Block
- Blockchain
- Mining

Hash

<https://anders.com/blockchain/hash.html>

<http://www.righto.com/2014/09/mining-bitcoin-with-pencil-and-paper.html>



Block

INSIDE BITCOIN'S BLOCKCHAIN

HEADER

The block header is hashed twice to create the fingerprint which is referred to in the next block.

Technical data	Previous block hash
Includes a Magic ID, a version number (to specify which set of protocol rules this block conforms to), the size of this block.	2x SHA256 hash of previous block header (excluding magic ID & block size). This is the link that creates the chain of blocks.
Merkle Root	Timestamp
Distills all the transactions in the block into a single hash.	Approximate timestamp of when the block was created. Used to figure out mining difficulty re-targets i.e. if the network is making blocks too quickly or too slowly.
Difficulty target	Nonce
Related to mining and how hard it is to successfully mine the block	A random number. One of the things you can change when mining to create different hashes, while searching for a suitable hash.

TECHNICAL DATA

Version number	Transaction lock time
Can be used for specifying which set of protocol rules this transaction confirms to.	Something which may be used in future for "future dating" a transaction, like writing a post-dated cheque.
Input count	Output count
How many inputs are in this transaction.	How many outputs does this transaction create.

INPUT

(Technical) Input script length	(Technical) Sequence number
How much data is in the input.	Not really used.
Previous transaction hash & index	Script data
This identifies where the coins are coming from, by specifying an output from a previous transaction.	This is where you "prove" you own the coins and you are allowed to spend it, by signing with the private key of the address that the bitcoins are in.

BLOCK

Blocks are the units of the blockchain, like pages of transactions in a ledger.

HEADER

Technical data	Previous block hash
Merkle Root	Timestamp
Difficulty target	Nonce

TRANSACTION COUNT

How many transactions are in the block, including the coinbase transaction.

BLOCK CONTENT

Coinbase transaction	Bitcoin transactions
----------------------	----------------------

TRANSACTION

Each transaction is a bitcoin payment

Technical Data

Inputs	Outputs
Which coins are being spent?	Who is getting the coins?

INPUT

(Technical) Input script length
(Technical) Sequence number
Previous transaction hash & index
Script data

OUTPUT

(Technical) Output script length
Amount
Output script

BLOCK CONTENT

Coinbase transaction

The bit where you get to pay yourself the mining reward (currently 25 BTC) plus the fees from the transactions included in the block.

It's a special transaction where there are no 'inputs' or 'from' addresses.

Bitcoin transactions

This is the main payload of the block. Contains bitcoin payments.

Transaction	Transaction
Transaction	Transaction

FOLLOWING THE MONEY

Bank accounts vs cryptocurrencies

Bank accounts mix money up. When you pay someone, you don't specify "use those pounds which I earned from my salary" or "use those pounds which I received for my birthday". Money is treated equally once it hits your account, and is untraceable.

On the other hand, with cryptocurrencies, you need to specify exactly which incoming deposits you are spending. This makes every transaction traceable, right back to the creation of the coins.

Inputs and Outputs

Every bitcoin transaction references some incoming deposits as inputs, and spends them entirely as new outputs, with change returned to one of your addresses.

This is like paying £43.50 by taking three £20 banknotes from your wallet and creating two new banknotes: £43.50 and £16.50. You hand over the £43.50 banknote and keep the £16.50 banknote. You can then spend the £16.50 later in one go. The other person can spend the £43.50 later in one go.

Inputs: 3 x £20

Outputs: £43.50 (payment), £16.50 (change)

OUTPUT

(Technical) Output script length	Amount
How much data is in this output	How many bitcoins (actually, Satoshis) are being sent.

Output script

Who (which address/es) are the bitcoins being sent to? Which signatures are needed to re-spend these coins?



www.bitsonblocks.net



Designed by www.fractalphia.com

Blockchain

Broadcasted transactions are gathered in a “block”.

Blocks are “mined” and appended to the blockchain.


Each block “points” to the previous one.

Mined blocks mint new bitcoins for the miner.

Mining

Who gets to write the next block in a decentralized peer-to-peer network?

1. Each miner **collects** broadcasted transactions
2. Attempts to generate a valid block by **hashing**:
 - all collected transactions (Merkle root)
 - the hash of the previous block
 - her public key
 - a **nonce**until the hash is smaller than **X**, where **X** is the Difficulty
3. **Broadcasts** valid block when found
 - mints Bitcoins as reward
4. Everyone **includes** this block in the longest known chain so far
 - longest chain = chain with the most total combined difficulty (work)



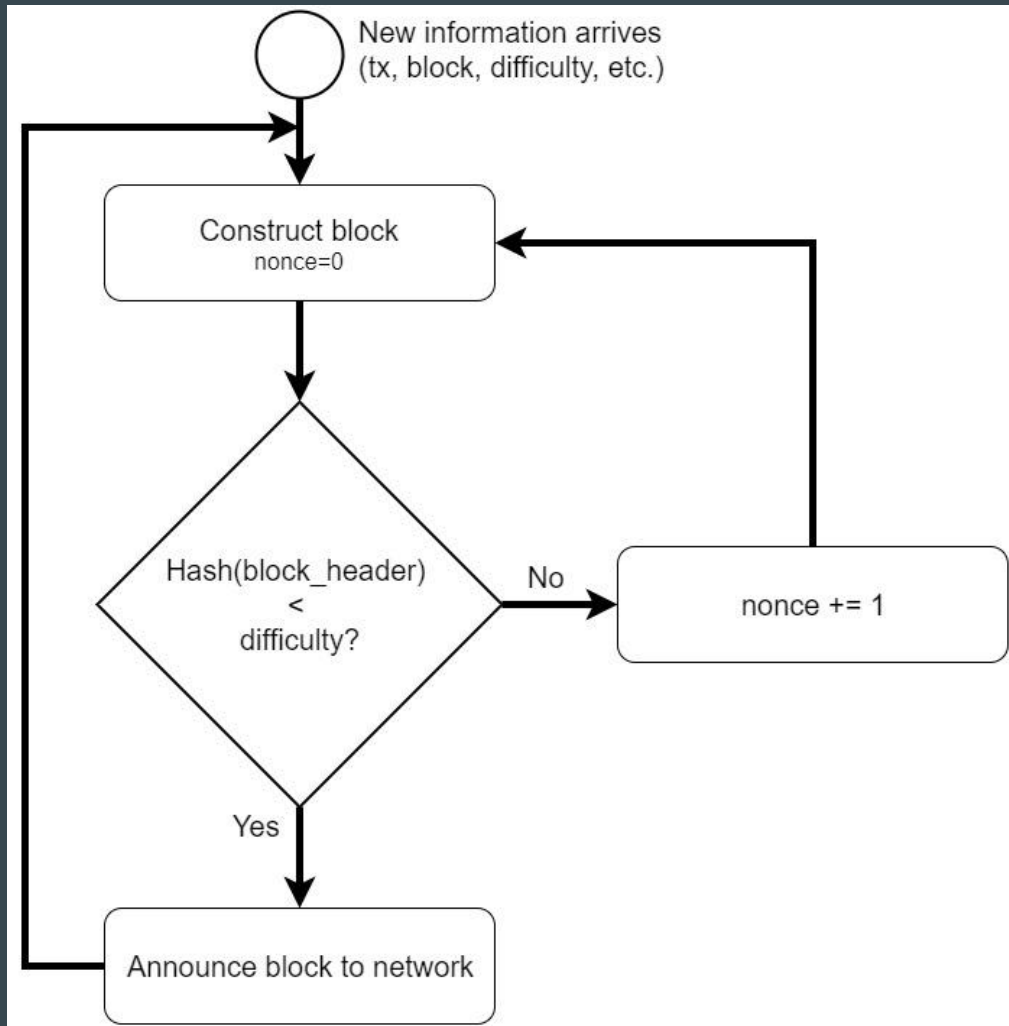
Proof-of-work
mining

Included transactions have now 1 confirmation.

Proof of Work

<https://anders.com/blockchain/block.html>

<https://anders.com/blockchain/coinbase.html>



Summary

A **peer-to-peer network** of nodes running the **Bitcoin software** and protocol

Transfer ownership of Bitcoins using **public/private key pairs**

Transactions are recorded in the **blockchain** decentralized public ledger

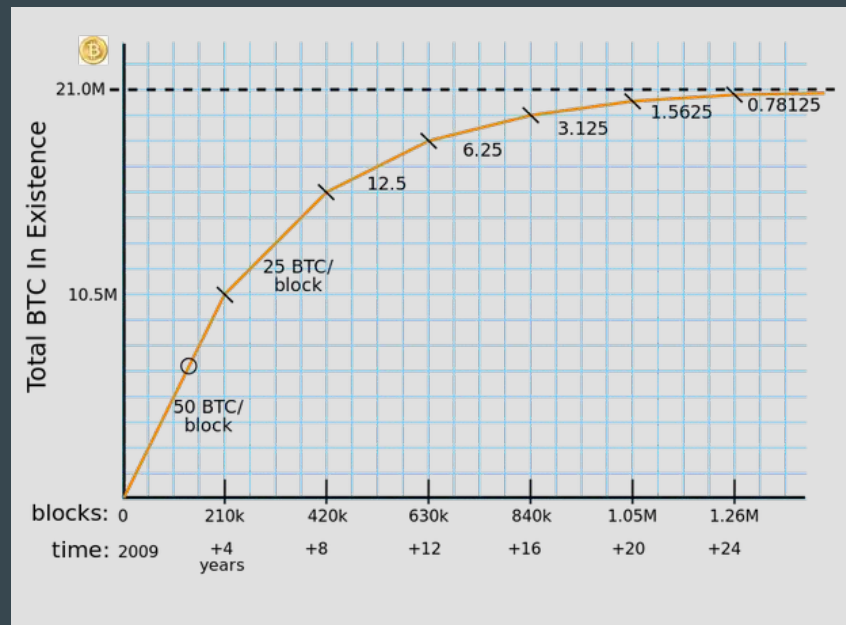
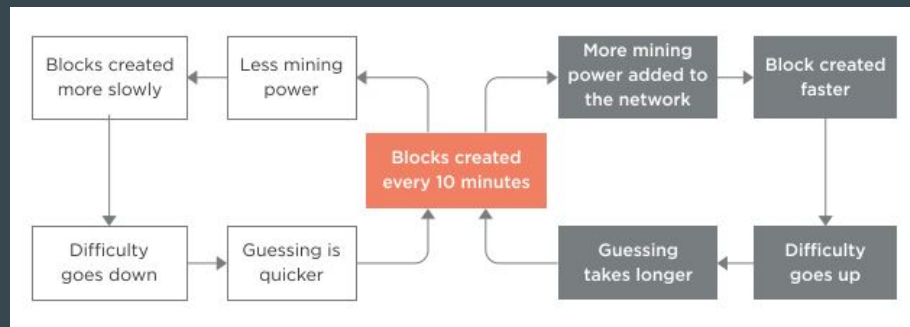
Miners secure the ledger and generate **Bitcoin tokens** as a reward

<https://www.coursera.org/learn/cryptocurrency>

Parameters

- Difficulty adjusts every 2 weeks
 - such that blocks are found in 10 mins on average.
- Rewards halve every 4 years.
 - mimics rate at which gold is dug from the earth.
- Granularity: 8 decimal places
- Max 21 Million bitcoins to ever exist
 - all to be mined by year 2140
- etc...

<https://blockchain.info/>



Practical

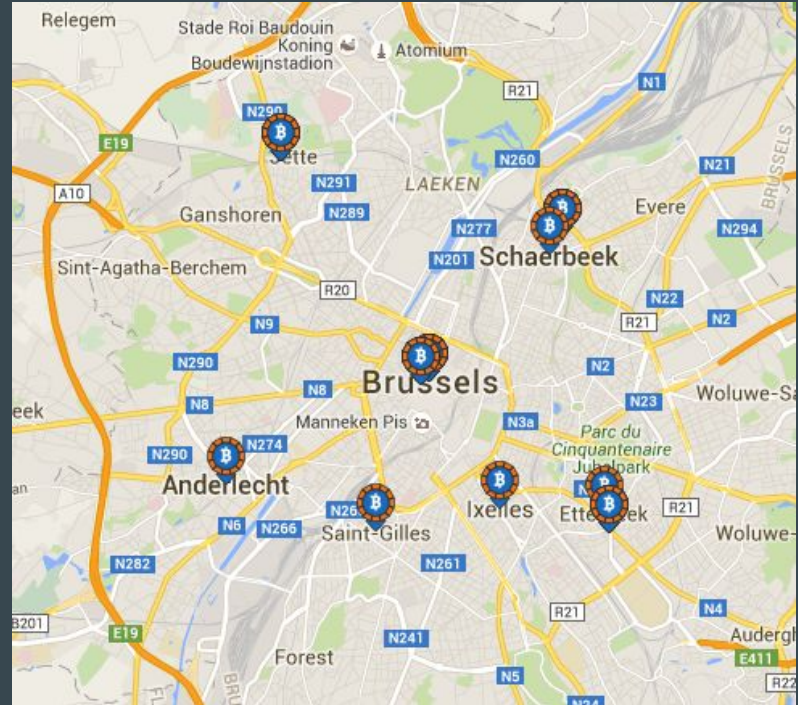
How do I get bitcoins?



Buy bitcoins on exchanges



bisq.network

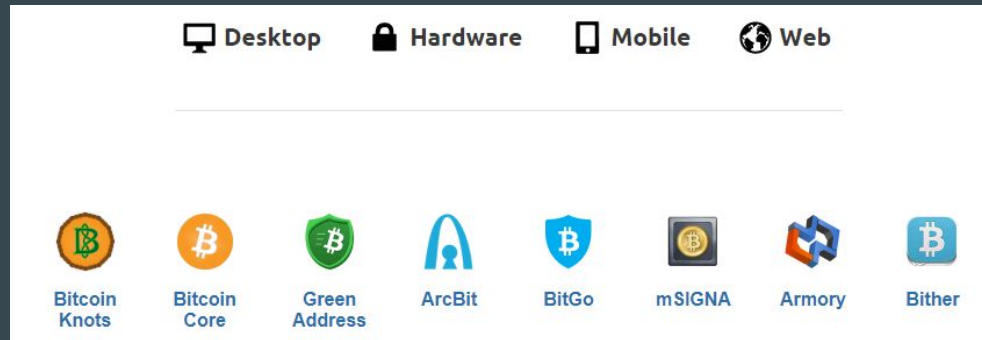


How do I store my bitcoins?

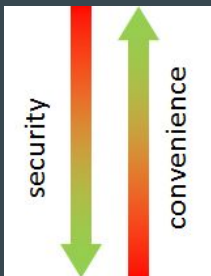
www.bitaddress.org

www.bitcoin.org

1. Create a wallet
 - a. online (web service)
 - b. offline (software)
 - c. offline (hardware)

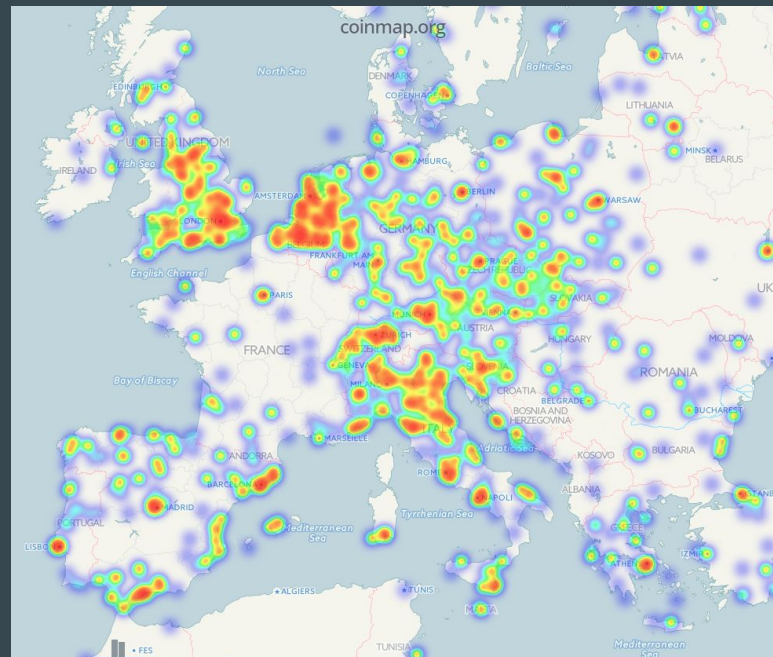


2. Keep your wallet safe
 - a. stored online
 - b. stored on computer
 - c. stored offline



Where to spend my bitcoins?

- <https://coinmap.org>
- Mobile Vikings
- Pizza.be
- Microsoft, Dell, Expedia
- Donations – bloggers, charities, community
- Things – Purse.io, overstock.com



Blockchain Use Cases

Blockchain Use cases (present)



Currency is just the first app

Remittance without middlemen (Rebit, Abra)

Proving ownership of land titles (Factom, Epigraph)

Timestamping of documents - immutable records ([Tx message](#))

Provably fair gambling - no “backstage” manipulations (SathoshiDice)

Micropayments - pay per second in video streaming (e.g. streamium.io), micro-donations (e.g. ChangeTip)

Decentralized markets - prediction (Augur), goods (OpenBazaar), currency (Bisq.network)

Blockchain Use cases (in development)

Tamper-resistant storage - police body camera videos stored on a Blockchain to ensure that video evidence has not been tampered with, and verify where and when the video is taken.

Transparent performance monitoring - making sure funds from the federal government are spent well by local governments. dynamic performance monitoring, rather than a static annual report.

Decentralized login security - login granted if user identified by >50% of participating computers

Transfer of ownership of physical goods (car, home,...) - ownership of a digital token grants access to device/home. Transfer ownership by transferring token.

Payment in IoT - devices can pay to each other without mediator