INTRODUCTION TO BITCOIN

STEFAN DZIEMBOWSKI

The First Greater Tel Aviv Area Symposium, 13.11.2014
Outline

1. Introduction
2. Main design principles
3. Bitcoin’s security?
4. Alternative ideas
Introduction
Digital vs. paper currencies

Paper:

Digital:

Very useful if is also digital.
Traditional ways of paying “digitally”

PROBLEMS

1. trusted server for each transaction is needed (money doesn’t “circulate”),
2. high transaction fees,
3. no anonymity.
Bitcoin – a “digital analogue” of the paper money


Based on the assumption that ``the majority of the computing power is honest”.

currency unit: **Bitcoin (BTC)** 1 BTC = $10^8$ Satoshi

as of 11.11.2014:

- **Market cap** $\approx 4.9$ billion USD
- 1 BTC $\approx 364$ USD
Probably one of the most discussed cryptographic technologies ever!
PROBLEMS WITH PREVIOUS APPROACHES

1. trusted server is needed (money doesn’t “circulate”),
2. high transaction fees,
3. no anonymity.

*in Bitcoin:*

- no trusted server, money circulates
- low fees
- “pseudonymity”
“No trusted server”

nobody “controls the money”, and therefore:

– The amount of money that will ever be “printer” is fixed (to around 21 mln BTC) → no inflation
– The exchange rate fluctuates:
Really “no trusted server”? 

The client software is written by people who are in power to change the system.

For example, this is the list of “desktop clients”:

The most popular client.

(open source)

The developers: Wladimir J. van der Laan, Gavin Andresen, Jeff Garzi, Gregory Maxwell, Pieter Wuille
Bitcoin $\approx \text{“real money”}$?

**Bitcoin** value comes from the fact that:

“people expect that other people will accept it in the future.”

enthusiasts:

![Smiley face]

It’s like all the other currencies

sceptics:

![Smiley face]

It’s a Ponzi scheme

P. Krugman  A. Greenspan
Some economists are more positive

While these types of innovations may pose risks related to law enforcement and supervisory matters, there are also areas in which they may hold long-term promise, particularly if the innovations promote a faster, more secure and more efficient payment system.

Ben Bernanke
Why did Bitcoin become so popular (1/2)?

- Ideological reasons (crypto-anarchism).
- Good timing (in 2008 the “quantitative easing” in the US started).
- Seeming anonymity (anonymous enough for trading illegal goods?)
Why did Bitcoin become so popular (2/2)?

• Low transaction fees.

• Hype?

• Very popular in some non-democratic countries (until the government forbids to use it).
Downsides of decentralization (1/2)

There are no “regulators”...

*MtGox* (handling 70% of all Bitcoin transactions) shut down on **Feb 2014** reporting 850,000 bitcoins (≈ **450 million USD**) stolen.
Downsides of decentralization (2/2)

Nobody can reverse transactions, so finally hackers have good reasons to break into personal computers.
Main design principles
Main problem with the digital money

Double spending...

Bits are easier to copy than paper!
Bitcoin idea (simplified):

The users emulate a **public trusted bulletin-board** containing a list of transactions.

A transaction is of a form:

“User \( P_1 \) transfers a coin \( \#16fab13fc6890 \) to user \( P_2 \)”

**This prevents double spending.**
What needs to be discussed

1. How is the **trusted bulletin-board** maintained?
2. How are the users identified?
3. Where does the money come from?
4. What is the syntax of the transactions?
Main difficulty: Some parties can cheat.

Classical result: simulation is possible if the “majority is honest”. For example for 5 players we can tolerate at most 2 “cheaters”.
Problem

How to define “majority” in a situation where everybody can join the network?
The Bitcoin solution

Define the “majority” as the majority of the computing power
Now creating multiple identities does not help!
How is this verified?

Main idea:

• use *Proofs of Work*
• *incentivize* honest users to constantly participate in the process

The honest users can use their *idle CPU cycles*.

**Nowadays**: often done on *dedicated hardware*. 
A simple hash-based PoW

$H$ -- a hash function whose computation takes time $\text{TIME}(H)$

**Prover** finds $s$ such that $H(s,x)$ starts with $n$ zeros (in binary)

- **salt**
- “hardness parameter"

takes time $2^n \cdot \text{TIME}(H)$

random $x$

**Verifier** checks if $H(s,x)$ starts with $n$ zeros

takes time $\text{TIME}(H)$
The users participating in the scheme are called the “miners”.

They maintain a chain of blocks:

- the “genesis block” created by Satoshi on 03/Jan/2009
- \( \text{block}_0 \) with transactions from period 1
- \( \text{block}_1 \) with transactions from period 2
- \( \text{block}_2 \) with transactions from period 3
- \( \text{block}_3 \) with ≈ 10 min.
How to post on the board

Just broadcast (over the internet) your transaction to the miners.

And hope they will add it to the next block.

The miners are incentivized to do it.
Main principles

1. It is **computationally hard** to extend the chain.

2. Once a miner finds an extension he **broadcasts it to everybody**.

3. The users will always accept “**the longest chain**” as the valid one.
How are the PoWs used?

**Main idea**: to extend it one needs to find salt such that

\[ H(salt, block_i, transactions) \] starts with some number n of zeros
The hardness parameter is periodically changed

- The computing power of the miners changes.
- The miners should generate the new block each 10 minutes (on average).
- Therefore the hardness parameter is periodically adjusted to the mining power
- This happens once each 2016 blocks.
- For example the block generated on 2014-03-17 18:52:10 looked like this:

000000000000000006d8733e03fa9f5e52ec912fa82c9adfed09fbca9563cb4ce
“Hashrate” = number of hashes computed per second

Note:

Nov 05 2014 : 283,494,086 GH/s
Nov 05 2013 : 3,657,378 GH/s

\[ \approx 2^{58} \text{ hash / second} \]
# How it looks in real life

<table>
<thead>
<tr>
<th>ID</th>
<th>Hash</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>291061</td>
<td>109adb5479...</td>
<td>2014-03-17 18:49:11</td>
</tr>
<tr>
<td>291060</td>
<td>418788ad79...</td>
<td>2014-03-17 18:44:35</td>
</tr>
<tr>
<td>291059</td>
<td>675b86077a...</td>
<td>2014-03-17 18:34:59</td>
</tr>
<tr>
<td>291058</td>
<td>ebcce6837fa...</td>
<td>2014-03-17 18:29:34</td>
</tr>
<tr>
<td>291057</td>
<td>ee7453e6d0...</td>
<td>2014-03-17 17:47:28</td>
</tr>
<tr>
<td>291056</td>
<td>d2c08a5ee9...</td>
<td>2014-03-17 17:26:21</td>
</tr>
</tbody>
</table>
What if there is a “fork”?

The “longest” chain counts.
Does it make sense to “work” on a shorter chain?

No!

Because everybody else is working on extending the longest chain.

Recall: we assumed that the majority follows the protocol.
How are the miners incentivized to participate in this game?

**Short answer:** they are paid (in Bitcoins) for this. We will discuss it in detail later...
What needs to be discussed

1. How is the **trusted bulletin-board** maintained?
2. How are the users identified?
3. Where does the money come from?
4. What is the syntax of the transactions?
User identification

We use the digital signature schemes.

The users are identified by their public keys.
Digital signature schemes

A **digital signature scheme** consists of algorithms **Gen**, **Sign** and **Vrfy**, where:

- **Input:**
  - `(sk, message M)`
  - `(pk, M, σ)`

- **Output:**
  - (secret key `sk`, public key `pk`)
  - signature `σ`
  - yes/no

**Correctness:**

for every `(sk, pk) := Gen()` and every `M` we have

\[ Vrfy(pk, M, Sign(sk, M)) = yes \]

**Security:**

“without knowing `sk` it is infeasible to compute `σ` such that

\[ Vrfy(pk, M, σ) = yes \]
Anonymity?

Can sometimes be de-anonymized: [Meiklejohn et al. *A fistful of bitcoins: characterizing payments among men with no names*, 2013]
What needs to be discussed

1. How is the trusted bulletin-board maintained?
2. How are the users identified?
3. Where does the money come from?
4. What is the syntax of the transactions?
Where does the money come from?

A miner who finds a new block gets a “reward” in BTC:

- for the first 210,000 blocks: 50 BTC
- for the next 210,000 blocks: 25 BTC
- for the next 210,000 blocks: 12.5 BTC,
  and so on…

Note: $210,000 \cdot (50 + 25 + 12.5 + \cdots) \rightarrow 21,000,000$
This is how it looks in detail

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Fee</th>
<th>Size (kB)</th>
<th>From (amount)</th>
<th>To (amount)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0ac34e9949...</td>
<td>0</td>
<td>0.173</td>
<td>Generation: 25 + 0.05974785 total fees</td>
<td>1KFHE7w8BhaENAawwryaoeDb6qeT6DbYY: 25.05974785</td>
</tr>
<tr>
<td>2055f19a51...</td>
<td>0.0002</td>
<td>0.259</td>
<td>1Kpv8JEcWLhUqidq8dnrwxiaZPKL4KUoeR: 179.9998</td>
<td>1HCukLGkCfKCryXT73hj2SyVAC9kzRGkC: 105</td>
</tr>
<tr>
<td>66815aff01...</td>
<td>0.001</td>
<td>0.258</td>
<td>1dice6DPtUMBpWgv8jdG8HMjXv9qDJWN: 0.35</td>
<td>15GPjviasjMD8QJvMTcS5qYsB8wQlQGbtP: 0.00175</td>
</tr>
</tbody>
</table>

“generation transaction”

“coinbase”
More details

Each block contains a transaction that transfers the reward to the miner.

**Advantages:**

1. It provides incentives to be a miner.
2. It also makes the miners interested in broadcasting new block asap.

This view was challenged in a recent paper:

Ittay Eyal, Emin Gun Sirer

*Majority is not Enough: Bitcoin Mining is Vulnerable*

(we will discuss it later)
What needs to be discussed

1. How is the **trusted bulletin-board** maintained?
2. How are the users identified?
3. Where does the money come from?
4. What is the syntax of the transactions?
Transaction syntax – simplified view

$T_1 = (\text{User } P_1 \text{ creates } 25 \text{ BTC})$

in the “mining process”

$T_2 = (\text{User } P_1 \text{ sends } 25 \text{ BTC from } T_1 \text{ to } P_2 $

signature of $P_1$ on $[T_2]$)

$T_3 = (\text{User } P_2 \text{ sends } 25 \text{ BTC from } T_2 \text{ to } P_3 $

signature of $P_2$ on $[T_3]$)

We say that $T_3$ redeems $T_2$
How to “divide money”?

Multi-output transactions:

\[ T_2 = \text{(User } P_1 \text{ sends 10 BTC from } T_1 \text{ to user } P_2, \]
\[ \text{User } P_1 \text{ sends 7 BTC from } T_1 \text{ to user } P_3, \]
\[ \text{User } P_1 \text{ sends 8 BTC from } T_1 \text{ to user } P_4 \text{ signature of } P_1 \text{ on } [T_2]) \]
Bitcoin’s security?
Possible attack goals

- double spending,

- get more money from mining than you should,

- “short selling” – bet that the price of BTC will drop and then destroy the system (to make the price of BTC go to zero),

- someone (government?) interested in shutting Bitcoin down...

**Note:** this can be done e.g. by a spectacular fork that lasts just for a few hours...
What we do (not) know about Bitcoin’s security?

1. Technical errors
2. Features/problems
3. Conceptual errors
4. Potential threats
Some notable cases of **programming errors**

- A block 74638 ([Aug 2010](#)) contained a transaction with **two outputs summing to over 184 billion BTC** – this was because of an **integer overflow** in Bitcoin software (solved by a software update and a “manual fork”). One double spending observed (worth 10,000 USD).

- A **fork** at block 225430 ([March 2013](#)) caused by an **error in the software update** of Bitcoin Core (lasted 6 hours, solved by reverting to an older version of the software).

**Moral:** nothing can be really “completely distributed”. Sometimes human intervention is needed...
Transaction Malleability

Problem: transactions are identified by their hashes

\[ T_2 = (\text{User } P_1 \text{ sends } 1 \text{ BTC from } T_1 \text{ to } P_2, \text{ signature of } P_1 \text{ on } [T_2]) \]

\[ \text{TxId} = \text{Hash}(T_2) \]

Hence one can change \text{TxId} by mauling the signature:

\[ (\text{User } P_1 \text{ sends } 1 \text{ BTC from } T_1 \text{ to } P_2, \sigma) \rightarrow (\text{User } P_1 \text{ sends } 1 \text{ BTC from } T_1 \text{ to } P_2, \sigma') \]
How to do it?

Bitcoin uses **ECDSA** signatures. Hence:

\[
\sigma = (r, s) \quad \text{is a valid signature on } \quad M \text{ w.r.t. } pk
\]

\[
\sigma' = (r, -s \pmod{N}) \quad \text{is a valid signature on } \quad M \text{ w.r.t. } pk
\]

Other methods also exists…
[Andrychowicz et al 2014, unpublished work]: very easy to perform in practice.
Is it a problem?

Often: **NO**

(the mauled transaction is semantically equivalent to the original one)

When things can go wrong?

• Bitcoin contracts
• buggy software
Claimed attack on MtGox

Since MtGox cannot see a transaction with \text{TxId} \text{Hash}(T) in the blockchain.

Thus it concludes that the transaction did not happen. (so A can double spend)

\[ \text{transaction } T = \text{“MtGox sends 1 BTC to A”} \]

\[ \text{transaction } T' = \text{mauled transaction } T \]

[Decker and Wattenhofer, ESORICS 2014]: this is probably not true.
What we do (not) know about Bitcoin’s security?

1. Technical errors
2. Features/problems
3. Conceptual errors
4. Potential threats
Hardware mining

History of mining:

CPU → GPU → FPGA → ASIC

<table>
<thead>
<tr>
<th>Product</th>
<th>Advertised Mhash/s</th>
<th>Mhash/J</th>
<th>Mhash/s/$</th>
<th>Watts</th>
<th>Price (USD)</th>
<th>Currently shipping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Achilles Labs AM-850 [1]</td>
<td>850,000</td>
<td>1478</td>
<td>1223</td>
<td>575</td>
<td>695</td>
<td>Discontinued</td>
</tr>
<tr>
<td>Achilles Labs AM-1700 [2]</td>
<td>1,700,000</td>
<td>1581</td>
<td>1553</td>
<td>1075</td>
<td>1095</td>
<td>Yes</td>
</tr>
<tr>
<td>Achilles Labs AM-3400 [3]</td>
<td>3,400,000</td>
<td>1581</td>
<td>1794</td>
<td>2150</td>
<td>1895</td>
<td>Yes</td>
</tr>
<tr>
<td>Achilles Labs AM-6000 [4]</td>
<td>6,000,000</td>
<td>1579</td>
<td>2073</td>
<td>3800</td>
<td>2895</td>
<td>Yes</td>
</tr>
<tr>
<td>AntMiner S2 [7]</td>
<td>1,000,000</td>
<td>900</td>
<td>442</td>
<td>1100</td>
<td>2259</td>
<td>Discontinued</td>
</tr>
<tr>
<td>AntMiner S3 [8]</td>
<td>441,000</td>
<td>1300</td>
<td>1154</td>
<td>340</td>
<td>382[6]</td>
<td>Yes</td>
</tr>
<tr>
<td>AntMiner S4 [9]</td>
<td>2,000,000</td>
<td>1429</td>
<td>1429</td>
<td>1400</td>
<td>1400</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Drawbacks of the hardware mining

1. Makes the whole process "non-democratic".

2. Easier to attack by very powerful adversary?

3. Excludes some applications (mining a as “micropayment”).
Advantages of the hardware mining

• Security against botnets.

• Makes the miners interested in the long-term stability of the system.

How “long term”?

Remember that the total hashrate went up almost 100x over the last year...
Mining pools

Miners create cartels called the mining pools. This allows them to reduce the variance of their income.

Note:

The total hashrate of the Bitcoin system as of 5.11.2014

\[
\frac{283,494,086 \text{ GHash/s}}{1,700 \text{ GHash/s}} \approx 166,761 = 3.17 \cdot (365 \cdot 24 \cdot 6)
\]

The hashrate of the Achilles Labs AM-1700 miner (1095 USD)

The user has to wait on average over 3 years to mine a block (even if the difficulty does not increase!)
Popular mining pools
The general picture

The mining pool is **operated centrally**.

Some of the mining pools **charge fees for their services**.

**Tricky part**: how to prevent cheating by miners? How to reward the miners?

(see Meni Rosenfeld: *Analysis of Bitcoin Pooled Mining Reward Systems*)
June 2014

Ghash.io got > 50% of the total hashpower.

Then this percentage went down...
Observation

What we were promised:

“distributed currency independent from the central banks”

What we got (in June 2014):

“currency controlled by a single company”...
A problem

Individual miners **lost control** over which blocks they mine.

For example in the **Stratum** protocol (commonly used by mining pools):

```
miners cannot choose Bitcoin transactions on their own
```

From mining.bitcoin.cz/stratum-mining:

“In my experience **99% of real miners don’t care about transaction selection anyway, they just want the highest possible block reward.** At this point they share the same interest with pool operator, so there’s no real reason to complicate mining protocol just for those 1% who want to create custom blocks for the pool.”
How to break Bitcoin?

1. Start a number of mining pools with a negative fee.
2. Wait until you get >50% of the total hashrate.

Will the miners join?

they just want the highest possible block reward...
What is really our security assumption?

1. No cartel controls the majority of the computing power, or
2. The majority of participants is 100% honest.

“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.”

we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power
In order for the Bitcoin to work we need a following (strong) assumption:

*The majority behaves honestly even if it has incentives not to do so.*

Is it realistic?

**enthusiast:**

Yes, since the majority is interested in maintaining the system

**sceptics:**

No, since this is not how capitalism works... *(e.g.: tragedy of the commons)*
Another risk

Why not to rent the hashpower to perform the attack?
Conjecture

Maybe the only reason why nobody broke Bitcoin yet is that there is no good way to short-sell BTC?
What we do (not) know about Bitcoin’s security?

1. Technical errors
2. Features/problems
3. Conceptual errors
4. Potential threats
Selfish mining

Ittay Eyal, Emin Gun Sirer
Majority is not Enough: Bitcoin Mining is Vulnerable

basic idea: when you mine a new block keep it to yourself.

We explain it with some simplifying assumptions.
What happens when there is a fork?

**Bitcoin specification:**

“from two blocks of equal length mine on the first one that you received”.

Assume that the adversary is always first (e.g. he puts a lot of “fake nodes” that act as sensors).
Assume that the adversary does not broadcast the new block that he found (and mines on it “privately”).

Two things can happen:

1. the adversary manages to extend his “private block chain” by one more block, or
2. the “honest users” manage to find an alternative extension.

In this case the adversary quickly publishes his block so he looses nothing.
If the adversary is lucky then he obtains advantage over the honest miners.

Note: this works even if the adversary has minority of computing power.
The assumption that “the adversary is always first” may look unrealistic.

Eyal and Sir show a modification of this strategy that works without this assumption.

- $\gamma$ – probability that the honest users choose adversary’s block
- $\alpha$ – fraction of adversary’s computing power

Their strategy works as long as $\alpha > \frac{1-\gamma}{3-2\gamma}$
Another clever attack

Lear Bahack *Theoretical Bitcoin Attacks with less than Half of the Computational Power*

The “Difficulty Raising Attack” – exploits the way the difficulty is adjusted in Bitcoin.
What we do (not) know about Bitcoin’s security?

1. Technical errors
2. Features/problems
3. Conceptual errors
4. Potential threats
Blocks without transactions

Example:

Block 310084

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Fee</th>
<th>Size (kB)</th>
<th>From (amount)</th>
<th>To (amount)</th>
</tr>
</thead>
<tbody>
<tr>
<td>660c7e9af8e</td>
<td>0</td>
<td>0.171</td>
<td>Generation: 25 + total fees</td>
<td>1KFHE7w8BhalENAswwwyapecDb6qeT6DbYY: 25</td>
</tr>
</tbody>
</table>

Reason: shorter blocks propagate faster.
In the future the opposite problem can happen

When the mining reward becomes negligible, we can experience:

**Tragedy of the commons:**

adding a transaction costs nothing, so the miners will not be able to keep the transaction fees high.
Yet another question

What happens if someone posts a transaction $T$ with a very high fee (say 100 BTC)?

For them it's more profitable to mine on the old block.
Alternative ideas
Litecoin

Released in **Oct 2011** by Charles Lee.

Uses **scrypt** hash function introduced in:

Idea: **scrypt** is memory-hard, so there should be no hardware-mining.

as of 11.11.2014:

**Market cap** ≈ 124 million USD
**1 BTC** ≈ 3.68 USD

really?

Asic Scrypt-N Miner Wolf V1 2048 Mh/s (2GH):

$19,995.00 excl. VAT

- Specifications
  - Hashrate: 2048 Mh/s (2 GH/s) for Scrypt (N)
  - Performance guarantee 100%
Proofs of Stake

The “voting power” depends on how much money one has.

**Justification**: people who have the money are naturally interested in the stability of the currency.

**Currencies**: BlackCoin, Peercoin, NXT,

Also has some problems...
Proofs of Space

Replace work by disk space.
S. Dziembowski, S. Faust, V. Kolmogorov, K. Pietrzak, Proofs of Space.

Main advantages:
– no “dedicated hardware”,
– less energy wasted (“greener”).

Problem: hard to construct (only interactive Proofs of Space are known)
Preventing mining pool creation

**Idea:** help the mining pool members to cheat.

Andrew Miller, Elaine Shi, and Jonathan Katz. *Non-outsourceable Scratch-Off Puzzles to Discourage Bitcoin Mining Coalitions*. June 2014
Conclusion

1. People want “cryptocurrencies”.
2. Bitcoin has some important weaknesses, new ideas are needed.
3. Tricky security model.
4. Bitcoin ideas that are interesting on their own:
   a) consensus based on the PoW
   b) generalized transaction format
Thank you!