BITCOIN
(A BASIC TUTORIAL)

Aviv Zohar
The Rachel & Selim Benin School of Eng. and Computer Science
The Hebrew University
In this tutorial:

- What is Bitcoin and how does it work?
- What are the main challenges?
- The surrounding ecosystem
- Pointers to related research & additional sources of information
Money isn’t perfect

Currently slower and more expensive than:
Invented by Satoshi Nakamoto in 2008
Launched in 2009

A decentralized digital currency

Built for the age of the internet
Features of Bitcoin

- Pseudonymous
- Fixed amount
- Irreversible Transfers
- Cannot be seized
- Can not be frozen
- Escrow
- Joint accounts
<table>
<thead>
<tr>
<th>Last Price:</th>
<th>$655.38</th>
<th>Daily Change:</th>
<th>$8.22</th>
<th>1.27%</th>
<th>Day's Range:</th>
<th>$635.88 - $656.84</th>
<th>Today's Open:</th>
<th>$647.16</th>
<th>24h Volume</th>
<th>8646 BTC</th>
</tr>
</thead>
</table>

Market Cap: $8,432,610,615.00  
Total BTC: 12,866,750 BTC  

*From Bitstamp.net*
• Bypass regulation & censorship
• Increase competition
• Disrupt

Please transfer $1 to Blue

Blue: 2
Red: 1
Transactions are thus public, addresses are (free) pseudonyms.
The Double-spend problem

A variant of the Byzantine general’s problem (Byzantine consensus in asynchronous dist. systems)
- Blocks aggregate transactions in batches
- Each block contains a cryptographic hash of the previous one, “proving” it is created afterwards.
- Can Read ledger from start to finish to “follow the money”
- Each node tries to grow the chain with recent transactions:
  - Create a block with recent consistent transactions
  - Send to peers
Inconsistency may occur if blocks are created simultaneously by different nodes

(double spend problem)
Solution:
1. Make block creation hard.
2. Adopt conflicting blocks if they make up a longer chain.

~ one block per 10 min. in the entire network
(Difficulty scales automatically to maintain this)
Current target has ~65 zeros in most significant digits
1. Make block creation hard (once every 10 minutes)

2. Adopt (conflicting) blocks iff they make up a longer chain.
The Double-Spend Attack

- A payment can be reversed!
- Easy if attacker has >50% of compute power
- Possible with less than 50%

**Bitcoin’s Guarantee [Satoshi]:**
If attacker controls < 50% of compute power, probability of block replacement decreases exponentially with time.
To encourage nodes to authorize transactions:

Block size is limited (currently to 1MB)
Transactions will compete to enter – highest fee first. (An auction!)

Block creation is known as “Mining”
Attacks
Analysis of the Double Spend Attack

The recipient has an acceptance strategy:
- # of “confirmations” (blocks) it waits for before transaction is considered “accepted”.
- Assumption: attacker has hashrate $q$. Yields distribution over the # of blocks in its chain.
Analysis of the Attack

- Consider a Markov Process representing the difference in length between the chains

If we ever get here, Attacker wins

Honest chain length minus attacker's

-1  0  1  2  3

Attacker creates block (q)
Network creates block (1-q)

n blocks built by honest nodes, attacker has strength q → probability distribution over initial states ∈ \{n, n − 1, n − 2, ... \}.
The Result:

Attacker’s strength: \( q < 0.5 \)
Receiver’s policy: wait for \( n \) confirmations

Probability of successful attack:

\[
    r = 1 - \sum_{m=0}^{n} \binom{m + n - 1}{m} \cdot ((1 - q)^n q^m - (1 - q)^m q^n)
\]

Result due to Meni Rosenfeld: “Analysis of hashrate-based double-spending”
<table>
<thead>
<tr>
<th>q</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>2%</td>
<td>4%</td>
<td>0.237%</td>
<td>0.016%</td>
<td>0.001%</td>
<td>≈0</td>
<td>≈0</td>
<td>≈0</td>
<td>≈0</td>
<td>≈0</td>
<td>≈0</td>
</tr>
<tr>
<td>4%</td>
<td>8%</td>
<td>0.934%</td>
<td>0.120%</td>
<td>0.016%</td>
<td>0.002%</td>
<td>≈0</td>
<td>≈0</td>
<td>≈0</td>
<td>≈0</td>
<td>≈0</td>
</tr>
<tr>
<td>6%</td>
<td>12%</td>
<td>2.074%</td>
<td>0.394%</td>
<td>0.078%</td>
<td>0.016%</td>
<td>0.003%</td>
<td>0.001%</td>
<td>≈0</td>
<td>≈0</td>
<td>≈0</td>
</tr>
<tr>
<td>8%</td>
<td>16%</td>
<td>3.635%</td>
<td>0.905%</td>
<td>0.235%</td>
<td>0.063%</td>
<td>0.017%</td>
<td>0.005%</td>
<td>0.001%</td>
<td>≈0</td>
<td>≈0</td>
</tr>
<tr>
<td>10%</td>
<td>20%</td>
<td>5.600%</td>
<td>1.712%</td>
<td>0.546%</td>
<td>0.178%</td>
<td>0.059%</td>
<td>0.020%</td>
<td>0.007%</td>
<td>0.002%</td>
<td>0.001%</td>
</tr>
<tr>
<td>12%</td>
<td>24%</td>
<td>7.949%</td>
<td>2.864%</td>
<td>1.074%</td>
<td>0.412%</td>
<td>0.161%</td>
<td>0.063%</td>
<td>0.025%</td>
<td>0.010%</td>
<td>0.004%</td>
</tr>
<tr>
<td>14%</td>
<td>28%</td>
<td>10.662%</td>
<td>4.400%</td>
<td>1.887%</td>
<td>0.828%</td>
<td>0.369%</td>
<td>0.166%</td>
<td>0.075%</td>
<td>0.034%</td>
<td>0.016%</td>
</tr>
<tr>
<td>16%</td>
<td>32%</td>
<td>13.722%</td>
<td>6.352%</td>
<td>3.050%</td>
<td>1.497%</td>
<td>0.745%</td>
<td>0.375%</td>
<td>0.190%</td>
<td>0.097%</td>
<td>0.050%</td>
</tr>
<tr>
<td>18%</td>
<td>36%</td>
<td>17.107%</td>
<td>8.741%</td>
<td>4.626%</td>
<td>2.499%</td>
<td>1.369%</td>
<td>0.758%</td>
<td>0.423%</td>
<td>0.237%</td>
<td>0.134%</td>
</tr>
<tr>
<td>20%</td>
<td>40%</td>
<td>20.800%</td>
<td>11.584%</td>
<td>6.669%</td>
<td>3.916%</td>
<td>2.331%</td>
<td>1.401%</td>
<td>0.848%</td>
<td>0.516%</td>
<td>0.316%</td>
</tr>
<tr>
<td>22%</td>
<td>44%</td>
<td>24.781%</td>
<td>14.887%</td>
<td>9.227%</td>
<td>5.828%</td>
<td>3.729%</td>
<td>2.407%</td>
<td>1.565%</td>
<td>1.023%</td>
<td>0.672%</td>
</tr>
<tr>
<td>24%</td>
<td>48%</td>
<td>29.030%</td>
<td>18.650%</td>
<td>12.339%</td>
<td>8.310%</td>
<td>5.664%</td>
<td>3.895%</td>
<td>2.696%</td>
<td>1.876%</td>
<td>1.311%</td>
</tr>
<tr>
<td>26%</td>
<td>52%</td>
<td>33.530%</td>
<td>22.868%</td>
<td>16.031%</td>
<td>11.427%</td>
<td>8.238%</td>
<td>5.988%</td>
<td>4.380%</td>
<td>3.220%</td>
<td>2.377%</td>
</tr>
<tr>
<td>28%</td>
<td>56%</td>
<td>38.259%</td>
<td>27.530%</td>
<td>20.319%</td>
<td>15.232%</td>
<td>11.539%</td>
<td>8.810%</td>
<td>6.766%</td>
<td>5.221%</td>
<td>4.044%</td>
</tr>
<tr>
<td>30%</td>
<td>60%</td>
<td>43.200%</td>
<td>32.616%</td>
<td>25.207%</td>
<td>19.762%</td>
<td>15.645%</td>
<td>12.475%</td>
<td>10.003%</td>
<td>8.055%</td>
<td>6.511%</td>
</tr>
<tr>
<td>32%</td>
<td>64%</td>
<td>48.333%</td>
<td>38.105%</td>
<td>30.687%</td>
<td>25.037%</td>
<td>20.611%</td>
<td>17.080%</td>
<td>14.226%</td>
<td>11.897%</td>
<td>9.983%</td>
</tr>
<tr>
<td>34%</td>
<td>68%</td>
<td>53.638%</td>
<td>43.970%</td>
<td>36.738%</td>
<td>31.058%</td>
<td>26.470%</td>
<td>22.695%</td>
<td>19.548%</td>
<td>16.900%</td>
<td>14.655%</td>
</tr>
<tr>
<td>36%</td>
<td>72%</td>
<td>59.098%</td>
<td>50.179%</td>
<td>43.330%</td>
<td>37.807%</td>
<td>33.226%</td>
<td>29.356%</td>
<td>26.044%</td>
<td>23.182%</td>
<td>20.692%</td>
</tr>
<tr>
<td>38%</td>
<td>76%</td>
<td>64.691%</td>
<td>56.698%</td>
<td>50.421%</td>
<td>45.245%</td>
<td>40.854%</td>
<td>37.062%</td>
<td>33.743%</td>
<td>30.811%</td>
<td>28.201%</td>
</tr>
<tr>
<td>40%</td>
<td>80%</td>
<td>70.400%</td>
<td>63.488%</td>
<td>57.958%</td>
<td>53.314%</td>
<td>49.300%</td>
<td>45.769%</td>
<td>42.621%</td>
<td>39.787%</td>
<td>37.218%</td>
</tr>
<tr>
<td>42%</td>
<td>84%</td>
<td>76.205%</td>
<td>70.508%</td>
<td>65.882%</td>
<td>61.938%</td>
<td>58.480%</td>
<td>55.390%</td>
<td>52.595%</td>
<td>50.042%</td>
<td>47.692%</td>
</tr>
<tr>
<td>44%</td>
<td>88%</td>
<td>82.086%</td>
<td>77.715%</td>
<td>74.125%</td>
<td>71.028%</td>
<td>68.282%</td>
<td>65.801%</td>
<td>63.530%</td>
<td>61.431%</td>
<td>59.478%</td>
</tr>
<tr>
<td>46%</td>
<td>92%</td>
<td>88.026%</td>
<td>85.064%</td>
<td>82.612%</td>
<td>80.480%</td>
<td>78.573%</td>
<td>76.836%</td>
<td>75.234%</td>
<td>73.742%</td>
<td>72.342%</td>
</tr>
<tr>
<td>48%</td>
<td>96%</td>
<td>94.003%</td>
<td>92.508%</td>
<td>91.264%</td>
<td>90.177%</td>
<td>89.201%</td>
<td>88.307%</td>
<td>87.478%</td>
<td>86.703%</td>
<td>85.972%</td>
</tr>
<tr>
<td>50%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 1: The probability of a successful double spend, as a function of the attacker’s hashrate $q$ and the number of confirmations $n$.

From Meni Rosenfeld’s paper “Analysis of hash-rate based double spending”.
Implications

- To get final approval for a transaction one has to wait several blocks (confirmations).
- Each block takes 10 minutes in expectation.

Risk of an attack should take transaction size into account.
The Times 03/Jan/2009

Chancellor on brink of second bailout for banks

Eat Out from £5
More than 900 great restaurants, including four Gordon Ramsay favourites from £5

Israel prepares to send tanks and troops into Gaza

Chancellor on brink of second bailout for banks
Billions may be needed as lending squeeze tightens
The Finney attack

Some Vendors cannot afford to wait. Accept 0-confirmation transactions.

Susceptible to a simple attack:

- Alice pre-mines block with a transaction to self.
- Alice creates and sends transaction paying Bob. Instantly receives goods from Bob.
- Alice releases pre-mined block before the transaction to Bob is authorized.
Additional Attack Vectors

- Network-structure attacks
  - Isolating a node implies you can use its computational power to launch double spend attacks
  - Sybil attacks

- DDoS attacks with amplification
  - Blocks are secure by difficulty, blocks that are too old are not allowed
  - Transactions are secured by fee

- Clock Drift attacks (Timejacking)

- 0-Confirmation attacks & chain splits based on different versions
Transactions
Addresses

- Addresses are (essentially) public keys
- Allow sending Bitcoins even when recipient is offline
- Signatures are used to prove ownership (generated with private keys)
- Security matters! paper wallets / cold storage.
Passphrase
\nlove

Secret Exponent
686f746a95b6f836d7d70567c302c3f9ebb5ee0def3d1220ee9d4e9f34f5e131

Point Conversion
Uncompressed  Compressed

Private Key
5JcHF3GtHTXHm2vWLYevaBYmp1MLEmrhQu4hL4gaPpXWxaQrJsa

Address
1Mm6ouhpHqbtahCRNYfT07Art1fbmk7PcR

Transactions (Newest First)

<table>
<thead>
<tr>
<th>Amount</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0095BTC</td>
<td>1Gh861fxMLVbpfAichU1YKm55q2pStlDsP</td>
</tr>
<tr>
<td>-0.01BTC</td>
<td>1Mm6ouhpHqbtahCRNYfT07Art1fbmk7PcR</td>
</tr>
<tr>
<td>0.01BTC</td>
<td>1GKRXcbfhJa6buSy657toKaiafTRehja</td>
</tr>
</tbody>
</table>
Transactions

- Each transaction is a transfer of money from inputs to outputs (many-to-many)

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 BTC</td>
<td>1 BTC</td>
</tr>
<tr>
<td>1.5 BTC</td>
<td>1 BTC</td>
</tr>
</tbody>
</table>

(The fee is the difference between outputs and inputs)
A transaction is valid if and only if
- It contains all required signatures,
- every input matches a previous unspent output
Transactions

- **outputs** specify amount and “script” for redeeming money.

```
OP_DUP OP_HASH160 5df3d323c10c8563e9086074c4bd94eab97b95a8 0P_EQUALVERIFY OP_CHECKSIG
```

- **Inputs** specify data for script to return “True”

- Some outputs cannot be redeemed.
Scripts allow for much more...

- k out of n signatures
- Delayed payments
- Savings accounts
- P2P bets
- Derivatives
- Distributed exchanges
- Implemented on top of Bitcoin
- or in alternative chains
Modifications of the protocol
Altcoins

Many Bitcoin clones
Zerocoin / Zerocash

[Ben-Sasson, Chiesa, Garman, Green, Miers, Tromer, and Virza]

- Improved anonymity for Bitcoin using advanced cryptographic tools
  - zero-knowledge Succinct Non-interactive ARguments of Knowledge (zk-SNARKs)

- Hides transaction origin, destination & amount.

- Most importantly: efficient implementation makes otherwise heavy crypto practical
Can Bitcoin Be Faster?

Block rate: one every 10 minutes

2.5 minutes

12 seconds

What is the effect of this? Why not go even faster?
Two related problems
[Sompolinsky & Zohar]

- A block every 10 minutes
  - A Long wait for transaction confirmations

- 1MB per block (per 10 minutes)
  - A limit on number of transactions per second (3.3 TPS)
Larger blocks

Higher block creation rates

More forks in chain

*Data generously shared by Decker & Wattenhofer*
Larger blocks

Higher block creation rates

More forks in chain

Modest increase in TPS

![Graph showing transactions per second (TPS) vs. block creation rate (λ)]
- Larger blocks
- Higher block creation rates
- More forks in chain
- Modest increase in TPS
- Lower security

50% attack with less than 50% of hash power
Greedy Heaviest Observed Sub-Tree (GHOST)

[Sompolinsky & Zohar]

An alternative chain selection rule (instead of “longest chain”)
- Begin at the “Genesis Block”
- At every split, pick the heaviest sub-tree.

Outcome: 50% attack only works with 50% of compute power.
The Pull Towards Centralization

- Advantage of large miners:
  - Economies of scale (e.g. datacenters in Iceland)
  - Block distribution to self not needed.
  - Attractive connections for other miners

Outcome:
- Large miners gain more than proportional share.
- Drive small miners out of business.
- System becomes centralized.

- Gets worse at high block rates / large blocks
Incentives

Is the protocol “incentive compatible”?

Two main issues found thus far:

1. Miners lack the incentive to flood transaction messages to others.
   On Bitcoin and Red Balloons [Babaioff, Dobzinsky, Oren & Zohar]

2. Miners do not necessarily want to mine on top of latest block or release their block instantly
   “Majority is not Enough“ [Eyal & Sirer]
Miners do not necessarily want to mine on top of latest block.
Mining Pools

- Bitcoin mining is a high risk “lottery”
- Miners can join together to split profits and reduce risk

Diagram:
- Mining Pool Server
- Miner
- Block header
- Fees
- Nonce
Hash rate distribution (from Blockchain.info)
How (not) to split rewards

- Miners that contribute more should get higher reward.

- **Win**: $\text{Hash}(\text{header}) < \text{target}$
- **Get a share**: $\text{Hash}(\text{header}) < k \cdot \text{target}$

Pay per share:

Split wins proportionately to # of shares contributed.
Pool Hopping

It is not known when a block will be created by the pool (a memoryless process).

- The first share may be worth a lot (if block found right after)
- The 50\textsuperscript{th} share is already very “diluted”
- Miners are better off switching to another pool / solo mining after several shares have been found.

Hop-proof reward schemes exist. Explore tradeoff between risk to pool, risk to player and time. [Meni Rosenfeld]
More on Block Structure: Merkle Trees

Specifying the root, is equivalent to committing to all transactions in the tree (unless we can easily find hash collisions)
Root of the Merkle tree is thus included in the block header.

Block Header (80 Byte)

P. Block Hash
Nonce
Merkle Root
Other Fields...

Hash

00001001011011001

Smaller than target value

Block Body
Light nodes

- Running a full Bitcoin node may be too expensive. (e.g. for smartphones)

- To prove that transaction occurred:
  - Download block headers and check nonce values, Merkle root
  - Request Merkle “branch” leading from some block to root
Saving space

- The same scheme allows full nodes to save space.

“Spent” transactions no longer needed
Unspent transaction outputs

- What about proving that money is in someone else’s account? (Unspent output)

- Suggested modification: Include a Merkle root of unspent transactions in the header.

- Show a Merkle branch to the output.

- Allows for more space savings
Suggested Reading

- Bitcoin Wiki
- BitcoinTalk forums
- Bitcoin on Stack-Exchange

Some papers (in no particular order):
- Eyal, Ittay, and Emin Gün Sirer. "Majority is not enough: Bitcoin mining is vulnerable." FC 2014.
Thank You!