HOW BITCOIN WORKS

Matthäus Wander

June 29, 2011
Overview

• Electronic currency system
• Decentralized
  – No trusted third party involved
  – Unstructured peer-to-peer network
• Non-reversible
  – Cryptographic proof instead of trust
• Open source beta software (C++, wxWidgets)
• Currency unit: BTC
Timeline

• 1993: ecash (DigiCash/David Chaum)
• 1998: Crypto-currency ideas (W. Dai, N. Szabo)
• Nov 2008: Whitepaper by Satoshi Nakamoto
• Feb 2009: Initial Bitcoin release
Concept

• Clients invest computing power to create coins
  – By solving cryptographic puzzles
  – Difficulty of puzzles adapts
  – Number of coins limited

• Public transactions for coin transfers
  – Senders and receivers have addresses
  – Authorized by private key signatures

• Honest majority prevents double-spending
  – Public transaction database
Transactions

- Public transfers between Bitcoin addresses
- Signed by previous coin owner

50 BTC created by Alice

Transaction
Bob’s Public Key

Hash

Alice’s Signature

Transaction
Carol’s Public Key

Hash

Bob’s Signature
Splitting and Combining Coins

• Coins can be split up to $10^{-8}$ BTC

<table>
<thead>
<tr>
<th>Transaction</th>
<th>In</th>
<th>Out</th>
<th>Out</th>
<th>Out</th>
<th>50 BTC</th>
<th>0.5 BTC</th>
<th>49.5 BTC</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Transaction</th>
<th>In</th>
<th>In</th>
<th>Out</th>
<th>Out</th>
<th>0.5 BTC</th>
<th>0.1 BTC</th>
<th>0.2 BTC</th>
<th>0.8 BTC</th>
</tr>
</thead>
</table>
**Bitcoin Address**

- **Fingerprint of public key**
  - 25 bytes identifier
    - Format version: 0x01
    - Fingerprint: RIPEMD160(SHA256(pub))
    - Checksum: 4 bytes of SHA256(fingerprint)

- **Base58 encoded**
  - Alphanumeric alphabet (without I, O, l, 0, +, /)
  - E.g. 1BpCB9Qzm2LePrQKu6RzASzEKvjc6utsQQ
Key Handling

• Client generates public/private key pairs
  – ECDSA 256 bit
  – Stores them in wallet file

• Wallet contains key pairs, not coins

• Private key authorizes transactions

• If keys are stolen, thief may use your coins

• If keys are lost, coins are lost

• No wallet encryption nor backup in v0.3.23 GUI
Networking (1/2)

- Unstructured peer-to-peer network
- IRC bootstrapping
  - Nickname contains IP endpoint, e.g. u4euc453wZ599zQ
  - ',u\', base58(IP address, port, checksum)
- Freenode used at the beginning
  - Users got k-lined for botnet-like behavior
- Moved to dedicated network in mid 2010
  - #bitcoin overcrowded, now using #bitcoin[00-99]
Networking (2/2)

- DNS bootstrapping
  - Without update channel
- Built-in fallback peer list
- Peer exchange
- Port 8333/TCP
- UPnP
- Purpose of networking:
  - Flood transactions
  - Share distributed database (block chain)
Mining Coins (1/3)

• Solve cryptographic challenge to create coins
  – Find a block whose hash value is below target value
  – $\text{SHA256(SHA256(block))} < \text{target}$

• Random client finds solution first

• Chance proportional to computing power

• On average one solution every 10 minutes

• Difficulty adapts to keep solving rate constant

• If found: announce block which is proof-of-work
Mining Coins (2/3)

- Payout per block: 50 BTC
- Halves every 4 years
- In 2033:
  - Payout < 1 BTC
  - 20.7 million BTC in circulation
- Total number of Bitcoins is a geometric series and approaches maximum of 21 million BTC
• Bitcoin requires processing power for operation
  – The more honest clients work, the harder cheating is
• Hash rate vs. power consumption
• GPU mining is common, CPU mining pointless
• ATI cards better suited than NVIDIA
• Mining pools share payout
• In future: FPGA, ASIC?
• Blocks commit transactions
• First transaction is generation of coins
• Clients have built-in *Genesis* block
  – Newer versions also have checkpoint blocks

• Download, validate and store block chain from untrusted network
  – Check whether block hashes < target value
  – Verify known checkpoints
  – Verify balance and check for double-spending

• Forging chain is computationally infeasible
Block chain may fork
- Due to propagation delay in p2p network
- Due to attacker injecting forged blocks

Use first block received, save the other one

Switch to other chain, if it becomes longer

Transactions confirmed after 6 blocks
- Double-spending becomes unlikely

Ignore orphaned block chains
- Generated coins mature after 100 blocks
Adapting Difficulty

- **Allowed block timestamp**
  - \( \text{time} < \text{now} + 2h \)
  - \( \text{time} > \text{median of past 11 blocks} \)

- **uint32 nBits value**
  - \( \text{nBits} \propto \frac{((\text{time}_{\text{cur}} - \text{time}_{\text{cur-2016}})}{2 \text{ weeks}}) \)
  - greater \( \propto \) less difficult

- **nBits adapts every 2016 blocks (≈2 weeks)**
  - \( \text{nBits} \cdot ((\text{time}_{\text{cur}} - \text{time}_{\text{cur-2016}}) / 2 \text{ weeks}) \)

- **256 bit target hash value**
  - \( \text{uint24} \cdot 2^{(8 \cdot (\text{uint8} - 3))} = 0x000000000000001D932F0... \)
Scalability

• Transactions flooded in network
  – Pending until someone commits them in new block

• Does not scale with number of transactions

• Block chain mirrored on all clients
  – 300 MB after 2.5 years of operation

• Storage usage can be further optimized
  – Compress block chain to 240 MB
  – Prune redeemed transactions from hash tree
  – Estimated ≈70% of transactions can be pruned
Transaction Fee and Priority

• Fee to keep transaction count low
  – Sender may pay fee to mining client who finds block
  – Fee is voluntary, but so is commitment in block

• Transaction priority \( \text{greater} \triangleq \text{higher priority} \)
  – \( \frac{\text{sum(value}_{\text{in}} \cdot \text{age}_{\text{in}})}{\text{size}} \)

• Transaction ignored if fee too low
  – May be done by both, relays and miners
  – Minimum fee depends on space left in new block

• Fee serves as incentive for mining clients
Transaction Script

- Scripting language for verification
- Simple stack machine without loops

```
Transaction

Out
OP_DUP
OP_HASH160
Address:=07b52e62...
OP_EQUALVERIFY
OP_CHECKSIG

1 BTC

In

Sig:=30450221...
PubKey:=046a6588...

Transaction

Out

OP_DUP
OP_HASH160
Address:=6d6f2539...
OP_EQUALVERIFY
OP_CHECKSIG

1 BTC
```
Privacy

- **Transactions traceable in public block chain**

<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>ea1fc5bf38...</td>
<td>0</td>
<td>0.135</td>
<td>Generation: 50 + 0.178 total fees</td>
<td>12oxGihmrgUEZRFs4p8ApzkzQzDk6M4Z26c: 50 178</td>
</tr>
<tr>
<td>0b6572ba5a...</td>
<td>0.01</td>
<td>0.439</td>
<td>16tMYLsDawEya3VPzAUFcPGEdAspxASRxy: 0.21</td>
<td>1Lsd745FDnQunjFK3wYxevLmqALD5ojAzM: 0.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1KX7s7hbo3PgwdfyTeT2Y153STVjB6E6y6D: 6.49</td>
<td>1FiWCkjmGижvWpUAM9H4YQCaR7kjH9Q: 6.5</td>
</tr>
</tbody>
</table>

- **Weak anonymity (pseudonymity)**
- Anonymity vanishes if identity linked to address
- To keep payments private, ...
  - ... keep addresses private
  - ... use different addresses
  - ... use trusted mixing service
Conclusion

• Peer-to-peer accounting system
• Relying on honest majority
• Growing public log file (block chain)
  – Limited scalability
  – Limited privacy
• Public key cryptography for authorization
• Proof-of-work to prevent double-spending
  – Requires vast amount of computing power
• Technically sophisticated experiment