MTAT.07.017
Applied Cryptography

Bitcoin

University of Tartu

Spring 2015
Bitcoin

“Bitcoin is a cryptocurrency whereby the creation and transfer of bitcoins is facilitated by an open-source peer-to-peer cryptographic protocol that functions without the intermediation of any central authority.”

http://en.wikipedia.org/wiki/Bitcoin
Traditional Banks

- Authenticates account holders and performs transactions
- Provides authenticity of transaction log
- Resolves disputes

How to do that without trusted central authority?
Bitcoin

- How to maintain transaction log without central authority?
  - Distribute to everyone over peer-to-peer network
- How to verify account holder’s intent without central authority?
  - Account holder signs transactions using digital signature
- How to bind account holder’s identity to public key?
  - Public key is an identity / account number itself
  - Anyone who can sign using the key can respend coins
  - Transactions are made between public keys
- How to verify transaction log integrity without central authority?
  - By majority vote using computing power
  - Requires active participation by honest majority
- How to get coins into the system?
  - Deterministic amount of money supplied through lottery
Transaction

Address is a hash of ECDSA public key

- One who can produce the signature can claim the money

- Every input must be unlocked by a signature
- Transaction is valid if signatures are valid and inputs unspent
- Difference between inputs and outputs is a transaction fee
Proof-of-work System

Hashcash:

- Challenge: find a $nonce$ such that first $x$ bits of $\text{hash}(\text{randomchallenge}||nonce)$ are zero bits.
- Solution requires brute force – $2^x$ tries on average
- Verification requires single hash operation $\text{hash}(\text{randomchallenge}||nonce) == "000000..."$?
- Non-interactive proof-of-work to fight spam
• Blocks are produced by miners who solve proof-of-work

• Chain with largest total difficulty is consensus chain
Miner collects all transaction fees
Miner earns 25 BTC “out of thin air”
  • Halved every 210’000 blocks (4 years)
Proof-of-work difficulty recalculated every 2016 blocks
  • To produce one block in 10 minutes
  • Difficulty cannot change more than by a factor of 4
  • Current difficulty – 65 bits
Bitcoin P2P Network

- Node listens on TCP port 8333
- Node connects to few other nodes
- Sends to other peers:
  - new transactions
  - new blocks
  - new peer addresses
  - blocks (on request)
  - block headers (on request)
  - peer addresses (on request)
- Node must not relay invalid blocks/transactions
- Node must implement DoS protection
Anonymity

- All transactions are public and traceable
- Transactions occur between public keys
- Backward security and forward security needed
- Solution: mixing services

- Zerocoin – complete anonymity using zero-knowledge proofs
  - proof size 40KB, 2 seconds to verify
Security Assumptions

- ECDSA scheme and SHA256/RIPEMD160 are secure
- Attacker does not control majority of the hashpower
  - Attacker could execute double-spending attacks
  - Attacker could destroy the network
  - Attacker could gain more by following the rules
  - Hashpower not uniformly distributed
    - Litecoin’s use of scrypt()
- Attacker cannot partition the network or isolate participants
  - Sybil attack
  - Forked chains cannot be merged
  - Profit by isolating other miners
Requirements

• Participants are able to store and verify transaction log
  • Transaction log size is 33 GB (excluding indexes)
  • Thin clients must trust power nodes
  • Transaction log pruning never implemented

• Participants are rational
  • Indirect incentive to keep network healthy

• No one can impose regulation
  • Regulation needed to fix security flaws
  • Changes without unanimous support will fork blockchain
    • Bitcoin software developers have an advantage here
  • Regulation needed to stop bitcoin thefts

Bitcoin security depends on a lot more than cryptography
Mining Pools

- Rewards shared proportionally to participants contribution
- Contribution proved by submitting lower difficulty solutions
- What prevents participants from cheating?
NOTICE OF EXTORTION

Your business, 900 Degrees Neapolitan Pizzeria, has been targeted for extortion. The selection process is random, and was not triggered by any event under your control.

Should you fail to pay the one-time monetary tribute, by the deadline provided below, your business will be severely and irreparably damaged. The following methods are commonly employed in cases of non-compliance:

- Negative Online Reviews
- BBB Complaints
- Harassing Telephone Calls
- Fraudulent Delivery Orders
- Telephone Denial-of-Service
- Bomb Threats
- Vandalism
- Mercury Contamination

Anonymous Reports of:
- Health Code Violations
- OSHA Violations
- Criminal Tax Evasion
- Money Laundering
- Illegal Drug Sales
- Marijuana Grow Operations
- Methamphetamine Production
- Terrorist Training Activity

The tribute price is only One Bitcoin (1 BTC), but must be paid by August 15, 2014. Payment is to be made to the Bitcoin Wallet Address listed below.

If payment is not received, our team will begin taking the actions listed above. Once engagement has begun, it can only be stopped for a tribute of Three Bitcoin (3 BTC). Because many of the actions we take are catastrophic and irreversible, is it advised to pay the tribute before the deadline is reached.

Payment Type: Bitcoin
Deadline: August 15, 2014

Amount Due: 1 Bitcoin
(If paid before deadline)

Amount Due: 3 Bitcoin
(If paid after deadline)

Purchase Bitcoin @
https://www.coinbase.com/

17gt1BancvtnnJwy4BA41VBUH3pfbUvzF
Task: Proof-of-work solver

Implement proof-of-work solving tool.

$ python pow.py --difficulty 26
[+] Solved in 296.456492 sec (0.2112 Mhash/sec)
[+] Input: 41726e6973205554000000003bb67af
[+] Solution: 00000031fc8ad63fa6070e341ccddd55bc36ac0b1e94965f2a8bb624d1a51071
[+] Nonce: 62613423

- Hash function – SHA256(SHA256())
- Input – your identity + 8 byte counter
- Difficulty – number of zero leftmost bits in the solution
- Provide your output for difficulty 26 in source code comments
  - Must push at least 0.1 Mhash/sec on today's hardware
  - Looking for fastest python implementation
- Verification of proof-of-work:

```python
>>> input = '41726e6973205554000000003bb67af'.decode('hex')
>>> input
'Arnis UT\x00\x00\x00\x00\x03\xbbg\xaf'

>>> hashlib.sha256(hashlib.sha256(input).digest()).hexdigest()
'00000031fc8ad63fa6070e341ccddd55bc36ac0b1e94965f2a8bb624d1a51071'
```
Exam

This was the last lecture!

Exam date:

- May 20, 2015, 14:15–16:00, Liivi 2-205
- Deadline for late homeworks
- Closed book exam
- Take your laptop with you
- Register on OIS

We might agree on a second exam date if you have a good reason why you cannot attend May 20 exam!
Task: Proof-of-work solver

<table>
<thead>
<tr>
<th>Nr</th>
<th>Student</th>
<th>Mhash/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>A20800</td>
<td>0.343</td>
</tr>
<tr>
<td>2.</td>
<td>A62103</td>
<td>0.287</td>
</tr>
<tr>
<td>3.</td>
<td>B45675</td>
<td>0.241</td>
</tr>
<tr>
<td>4.</td>
<td>B36761</td>
<td>0.230</td>
</tr>
<tr>
<td>5.</td>
<td>B46036</td>
<td>0.223</td>
</tr>
<tr>
<td>6.</td>
<td>B46044</td>
<td>0.221</td>
</tr>
<tr>
<td>7.</td>
<td>B11235</td>
<td>0.216</td>
</tr>
<tr>
<td>8.</td>
<td>B46871</td>
<td>0.216</td>
</tr>
<tr>
<td>9.</td>
<td>B46861</td>
<td>0.214</td>
</tr>
<tr>
<td>10.</td>
<td>B50783</td>
<td>0.211</td>
</tr>
<tr>
<td>11.</td>
<td>A</td>
<td>0.211</td>
</tr>
<tr>
<td>12.</td>
<td>B26600</td>
<td>0.200</td>
</tr>
<tr>
<td>13.</td>
<td>B50781</td>
<td>0.178</td>
</tr>
<tr>
<td>14.</td>
<td>B46864</td>
<td>0.157</td>
</tr>
<tr>
<td>15.</td>
<td>B50782</td>
<td>0.148</td>
</tr>
</tbody>
</table>

Table: Homework 14 performance top