MTAT.07.017
Applied Cryptography

Bitcoin

University of Tartu

Spring 2016
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 \textit{nonce} such that first \( x \) bits of \( hash(\text{randomchallenge}||\text{nonce}) \) are zero bits.
- Solution requires brute force – \( 2^x \) tries on average
- Verification requires single hash operation
  \[ hash(\text{randomchallenge}||\text{nonce}) == "000000..."? \]
- 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 – 71 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
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 69 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?
Opening Pandora’s Box

Extortion in Estonia

--- Forwarded message ---
From: DD4BC TEAM <[redacted]>
Date: Sat, Nov 1, 2014 at 4:57 AM
Subject: DDOS ATTACK!
To: [redacted]

Hello

Your site is extremely vulnerable to ddos attacks.

I want to offer you info how to properly setup your protection, so that you can’t be ddosed! My price is 1 Bitcoin only.

Right now I will start small (very small) attack which will not crash your server, but you should notice it in logs. Just check it.

I want to offer you info on how I did it and what you have to do to prevent it. If interested pay me 1 BTC to 17aLGgw8AwJdq1BtMMG1QtQJgNQQkYEsp

Thank you.

https://cybersec.ee/2015/05/14/two-estonian-companies-received-bitcoin-extortion-letters/
Task: Proof-of-work solver (3 points)

Implement proof-of-work solving tool.

$ python pow.py --difficulty 26
[+] Solved in 296.456492 sec (0.2112 Mhash/sec)
[+] Input: 41726e697320555400000000003bb67af
[+] 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:

  >>> input = '41726e697320555400000000003bb67af'.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:
- June 3, 2016, 12:15–14:00, Liivi 2-203
- Deadline for homework 14
- 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 June 3 exam!