MTAT.07.017
Applied Cryptography

Bitcoin

University of Tartu

Fall 2021
“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 can we do that without a trusted central authority?
Bitcoin

- How do we maintain the transaction log without a central authority?
  - Distribute to everyone over a peer-to-peer network

- How do we verify account holder’s intent without a central authority?
  - The account holder signs transactions using a digital signature

- How do we bind account holder’s identity to a public key?
  - The public key is an identity (account number) itself
  - Anyone who can sign using the key can spend the coins
  - Transactions are made between public keys

- How do we verify the integrity of the transaction log without a central authority?
  - By a majority vote using computing power
  - Requires active participation by the honest majority

- How do we get coins into the system?
  - Deterministic amount of money is supplied through a lottery
Transaction

- Address is a hash of an elliptic curve public key
- One who can produce a signature can claim the coins

- Every input must be unlocked by a signature
- The transaction is valid if the signatures are valid and the inputs are unspent
- The difference between inputs and outputs is a transaction fee
Proof-of-work system

Client

1. request service
2. choose
3. challenge
4. solve

Server

5. solution
6. verify
7. grant service

Hashcash:

- Challenge: find a nonce such that the first $x$ bits of $\text{hash}(\text{random\_challenge}||\text{nonce})$ are zero bits.
- Solution requires a brute force: $2^x$ tries on average
- Verification requires a single hash operation $\text{hash}(\text{random\_challenge}||\text{nonce}) == "000000..."$?
• Blocks are produced by miners who solve proof-of-work
• Miner earns 6.25 BTC “out of thin air”
  • Halved every 210 000 blocks (4 years)
• Miner collects all transaction fees
• The chain with the largest total difficulty is the consensus chain
• The proof-of-work difficulty is recalculated every 2016 blocks
  • To produce one block in 10 minutes
  • The difficulty cannot change more than by a factor of 4
  • The current difficulty – 76 bits
Peer-to-peer network

- Node listens on TCP port 8333
- Node connects to few other nodes
- Node advertises 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 denial-of-service protection
Anonymity

- All transactions are public and traceable
- Transactions occur between public keys
- Backward and forward privacy is needed
- Solution: mixing services
Security assumptions

- ECDSA scheme and SHA256/RIPEMD160 are secure
- Attacker does not control the majority of hashpower
  - Attacker could execute double-spending attacks
  - Attacker could destroy the network
  - Attacker could gain more by following the rules
  - Hashpower is not distributed uniformly
    - 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 the transaction log
  • Transaction log size is 344 GB (excluding indexes)
  • Thin clients must trust power nodes

• Participants are rational
  • Indirect incentive to keep the network healthy

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

  Bitcoin security depends on a lot more than cryptography
Opening Pandora’s box

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
- Increasing Telephone Calls
- Fraudulent Delivery Orders
- Telephone Denial-of-Service
- Rude Thieves
- 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, it is advised to pay the tribute before the deadline is reached.

Payment Type: Bitcoin
Deadline: August 15, 2014
Amount Due: 1 Bitcoin

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

17gt1BancvttnJwy4BA41VBUH3pfbUvzE
Questions

• How is transaction integrity and authenticity provided in Bitcoin?
• When seeing a Bitcoin blockchain, how can its authenticity be established?
• Why do Bitcoin miners solve blocks?
• Why can’t an attacker replace a transaction in a solved Bitcoin block?
• How can an adversary who has the majority of hashpower destroy the Bitcoin system?
• How can an account be opened in the Bitcoin system?
• Who has control over the Bitcoin system?
Task: Proof-of-work solver

Implement a proof-of-work solving tool:

$ ./pow.py --difficulty 26
[+] Solved in 248.347607 sec (0.1414 Mhash/sec)
[+] Input: 41726e69730000000217e8b4
[+] Solution: 000000092f7742b032c8495aa7ba0579af714402d6bb7b1c5c6ee21453c91764
[+] Nonce: 35121332

- Hash function – SHA256(SHA256())
- Input: your identity + 8-byte counter
- Difficulty: the number of zero leftmost bits in the solution
- Provide the output for the difficulty 26 in your solution’s source code
  - Must push at least 0.1 Mhash/sec on today’s hardware
  - Reject invalid candidates as early as possible

Verification of proof-of-work:

```python
>>> input = codecs.decode('41726e697320555400000003bb67af', 'hex')
>>> input
b'Arnis\00\00\00\00\02\17\xe8\xb4'

>>> hashlib.sha256(hashlib.sha256(input).digest()).hexdigest()
'000000092f7742b032c8495aa7ba0579af714402d6bb7b1c5c6ee21453c91764'
```