8271 discussion of: "Double Spending Fast Payments in Bitcoin"

Stephen McCamant (Original paper: Ghassan O. Karame, Elli Androulaki, and Srdjan Čapkun)
University of Minnesota (Original paper: NEC Labs and ETH Zurich)

Outline

Bitcoin background

Double spends and fast payments

Administrative reminders

Bitcoin addresses

- Address is basically a public/private signing key pair
  - Randomized naming, collision unlikely
- At any moment, balance is a perhaps fraction number of bitcoins (BTC)
- Anyone one can send to an address, private key needed to spend

Global transaction log

- Basic transaction: Take x_1 from a_1, x_2 from a_2, ..., put y_1 in a'_1, y_2 in a'_2, ...
  - Of course require $\sum_i x_i = \sum_j y_j$
- Keep one big list of all transactions ever
- Check all balances in addresses taken from are sufficient

Bitcoin network

- Use peer-to-peer network to distribute transaction log
- Roughly similar to BitTorrent, etc. for old data
- Once a client is in sync, only updates need to be sent
- New transactions sent broadcast

Consistency and double-spending

- If all clients always saw the same log, double-spending would be impossible
- But how to ensure consistency, if multiple clients update at once?
- Symmetric situation: me and "me" in Australia both try to spend the same $100 at the same time

Bitcoin blocks

- Group ~10 minutes of latest transactions into one "block"
- Use a proof of work so creating a block is very hard
- All clients race, winning block propagates

Bitcoin blockchains

- Each block contains a pointer to the previous one
- Clients prefer the longest chain they know
- E.g., inconsistency usually resolved by next block

Regulating difficulty

- Difficulty of the proof-of-work is adjusted to target the 10 minute block frequency
- Recomputed over two-week (2016 block) average
- Network adjusts to amount of computing power available

Bitcoin mining

- Where do bitcoins come from originally?
- Fixed number created per block, assigned by the client that made it
- Incentive to compete in the block generation race
- Called *mining* by analogy with gold

Enforcing consistency

- Structure of network very resistant to protocol change
  - Inertia of everybody else's code
- Changes unpopular among miners will not stick
- Minor crisis in March 2013: details of database lock allocation cause half of network to reject large block

Outline

Bitcoin background

Double spends and fast payments

Administrative reminders

Fast payments

- You'd like to use Bitcoin for instantaneous transactions:
  - In-person snack machine, grocery store, etc.
  - Pure digital goods like MP3s, e-books, etc.
  - ATM withdrawal, currency conversion
- But no possibility of reversing transactions later
  - So need strong protection against fraud

Reception vs. confirmation

- *Reception*: transaction propagated through P2P network
  - Average about 3 seconds
- *Confirmation*: transaction incorporated in block chain
  - Average 10 minutes per block
- Conservative 6 confirmations: 1 hour, mail-order speed

Block generation time distribution

- Expected: exponential with $\lambda = 10$ min
- Paper's analysis basically confirms this with extra complexities
  - Good fit between 2-minute binned results and shifted geometric distribution with $p = 0.19$ (vs. expected $2/10$)

Basic double-spend attack

- Attacker A, victim (e.g., vendor) V
- Two transactions TR_V and TR_A spend the same coins
- Attacker wins if TR_V accepted by vendor, but TR_A ends up in block chain
- Send TR_V to vendor, "helpers" introduce TR_A elsewhere

Analyzing difficulty

- Nodes store whichever transaction they see first, ignore other
- Easy to ensure vendor sees TR_V first
- Which TR appears in the next block proportional to how many nodes keep it

Experimental difficulty

- Ten geographically-distributed test nodes
  - Vary vendor location, randomly choose 1-2 helpers
- Vary relative time of two introductions
- Many configurations had 100% success rate

Countermeasure: listening period

- Idea: after receiving transaction, wait 15 s to see a double spend
  - Proposed in a Bitcoin FAQ
- Attacker has tradeoff between probabilities of detection and success
- Attractive attacks (5–30% success) possible
  - Especially when vendor has few connections

CM: network observers

- Recruit extra nodes to listen for double spends
- In experiments with 5 observers, all double-spends were seen within a few seconds
- Authors recommend at least 3 observers, arguably expensive

CM: forwarding double-spends

- Authors propose: always forward transactions that appear to be double spends
  - But do not use for block generation
- Affects only detection, not attack success
- Possible problems: load, DoS
- Not deployed as far as I know

Outline

Bitcoin background

Double spends and fast payments

Administrative reminders

Last call for Zerocoin

- If anyone besides me wants to present the "Zerocoin" paper for Wednesday, now is your last chance to volunteer

More papers posted

- New potential papers posted for:
  - Infrastructure paranoia
  - Smartphone security
  - Tor
  - (Anti-)censorship

Topic popularity survey

- By Tuesday night, email me your list of the topic areas from the web page, sorted by order of your interest
- Mentioning specific papers is optional