CSci 5271
Introduction to Computer Security
Day 25: Electronic cash and Bitcoin

Stephen McCamant
University of Minnesota, Computer Science & Engineering

Outline

End-to-end integrity and verification

- Tabulation cannot be 100% public
- But how can we still have confidence in it?
- Cryptography to the rescue, maybe
  - Techniques from privacy systems, others
  - Adoption requires to be very usable

Commitment to values

- Two phases: commit, later open
  - Another analogy to a use of envelopes
- Binding property: can only commit to a single value
- Hiding property: value not revealed until opened
- Trivia: either binding or hiding, but not both, can be perfect
  - Information-theoretic, like a one-time pad

Randomized auditing

- How can I prove what's in the envelope without opening it?
- n envelopes, you pick one and open the rest
  - Chance $1/n$ of successful cheating
- Better protection with repetition

Election mix-nets

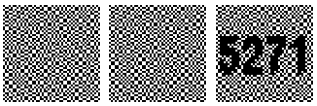
- Independent election authorities similar to remailers or Tor nodes
- Onion-encrypt ballot, each authority shuffles and decrypts
- Extra twist: prove no ballots added or removed, without revealing permutation
  - Instance of "zero-knowledge proof"
- Privacy preserved as long as at least one authority is honest

Pattern voting attack

- Widely applicable against techniques that reveal whole (anonymized) ballots)
- Even a single race, if choices have enough entropy
  - 3-choice IRV with 35 candidates: 15 bits
- Buyer says: vote first for Bob, then 2nd and 3rd for Kenny and Xavier
  - Chosen so ballot is unique

Fun tricks with paper: visual crypto

- Want to avoid trusted client, but voters can't do computations by hand
- Analogues to crypto primitives using physical objects
- One-time pad using transparencies:



Scantegrity II

- Designed as end-to-end add-on to optical scan system
- Fun with paper 2: invisible ink
- Single trusted shuffle
  - Checked by random audits of commitments
- Version used in a DC-suburb municipal election

Outline

Cryptography for voting

Previous e-cash and techniques

Announcements intermission

Bitcoin design

Bitcoin experience

Kinds of Internet payments

- Credit/debit cards: most popular
  - Wide adoption among consumers, little consumer fraud liability
  - Restrictive merchant procedures
- PayPal
  - Easier to accept payments
  - Centrally managed to deal with fraud

Ideal: electronic cash

- Direct transactions without third party
- No transaction fees
- Potentially anonymous
- Non-revocable: buyer bears fraud risk

Micropayments

- Claim: what the web needs is small payments to support content
  - Too small for existing mechanisms
- One idea (Peppercoin): simulate small payment with small probability of larger payment
- Actual market for micropayments has been small
  - Most buyers and sellers prefer free + other revenue

Blinded signatures

- Sign something without knowing its value
  - Often used together with randomized auditing
  - For RSA, multiple message by r^e, r random
- Allows a bank to "mint" coins that can still be anonymous

Challenge: double spending

- Any purely electronic data can be duplicated, including electronic money
- Can't allow two copies to both be spent
- Shows ideal no-third-party e-cash can't be possible

Puzzles / proof-of-work

- Computational problem you solve to show you spent some effort
- Common: choose s so that $h(m \| s)$ starts with many 0 bits
- For instance, required solved puzzles can be a countermeasure against DoS

Hashcash and spam

- Idea: use proof of work to solve email spam problem
- Puzzle based on date and recipient
- Legitimate users send only a few messages
  - Problem 1: mailing lists
  - Problem 2: spam botnets
- Never caught on

Hash trees and timestamp services

- Merkle tree: parent node includes hash of children
- Good hash function $\rightarrow$ root determines whole tree
- Can prove value of leaf with log-sized evidence
- Application: document timestamping (commitment) service

Outline

Cryptography for voting

Previous e-cash and techniques

Announcements intermission

Bitcoin design

Bitcoin experience

HW2 due Sunday

- Non-early due date: 11:55pm this Sunday
- Q5 performance/load issues
  - Avoid by not doing Q5 at the last minute, testing on yourself

Group project presentations

- Start Monday, run next two weeks
- Plan 12 minute presentation plus 3 minutes Q&A
- One student per group presents
- Slides, BYO laptop recommended

Project progress reports Monday

- Due Monday 11:55pm
- Progress meetings next week will mostly be after
- Email to start the conversation early

Outline

Cryptography for voting

Previous e-cash and techniques

Announcements intermission

Bitcoin design

Bitcoin experience

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

Outline

Cryptography for voting

Previous e-cash and techniques

Announcements intermission

Bitcoin design

Bitcoin experience

Where Bitcoin came from

- Paper and early implementation by Satoshi Nakamoto
  - Generally presumed to be a pseudonym
- "Genesis block" created January 2009
  - Containing headline from The Times (of London) about a bank bailout

Current statistics

- Block chain 271,000 blocks, about 14GB
- 12M BTC minted (many presumed lost)
- Theoretical value at market exchange rate > $1 billion
- Millions of addresses, probably many fewer users
- Mining power: 5 petahash/sec

What can you buy with Bitcoin?

- Random stuff from many small online retailers
- Novelty/trials of some in-person purchases
- Donations to like-minded non-profits
- Illegal drugs (Silk Road successors)
- Murder for hire: currently probably a fraud

Bitcoin as a currency

- Can be exchanged for dollars, etc.
  - Currently pretty cumbersome
- In some ways more like gold than fiat currencies
  - No central authority
  - Price changes driven more by demand than supply
- Exchange rate trend: volatile but upward

Deflation and speculation

- Some people want bitcoins to spend on purchases
  - Demand based on "velocity"
  - Supply does not keep up with interest
  - So, value of 1 BTC has to go up
- Others want bitcoins because they think the price will go up in the future
  - Self-fulfilling prophecy
  - But vulnerable to steep drops if expectations change

Bitcoin mining trends

- Exponentially increasing rates
- CPU $\longrightarrow$ GPU $\longrightarrow$ FPGA $\longrightarrow$ ASIC
- Specialized hardware eclipsing general purpose
  - Including malware and botnets
- Recent price trends suggest continuing investment

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: details of database lock allocation cause half of network to reject large block

Stealing bitcoins

- Bitcoins are a very tempting target for malware
  - Private keys stored directly on client machines
  - Theft is non-reversible
  - Much easier than PayPal or identity theft
- Standard recommendation is to keep keys mostly offline

Bitcoin (non-)anonymity

- Bitcoin addresses are not directly tied to any other identity
- But the block chain is public, so there's lots of information
  - List of largest balances on Wikipedia, academic research
  - `http://eprint.iacr.org/2013/782`
- Real unlinkability is a research topic

Next time

- Group project presentations