CSci 5271 Introduction to Computer Security Day 26: Electronic cash and Bitcoin

Stephen McCamant University of Minnesota, Computer Science & Engineering

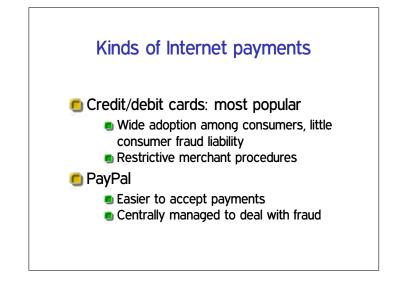
Outline

Previous e-cash and techniques

Bitcoin design

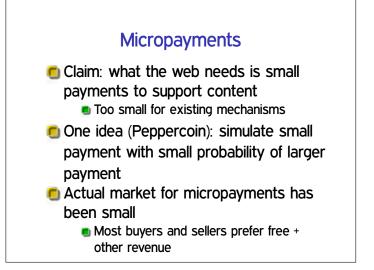
Announcements

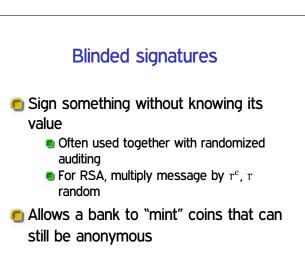
Bitcoin experience



One ideal: electronic cash

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



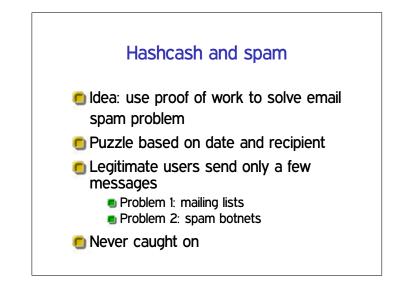


Challenge: double spending

- 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 $h(m \parallel s)$ starts with many 0 bits
- For instance, required solved puzzles can be a countermeasure against DoS



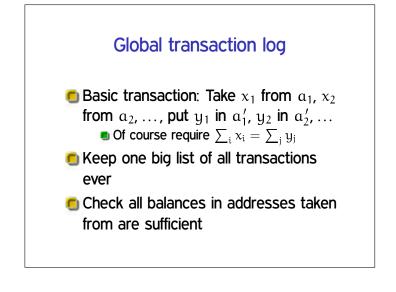
Hash trees and timestamp services

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



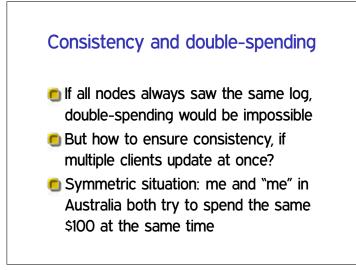
Bitcoin addresses

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



Bitcoin network

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



Bitcoin blocks

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



- Each block contains a pointer to the previous one
- Nodes 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 node that made it
- An incentive to compete in the block generation race
- Called *mining* by analogy with gold

Outline

Previous e-cash and techniques

Bitcoin design

Announcements

Bitcoin experience

Group project presentations

- Start next Wednesday, run three lectures
- Plan 10 minute presentation plus say 3 minutes Q&A
- One student per group presents
- Slides, BYO laptop recommended
 - Can send me backup slides (PDF, PPT) night before

Wednesday presentations

1:00 - 1:13 JS API checking (Q) 1:14 - 1:25 Password models (LMS) 1:26 - 1:39 Reading CAPTCHAs (NORR) 1:40 - 1:45 announcements 1:46 - 1:59 Evil-twin WiFi (CNQT) 2:00 - 2:13 Password managers (DEK)

December dates

- Final project progress reports due tonight
- Exercise set 5 due Tuesday 12/12
- Project final reports due Wednesday 12/13

Outline

Previous e-cash and techniques

Bitcoin design

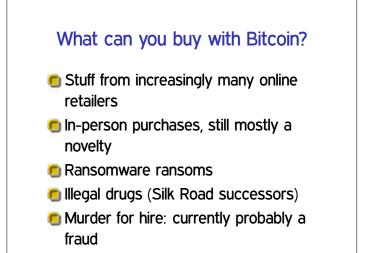
Announcements

Bitcoin experience

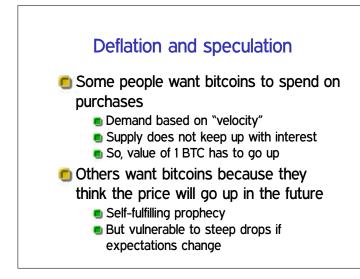


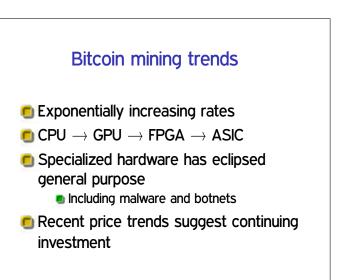
Current statistics

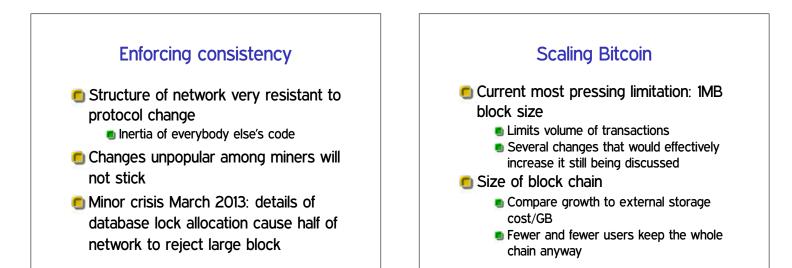
- Block chain 497,498 blocks, ~154GB
- 16.7M BTC minted (many presumed lost)
- Theoretical value at market exchange rate > \$184 billion
- Millions of addresses, probably many fewer users
- 🖲 Mining power: 11 etahash/sec

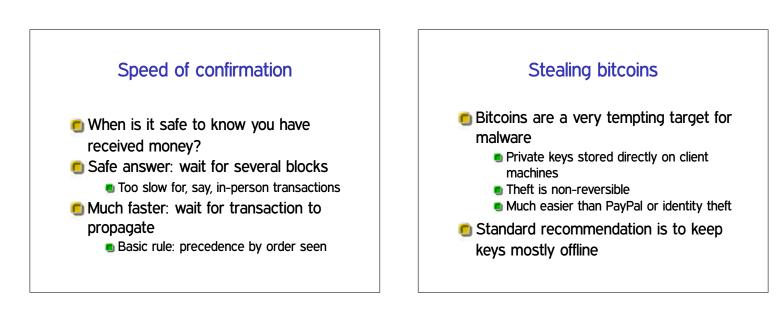


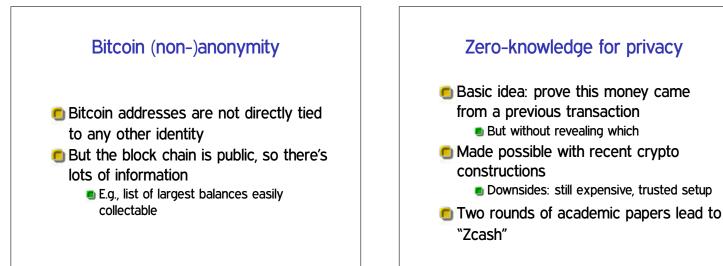


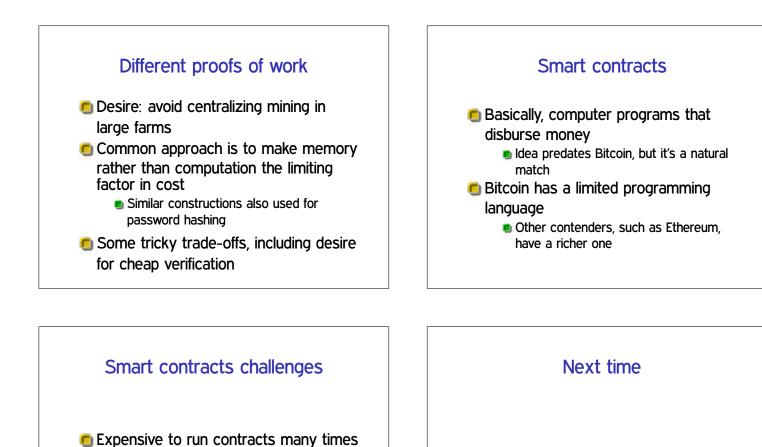












- **–** C
- Code visible, but bugs can't be fixed

(e.g., during mining)

Hack of high-profile Ethereum "DAO" application lead to a community fork

