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

One 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, multiply 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 \( h(m \mid s) \) 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 → root determines whole tree
- Can prove value of leaf with log-sized evidence
- Application: document timestamping (commitment) service

Outline

- Previous e-cash and techniques
- Bitcoin design
- Announcements
- Bitcoin experience

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 node is in sync, only updates need to be sent
- New transactions sent broadcast

Consistency and double-spending

- If all nodes 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 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 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
Bitcoin blockchains
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

Project progress report #3
Due tonight
Should also include sample of report formatting
Content can be draft or just progress material
Choice of MS Word or LaTeX
NB, format is more dense than typical class report

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
Let me know if you’d prefer a remote Zoom presentation

Hands-on assignment status
Current target for VM availability is late tonight
PDF instructions updated with .4 monitoring interface
VM-specific detailed instructions posted
Many groups still haven’t registered

Exercise set 4
Also targeting late tonight for release
Questions covering the last few course topics
Compensating for late releases

- Staggered due dates planned: 12/8 for Ex. 4, 12/13 for HA, 12/15 for project final report
- Prefer not to extend, lest all the due dates pile up
- Instead, reduce weighting:
  - For each day HA is late (2 already), 1% of HA weight (out of 15) replaced with automatic 100%
  - For each day Ex. 4 is late, 1% of Ex. weight (out of 10, max 4) replaced with automatic 100%

Outline

- Previous e-cash and techniques
- Bitcoin design
- Announcements
- 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

Example statistics (Dec. 2021)

- Block chain 712,038 blocks, ~438GB
- 18.9M BTC minted (many presumed lost)
- Theoretical value at market exchange rate > $1,072 billion
- > 30 million addresses, probably many fewer users
- Mining power: 150 etahash/sec

What can you buy with Bitcoin?

- Stuff from increasingly many online retailers
- In-person purchases, still mostly a novelty
- Ransomware ransoms
- 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, recently up

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 → GPU → FPGA → ASIC
- Specialized hardware has eclipsed 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 March 2013: details of database lock allocation cause half of network to reject large block

Scaling Bitcoin
- Most-discussed limitation: block size
  - Long limited to 1MB, currently more like 2MB
  - Limits volume of transactions
- Trade-offs affect transaction fees and network size
- Size of block chain
  - Compare growth to external storage cost/GB
  - Fewer and fewer users keep the whole chain anyway

Speed of confirmation
- When is it safe to know you have received money?
- Safe answer: wait for several blocks
  - Too slow for, say, in-person transactions
- Much faster: wait for transaction to propagate
  - Basic rule: precedence by order seen

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
  - E.g., list of largest balances easily collectable

Zero-knowledge for privacy
- Basic idea: prove this money came from a previous transaction
  - But without revealing which
- Made possible with recent crypto constructions
  - Downsides: still expensive, trusted setup
- Two rounds of academic papers lead to "Zcash"

Different proofs of work
- Desire: avoid centralizing mining in large farms
- Common approach is to make memory rather than computation the limiting factor in cost
  - Similar constructions also used for password hashing
- Some tricky trade-offs, including desire for cheap verification

Smart contracts
- Basically, computer programs that disburse money
  - Idea predates Bitcoin, but it's a natural match
- Bitcoin has a limited programming language
  - Other contenders, such as Ethereum, have a richer one
Smart contracts challenges

- Expensive to run contracts many times (e.g., during mining)
- Code visible, but bugs can't be fixed
  - Hack of high-profile Ethereum "DAO" application lead to a community fork