Bitcoin

CS 161: Computer Security

Prof. David Wagner

April 15, 2016

Distributed Logging

- Let’s do distributed peer-to-peer logging of public data. We have *n* computers; they all know each others’ public keys. Any computer can broadcast to all others (instantaneously, reliably). Any computer should be able to append a signed entry to the log, and to verify integrity of any previous log entry.
- Security goal: Malicious computers should not be able to back-date entries or modify past log entries. Assume ≤ 3 computers are malicious.

- **Problem 1.** Describe a protocol for this. What does Alice do to append an entry? What do other computers need to do?

Your Solution

- To append log entry e:
- Other computers should:

Distributed Logging

- **Problem 2.** Let’s generalize. Suppose *m* of the *n* computers are malicious. If we make the obvious change to your protocol, for which *m* can it be made secure?

- (a): for all m < n.
- (b): for all m < n/2.
- (c): for all m < n/3.
- (d): for all m < √n.
- (e): for all m < O(lg n).

Distributed Logging

- **Problem 2.** Let's generalize. Suppose *m* of the *n* computers are malicious. If we make the obvious change to your protocol, for which *m* can it be made secure?

- (a): for all m < n.
- **(b): for all m < n/2.**
- (c): for all m < n/3.
- (d): for all m < √n.
- (e): for all m < O(lg n).

Distributed Money

- Donna gets the brilliant idea to use this log to store financial transactions. Each person’s initial balance is public.
- To transfer $10 from Alice to Bob, Alice appends a signed log entry saying “I transfer $10 to Bob” and broadcasts it. Everyone can compute the updated balance for Alice and Bob.

- **Problem 3.** What are some ways that a malicious actor might try to attack this scheme? Is this a good scheme?

Your Answers

- Replay
- Denial of service attacks
- Broadcast doesn’t scale
- TOCTTOU vulnerability

Problems with This Scheme

- Initial balance is arbitrary
- Broadcasting is expensive and doesn’t scale
- A conspiracy of *n*/2 malicious computers can fork the audit log and steal all the money
- Sybil attacks: Anyone can set up millions of servers and thus have a 50% majority

Problems with Naïve Scheme

- Transactions aren’t authenticated
- Double-spending
- Synchronization: Not clear how to resolve inconsistencies
- Sybil attacks: Anyone can set up millions of servers and thus take over most of the network; then they can steal all the money
- Graph cut: If all nodes you’re connected to are malicious, they can lie to you (eclipse attack)

Idea #1: Transactions are signed

- Alice signs transaction paying money to Bob
- Technical trick: money is represented by “coins”; Alice can pay Bob by transferring ownership of the coin to Bob, which she does by publishing
    “I give coin c to Bob”, Signature
- Everyone can validate this by checking transfer is signed by current owner of coin
- Technical trick: use public keys to identify users, instead of names or accounts

Idea #2: Linearize

- To prevent double-spending, “linearize” history: Public log has a sequence of transactions.
- Only current owner of coin can transfer it to someone else, so there’s no ambiguity about who owns a coin

Idea #3: Hashchain

- To prevent retroactively changing history, store transaction log in an hash chain.
- Hash chain is public, broadcasted on peer-to-peer network, and append-only: honest nodes will reject any broadcasts that do anything other than append to the log.

- (Otherwise, Alice could append “I give coin c to Amazon”, get her book from Amazon, then undo and change that to “I give coin c to Barnes and Noble”.)

Idea #4: “Longest chain” wins

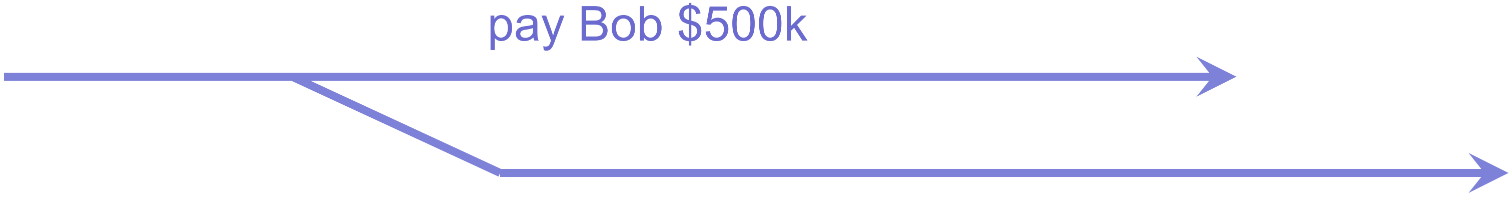
- Problem: Consistency

- What if two different parts of network have different hash chains?

- Solution: Whichever is “longer” wins; the other is discarded

Problem: Consensus

- Problem: Mallory can fork the hash chain
- Say she buys Bob's house from him for $500K in Bitcoins.   Then, she goes back in time and, starting from the block chain just before this transaction was added to it, she starts appending new entries from there. Can she get others to accept this forked chain, so she gets her $500K back? Yes.



Idea #5: Reward miners

- Each item appended to hash chain must be a proof of work (its hash must start with 33 zero bits)
- Give a reward to anyone who successfully appends – they receive a free coin

Bitcoin

- Public, distributed, peer-to-peer audit log of all transactions.
- To append an entry to the log, the latest value must hash to something whose first 33 bits are zero; then broadcast it to everyone.
- Anyone who appends an entry to the log is given a small reward, in new money (a fraction of a Bitcoin).

Bitcoin

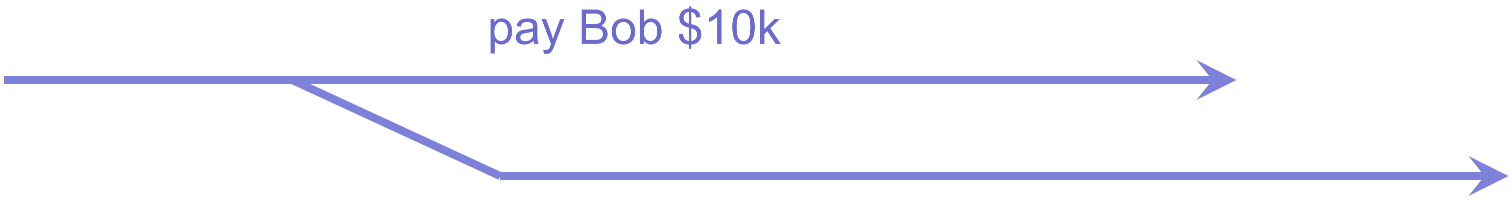
- Public, distributed, peer-to-peer, hash-chained audit log of all transactions (“block chain”).
- Mining: Each entry in block chain must come with a proof of work (its hash value starts with *k* zeros). Thus, appending takes computation.
- Lottery: First to successfully append to block chain gets a small reward (if append is accepted by others). This creates new money. Each block contains a list of transactions, and identity of miner (who receives the reward).
- Consensus: If there are multiple versions of the block chain, longest one wins.

Bitcoin

- Transactions: If Alice wants to give $10 to Bob, she signs this transaction. She gives the signed transaction to all miners and asks them to include it in the block they're trying to append to the chain.
- Honest miners check integrity of block chain entries and try to append to the latest, longest valid version of block chain.
- Bob knows he has received $10 once this transaction appears in the consensus block chain.

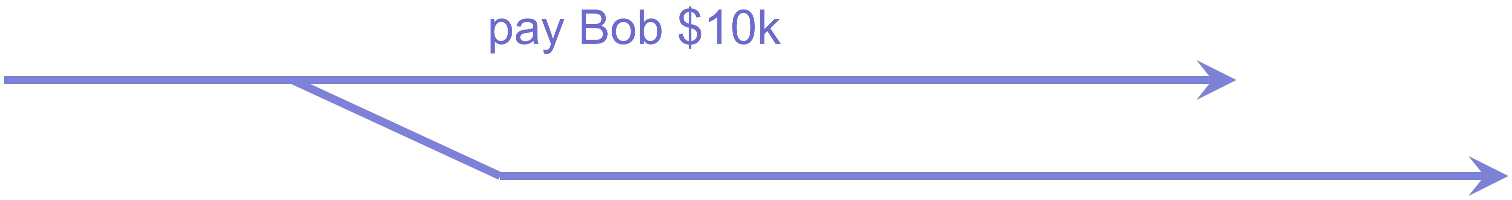
Consensus

- Can Mallory fork the block chain?
- Say she buys Bob's from him for $10,000 in Bitcoins. Then, she goes back in time and, starting from the block chain just before this transaction was added to it, she starts appending new entries from there. Can she get others to accept this forked chain, so she gets her $10,000 back?



Consensus

- Can Mallory fork the block chain?
- Answer: No, not unless she has ≥51% of the computing power in the world. Longest chain wins, and her forked one will be shorter (unless she can mine new entries faster than aggregate mining power of everyone else in the world).



How Bitcoin Addresses Criticisms of Naïve Scheme

- *Initial balance is arbitrary*: in Bitcoin, initial balances are zero
- *Broadcasting is expensive and doesn’t scale:* gossip protocol
- *A conspiracy of n/2 malicious computers can fork the audit log and steal all the money:* they’d have to own 51% of all the computing power in the Bitcoin world
- *Sybil attacks: Anyone can set up millions of servers and thus have a 50% majority:* they’d have to own 51% of all the computing power in the Bitcoin world

Discussion

- How can Alice turn dollars into bitcoins, or vice versa?
- Is Bitcoin anonymous?
- Should I think of Bitcoin as a short-term currency or as a long-term investment?
- Is it ethical to build a system that relies upon wasting CPU cycles (and thus energy)?

Bitcoin Take-away

- Crypto tools allow for sophisticated solutions to integrity and trust in peer-to-peer systems