6.033: Security - Bitcoin
Lecture 24
Katrina LaCurts, lacurts@mit.edu

```
**************************************************************************
* Disclaimer: This is part of the security section in 6.033. Only        *
* use the information you learn in this portion of the class to          *
* secure your own systems, not to attack others.                         *
**************************************************************************
```

0. Today
   - Bitcoin! Deals with anonymity, which we haven't considered yet.

1. Distributed Public Logs
   - Most of this lecture is about the blockchain, the data structure that underlies Bitcoin.
   - The blockchain provides a distributed public log (or ledger) that multiple users can read to/write from.
   - Basics: users append their transactions to the log, everyone in the system has a copy of the log, totally decentralized so no part of the system is more powerful than any other.
   - Challenge: anonymity; Solution: users identified only by their public keys.
   - Challenge: prevent adversaries from editing transactions (e.g., causing me to give Howie $100 instead of $10); Solution: users sign their transactions.
   - Challenge: prevent adversaries from duplicating, removing, reordering transactions; Solution: hash chains.
     - Each block includes the hash of the block before it in the chain.

2. Consensus
   - With multiple users writing to the log, how do we know which log is "correct"? How do we confirm a transaction?
     - Alternate motivation: Eve wants to "double-spend" a coin
   - Idea: Raft? No; we don't want a leader
   - Idea: Broadcast transactions, wait for majority of network to confirm them before appending to log. No: thwarted by Sybil Attacks.
     - Eve creates a lot of identities, confirms anything she wants
   - Idea: If a user receives two (or more) copies of the log, and they conflict in some way, only keep (and forward) the one that is longest. A transaction is confirmed when it appears in the log with N blocks after it.
   - Not bad, but Eve can still thwart this with a Sybil Attack.
   - Underlying issue: it's *easy* to valid blocks; calculating hashes is easy.

3. Proofs of Work (one way to handle this issue; NOT the only way)
   - Idea: make it computationally expensive to validate a block

- Bitcoin's mechanism:
  - A user receives the log with an unvalidated transaction pending. Checks that that transaction is legit (e.g., if the transaction is A transferring coin X to B, need to check via prior log entries that A indeed owns coin X). Then begins to solve puzzle:
    - t = pending block
    - Find nonce such that H(t|nonce) < target, where H is the hash function, target is set by the system (can change)
    - In practice, target is tuned so that nonce takes ~10min to find
   - Once a user finds a valid nonce, they publish it as part of the block, and broadcast the solution
  - V can't mount Sybil attack now; would need to own a lot of computational resources, not just a lot of identities

4. Vocabulary
   - The log is the blockchain
   - In the world of Bitcoin, validating blocks is the process of mining
     - There is incentive to do this; users get paid (in Bitcoin) for successfully mined blocks
   - If two users validate the same block at (roughly) the same time, we say the blockchain forks

5. Discussion
   - Performance
     - It takes time and energy (power) to validate blocks
       - In reality, each block contains multiple transactions
     - It takes time to get a transaction confirmed
     - Downloading the entire blockchain at the start of using Bitcoin can take a very long time
     - Proofs of work waste computation, and normal users are unlikely to mine successfully
   - Lots of other uses for blockchains, not just digital currency
   - Are users truly anonymous?