6.033: Security - Underground Web Technologies
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.                         *
**************************************************************************
```

An invaluable resource for Bitcoin is this blog post:
http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

0. Introduction
   - We've covered how to provide confidentiality, integrity, and authenticity
   - Today we're talking about anonymity
   - Focus for next two lectures: Bitcoin and Tor
     - Bitcoin: digitial currency system, which (possibly) provides anonymity
     - Tor: network for users to remain anonymous
     - Both deal with interesting technical problems
     - Both solve problems using things we've taught you (public keys, signatures, etc.)
     - Very popular as of late
   - You'll see some threat models we haven't considered yet
   - So today: Bitcoin; next lecture: Tor

1. Currency
   - What is money?
     - A medium for exchange. It's not valuable for itself, but for future exchanges
     - A store of value.
   - Good things about physical money
     - Portable
     - Semi-anonymous (modulo, e.g., tracking serial numbers)
     - Can't double-spend: I can't spend the same bill in two places, since once I spend the bill it's gone from my possession
     - Can't "repudiate" after payment: Once I pay someone with bills, I can't lie and say "Oh no I didn't really pay that person"
     - No need for trusted third-party to make a transaction. You just hand the bill to the other person
     - Government can print more as the economy expands or conditions dictate
   - Neutral thing about physical money
     - Difficult to tax/monitor transactions (good or bad depending on your goals)
   - Bad things about physical money
     - Easy to steal

  - Doesn't work online
- Good things about electronic money (credit cards, paypal)
  - Works online
  - Harder to steal (sometimes)
- Bad things about electronic money
  - No privacy

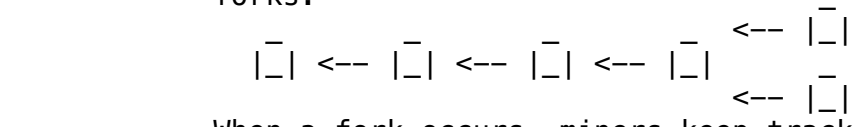
2. A Decentralized Digital Currency (Almost)
- Goal for right now: try to build a digital currency that doesn't require a trusted third party (a bank)
- Need to address
  - Forgery (can't generate money you don't have)
  - Double-spending (can't spend the same money twice)
  - Theft
- Idea 1
  - Every time Alice wants to give Bob a coin, she simply sends him a message that says "I, Alice, am giving Bob one coin."
  - Problem: Forgery! Anyone could forge that message.
- Idea 2:
  - Sign those messages. Prevents forgery
  - Problem: Duplicates. If Alice sends ten messages like the one above, did she give Bob ten coins? Were some retransmission?
- Idea 3:
  - Add sequence numbers. It's money, so let's call those "serial numbers" instead.
- Still to solve
  - Keeping track of who owns which coins
  - Assigning new serial numbers
  - Verifying that a particular coin hasn't already been spent
- Those unsolved problems are the things banks normally deal with

3. Dealing with Double-spending
- Problem: Alice tries to send the same coin to Bob and Charlie, i.e., to spend the coin twice.
- Idea: Publish all transactions in some sort of public log. Instead of having a centralized bank keep track of all transactions, we'll let *everybody* keep track of all transactions.
  - Need a way to secure the log, and to make sure everyone has the correct version of the log
- Sketch of this idea
  - If Alice sends a coin to Bob, he publishes that transaction and alerts everyone
  - If Alice later tries to send the same coin to Charlie, all parties in the network will know that something is amiss: they'll see that Alice already sent that coin to Bob
- Problem: What if Alice sent the coin to Bob and Charlie at the time, effectively spending it with both parties before either has had a chance to publish the transaction.
- One idea: get consensus. Agreement from "enough people" as to

which transaction is valid (either the one with Bob or the one with Charlie).
   - Instead of "enough people", let's say, specifically, "more than 50% of the network". There are at least two problems with this:
     1. We might not know how many people are in the network.
     2. Even if we know how many people are in the network, Alice could create multiple identities herself. This is known as a Sybil Attack: a user subverts a system by creating multiple identities

4. Proofs-of-work
   - Proofs-of-work thwart Sybil Attacks. Make it computationally expensive to perform an action. Then, multiple identities don't help. You need a lot of computational power to perform a Sybil Attack, which is much harder to get.
   - Bitcoin's proof of work:
     - A user K broadcasts the following message to users on the network: "I, K, want to spend coin 198601 with Bob"
     - Users who hear that message add it to their queue of pending transactions. Their goal is to verify a block of those transactions
     - After a user checks that the block is valid - that everyone owns the coins they're trying to spend - they set about solving the following "puzzle":
       - t = block of transactions (as a bitstring)
       - Find x such that H(t|x) < target
       - Target changes frequently. Adjusted so that it takes roughly ten minutes to solve this puzzle.
     - When a user solves the puzzle (finds x), they broadcast the block out along with x. They receive a monetary reward for solving the proof of work, to motivate users to take part in this process.
     - This process is known as "mining" bitcoins
   - What if two people solve the puzzle at the same time? We'll save that problem for a second.

5. Ordering Transactions
   - How does validating with the log work?
   - Log contains verified transaction blocks *and* a pointer to the previous block. This provides an ordering on the transactions, so we know who owns what coin.
     - The "pointer" is a hash of the previous block
     - The log is actually called the blockchain
   - If two people solve the puzzle at the same time, the blockchain forks:



   - When a fork occurs, miners keep track of both forks and work to

extend the longest fork. Pretty quickly, the shorter fork is rendered invalid.
   - Formally: a transaction is not confirmed until it is part of a block on the longest fork and at least five blocks follow it in the longest fork.

6. Did we prevent double-spending?
   - Can Alice validate a block that includes a transaction with the same coin at two different people? No, the rest of the network will see something is amiss when they get that block.
   - Can Alice try to spend the same coin twice with Bob and Charlie and get the network to verify it? No. At best, the blockchain will fork, but eventually only one fork (and so one transaction) will be validated.
   - Can Alice spend the same coin with Charlie and herself (as a Sybil)? Send the coin to Charlie, wait for the transaction to be entirely confirmed, then have her Sybil fork the chain prior to the transaction with Charlie, and "catch up" with the system so that the new branch becomes longer? (Effectively invalidating the previous transaction, but Alice already spent that coin) Unlikely -- other miners won't help, so she'd need a lot of compute power.

7. Discussion
   - Bitcoin provides you with a distributed public ledger. You can build other things on top of that, not just a digital currency.
   - Possible criticisms:
     - Proofs-of-work waste computation
     - In practice, users are either "normal" users who spend but don't mine, or mining pools who mine but don't spend.
     - It can take a long time to confirm a transaction
   - Alternatives exist for solving some of these problems
     - Ethereum and Algorand are two
   - Are users actually anonymous?