March 10 - Bitcoin

Lecture Discussion

Main pieces of the Bitcoin protocol

- The Ledger:
  - An append-only log of blocks where each block has a bunch of transactions. Each transaction has an amount and transfers money from one public key to another.
  - Anyone who does a linear scan over this ledger can keep track of the balances of all the accounts effectively. This is enough to enable you to figure out whether a new transaction is valid.
- The network layer:
  - A peer-to-peer system where whenever you have a transaction that you would like added to the blockchain, you broadcast it on this peer-to-peer system. So you broadcast it to everyone else.
  - When you receive a block, check if it is all valid. If yes, then you accept and redistribute it; if not, then just discard.

Assuming most of the people on this system are honest, and proof-of-work is not considered, is this mechanism enough for security?

- No, it is vulnerable to Sybil attack, where I sign up for this network a million times and become a million nodes in this peer-to-peer system. And I now have the overwhelming majority of nodes. Then, I could spread this to basically everyone.
- Another issue is even without malicious hosts. There are two people who are adding a block at the same time, Alice and Bob. They're both totally honest with a valid block. Half of the nodes believe in Alice and the other half believe in Bob. And now we have this situation where people don't agree. How does this network ever come to consensus?

Bitcoin prevents Sybil attack

- Proof of work: Bitcoin has one vote per CPU, so it really expensive to try to solve a million proof-of-work puzzles, so malicious attackers cannot conduct Sybil attack without a prohibitive (over 50% of the entire compute power in the chain) cost.

Bitcoin and proof-of-work:

- Energy consumption due to blockchain mining
  - It is a waste of energy to do blockchain mining, where people burn their compute power in something completely useless. This results in great environmental impacts.
  - There are proposals to do meaningful computations instead. The challenge is can you devise one very specific problem that would be useful to solve, and could come up with a proof of work around it. It is not clear how to generalize that.

- Someone with over 50% of the computing power is able to attack all coins having similar algorithm.
- Mechanism that bitcoin has to defend against selective inclusion of transactions is that there's a transaction fee. So, you get an additional small fee per transaction that's included in the block.

Proof-of-stake:

- Proof-of-stake is an alternative proposed, where the chance to win a mining lottery is proportional to your current balance, the current number of coins you have in this currency. The idea there is the people who own a lot of coins have an incentive for the system to work well.
- One problem with the naive proof-of-stake protocol is the incentive for honest miners to mine on both chains if there is a fork. This could result in an attacker who can intentionally double spend: an attacker can spend on both chains, but only mine on one of them, so they can double spend on the other chain.

Proof-of-burn:

- Someone needs to irrevocably burn something valuable to participate in a mining lottery. And my chance of winning a lottery is proportional to the money I burnt for that lottery.
- People don't waste computing power like in proof-of-work
- It stops Sybil attacks because participants needs to burn something expensive

Some questions:

- **Has anyone got over 50% of the computing power in Bitcoin?**
    - Yes, a Bitcoin mining pool once had more than 51% of the computing power. That's bad because if the pool was malicious, they could send instructions to their miners that would do 51% attacks. So what happened was when, when that was discovered, they voluntarily divested enough of their clients so that they went down to 30 percent or so. There was no attack that was actually done.
- **Any persistent fork in Bitcoin?**
    - Yes, there had been persistent forks of Bitcoin, but not because of any attack. There is a foundation who is in charge of making decisions about revising the protocol, and people within the foundation couldn't agree. Half of the organization founded a new organization which runs a different Bitcoin protocol. There is a persistent fork between people who follow the new organization and the people who follow the original one.
- **(After student lead discussion) What are the barriers for Bitcoin to widespread use in real life?**
    - Frequency and latency issues
        - it takes around 10 minutes for a new block to appear on the chain, and each block has a fixed capacity to include limited transactions.
    - Privacy issues
        - Even though individual transactions are anonymous, correlation between transactions can be detected.

        - Mix nets could be a solution: it is a trusted entity where if I want anonymity, I can submit my coin to the trusted entity, and the entity will send me a different coin. The trusted entity acts like an exchange who removes the trace of a coin.
    - Stable pricing
        - as everyone knows, the price of bitcoin fluctuates greatly
    - Fraudulent charges are irrevocable
        - in a bank-backed system, fraudulent charges can be handled by the band, but it is very hard if possible to do so with cryptocurrency
    - Managing keys securely and conveniently is very hard for regular users
    - Political reasons
        - the government's power and ability are compromised
    - Environmental issues as mentioned above

Lead Discussion on Ethereum and Smart Contracts - Neil

Smart contracts

- You can think of smart contracts as code, the collection of functions which you can send from users to each other and to other contracts and transactions. You can initialize these new contracts and functions. And, transfer ethereum, which is the currency between them.

Gas

- Gas is used to pay to execute these transactions. So gases bounds the number of execution steps that the contract will execute. So gas is used to prevent denial of service attack

Two main types of smart contract attacks

- Call to unknown
    - When a function invocation or an ethereum transfer unexpectedly invokes the fallback function of the callee/recipient
- Reentrancy
    - It can occur when you create a function that makes an external call to another untrusted contract before it resolves any effects. If the attacker can control the untrusted contract, they can make a recursive call back to the original function, repeating interactions that would have otherwise not run after the effects were resolved.

DAO attack, exploiting the reentrancy vulnerability:

- **Step 1:** The `Attacker` initiates a transaction by calling the `Withdraw` function of the `Victim`;
- **Step 2:** The `Victim` transfers the money and calls the `fallback` function of the `Attacker`;
- **Step 3:** The `fallback` function recursively calls the `withdraw` function again, i.e., Reentrancy;

- **Step 4:** Within an iteration bound, extra ether will be transferred multiple times to the `Attacker`.

Other vulnerabilities

- Execution disorder: gas runs out, call stack reaches limit, or throw
- Stack size limit
- Miners can choose block timestamps for possible advantage