Bitcoin Mining and Attacks (Sybil, Eclipse, 51% …)

CMSC 23280/ECON 23040, Winter 2019 Lecture 5

David Cash, Harald Uhlig, Ben Zhao
University of Chicago

Transition… of Sorts

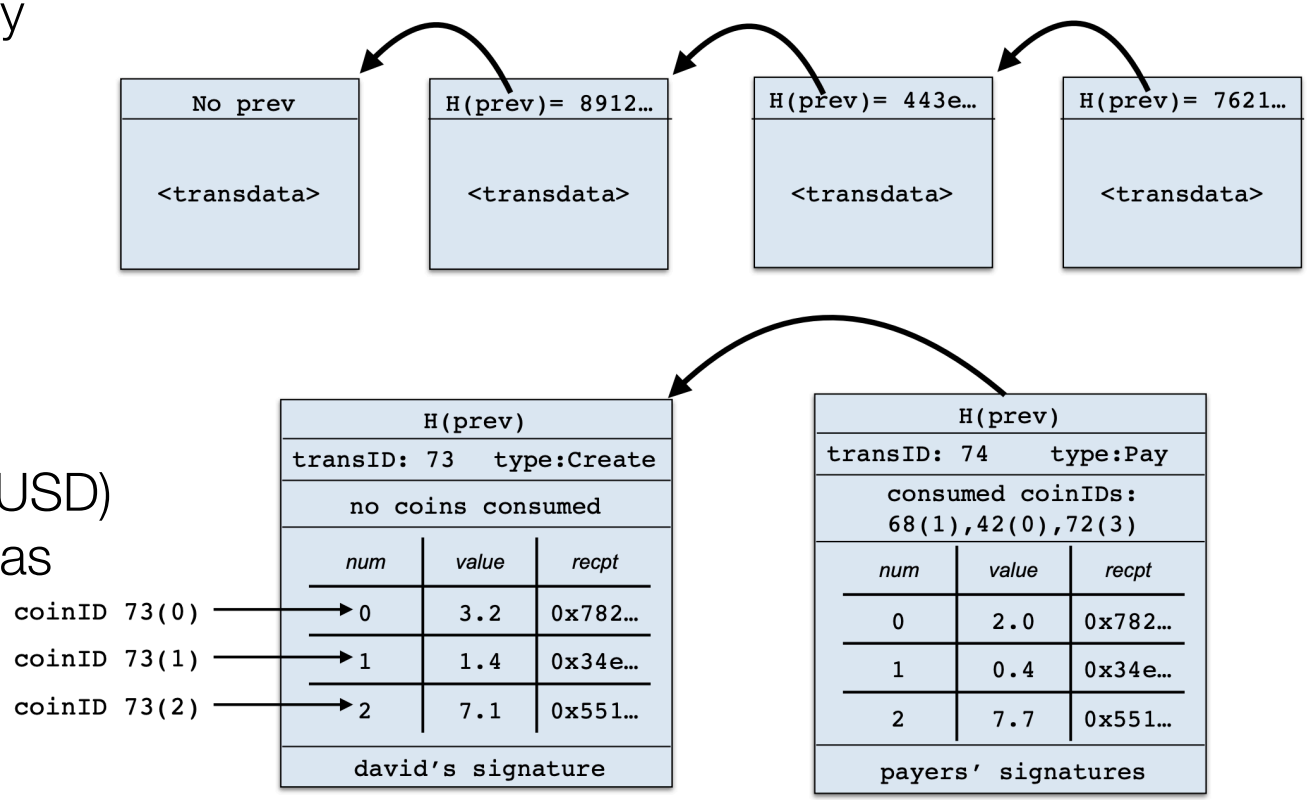
- I'll be leading remaining CS lectures
  - Jan 23 (today): Bitcoin mining and network attacks
  - Jan 28: Ethereum and smart contracts
  - Jan 30: Scaling for Throughput
  - Feb 4: Potpourri or Open Q&A
  - Feb 6: Class midterm
- Other logistic notes
  - Assignment 2 gets a one-day extension: now due Thursday night
  - Assignment 3 released at Assignment 2 due time

Lecture 5 Outline

1. **Evolution of mining and mining pools in Bitcoin**
2. Attacks on and using mining pools
3. Sybil Attacks and Eclipse Attacks

Recall: Mining a Block

- Intertwines recording of transactions with generation of currency
- Miners incentivized by
  - Block reward
    Currently 12.5 BTC
    (~ $45537 today)
  - Transaction fees
    currently quite low (0.3 USD)
    likely to grow over time as block rewards diminish




So You Want to
be a Miner...

What a Miner Does

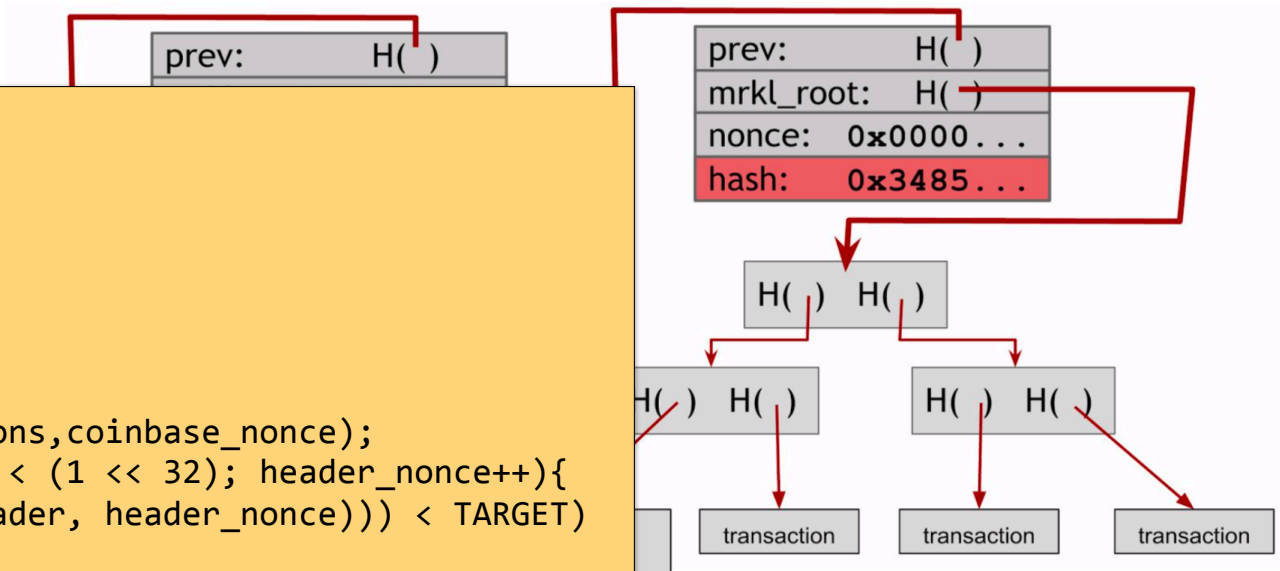
1. Listen for transactions
2. Maintain block chain & listen for new blocks
3. Assemble a candidate block
4. Find a nonce that makes your block valid **(proof of work)**

← **The tricky part is here**

5. “Hope” your block is accepted
6. Profit, and repeat

Recall: Merkle Trees

- Hierarchy of hashes going up the tree
- Two nonces

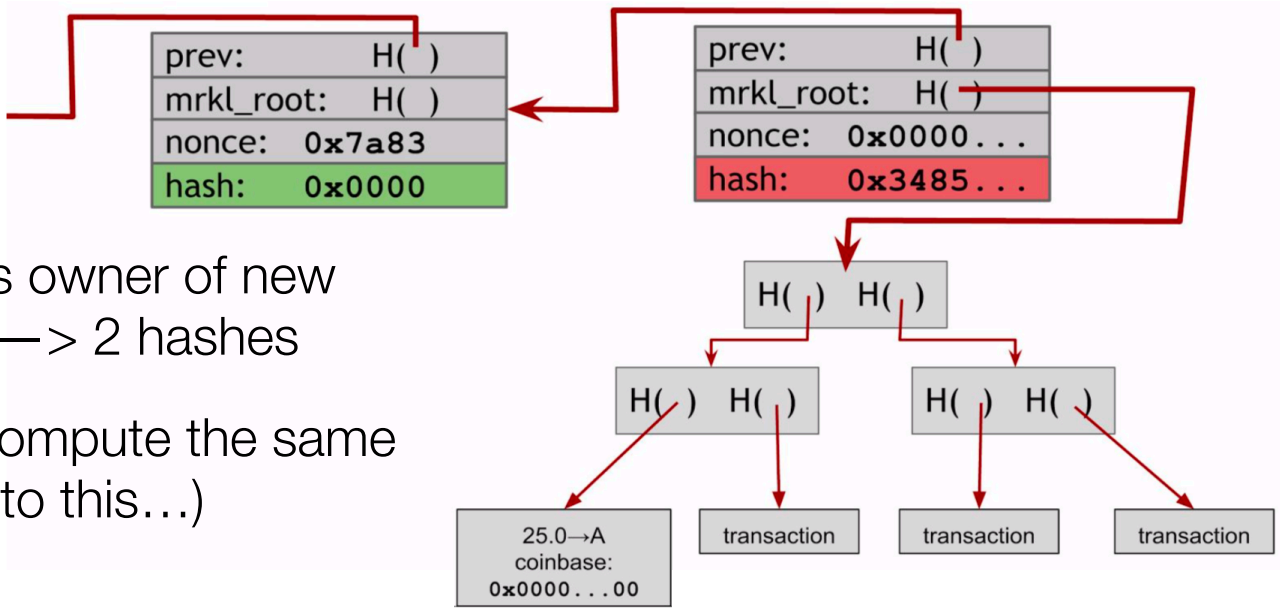



```
Mining Pseudo Algorithm

block := construct next block head
difficulty := next difficulty
// look for nonce
TARGET=(65535<<208)/DIFFICULTY;
coinbase_nonce=0;
while(1){
   header=makeBlockHeader(transactions,coinbase_nonce);
   for(header_nonce=0; header_nonce < (1 << 32); header_nonce++){
      if(SHA256(SHA256(makeBlock(header, header_nonce))) < TARGET)
        break; //we're rich!
   }
   coinbase_nonce++;
}
```

Uniqueness in PoW

- Worth noting each computational puzzle is different. WHY?
  - Different sample of transactions?
  - Specify own address as owner of new block reward: 2 nodes —> 2 hashes
- But what if you ***want*** to compute the same puzzle? (we'll come back to this…)

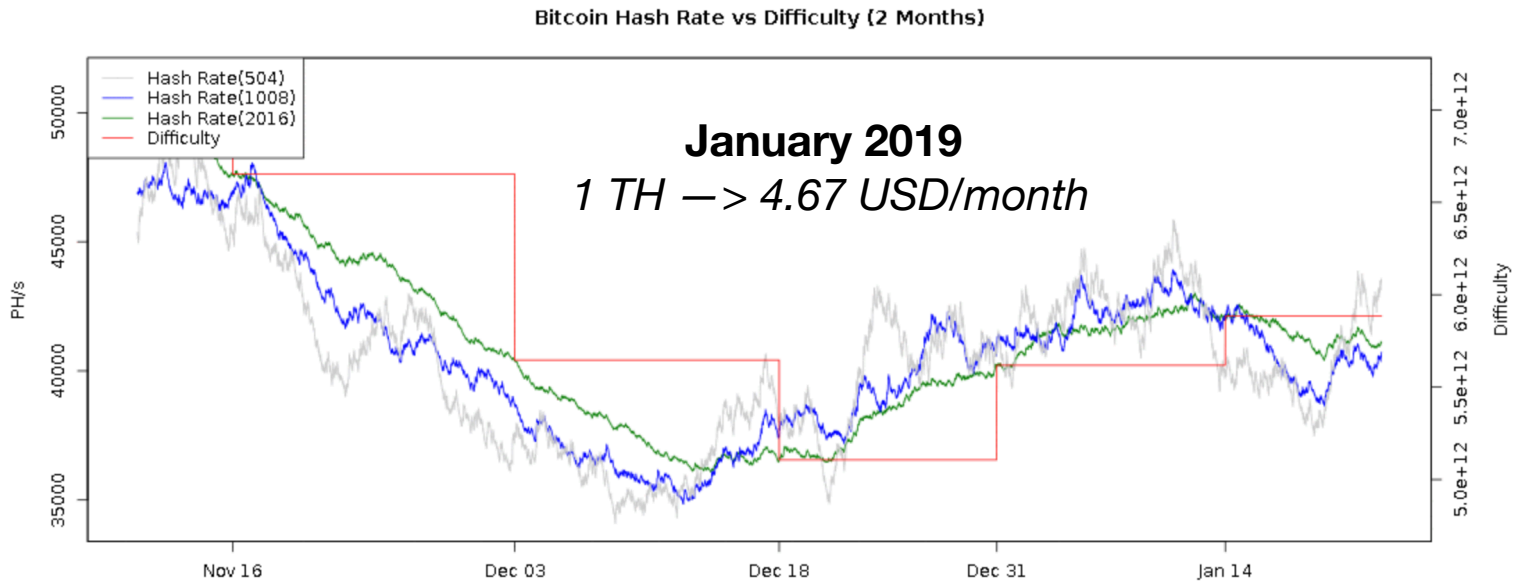


PoW Adaptive Difficulty

- Mining difficulty changes every 2,016 blocks

  $Diff_{T+1} = (Diff_T * 2016 * 10 \text{ mins}) / (\text{time to mine last 2016 blocks})$

  - Scales naturally with aggregate power of miners in network

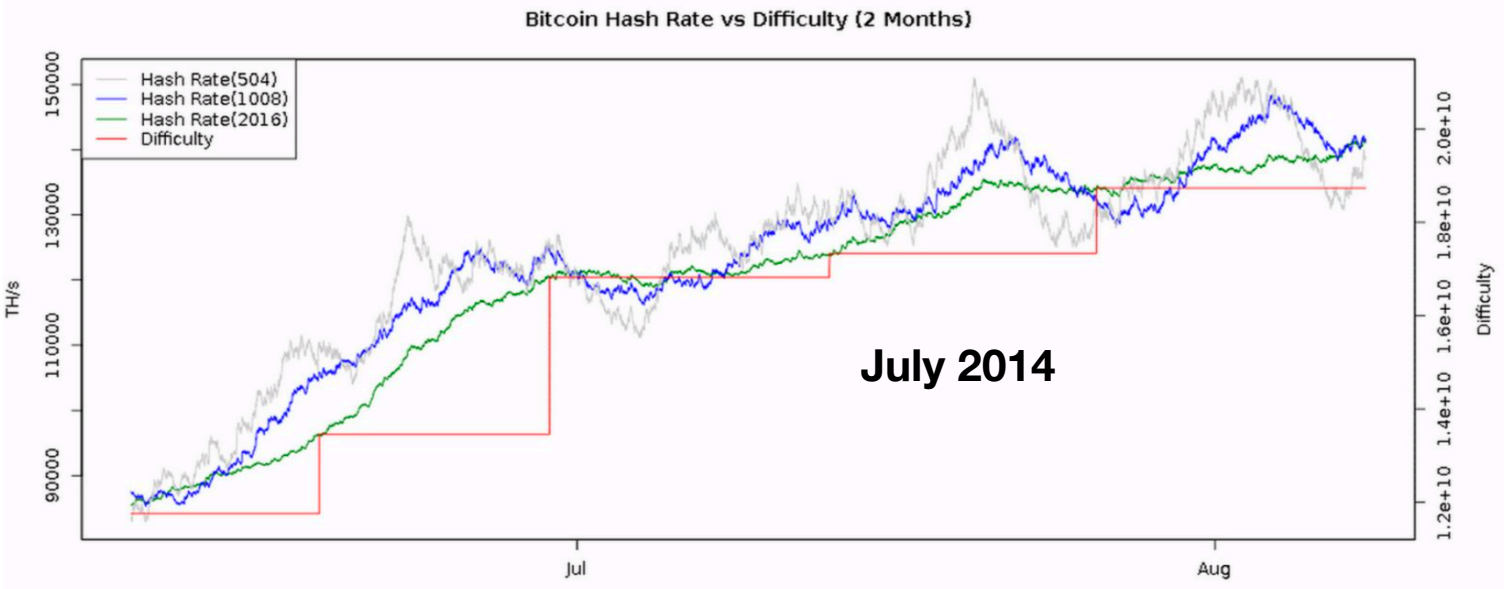


PoW Adaptive Difficulty

- Mining difficulty changes every 2,016 blocks

  $Diff_{T+1} = (Diff_T * 2016 * 10 \text{ mins}) / (\text{time to mine last 2016 blocks})$

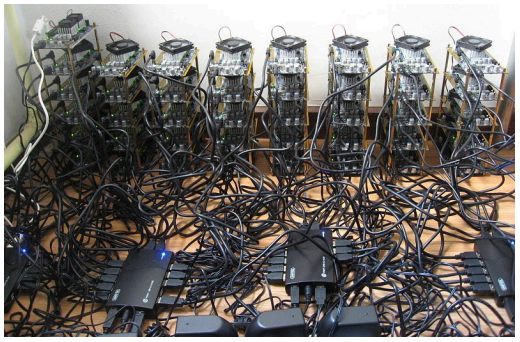
  - Scales naturally with aggregate power of miners in network



How Fast Can We Mine?

- In 2015 (when difficulty was 1% of today)
  - Single Amazon large instance (M3.large) takes roughly 12M years / block
  - Today, expected return of mining with a server is ***negative***!

GPU mining code released (2010)

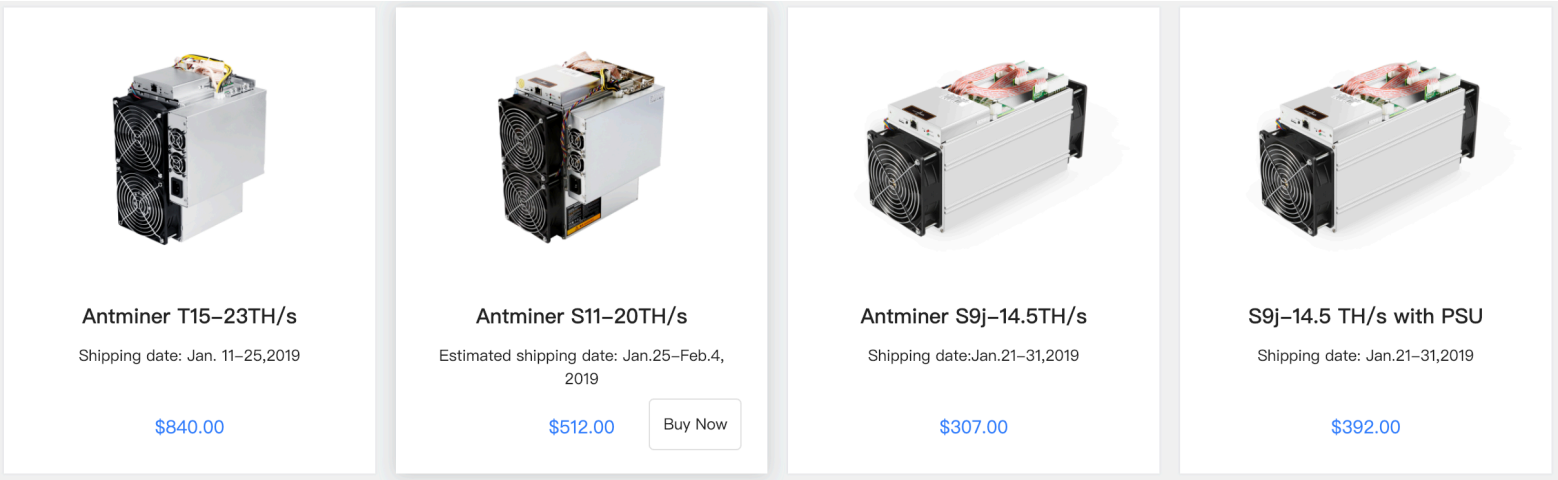


FPGAs Peak Popularity (2011)
1/3 to 1/5 power consumed

ASICS rise in 2012

How Fast Can We Mine?

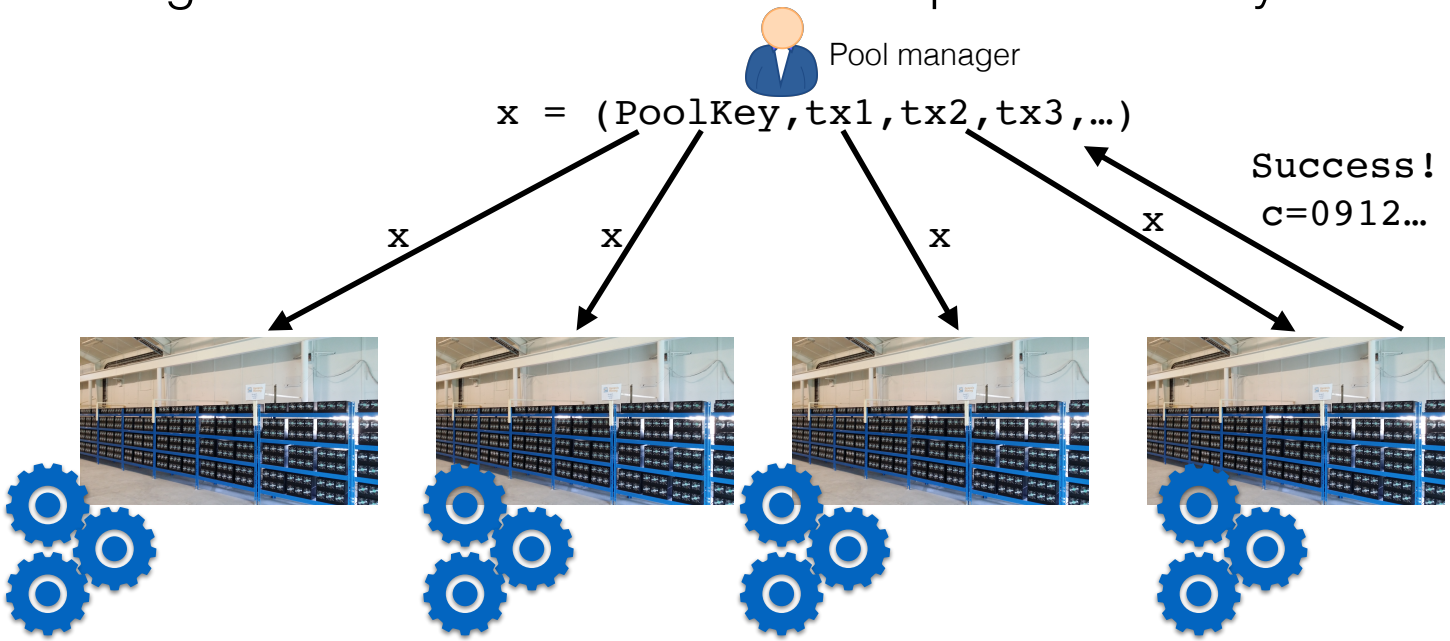
- In 2015 (when difficulty was 1% of today)
  - Single Amazon large instance (M3.large) takes roughly 12M years / block
  - Today, expected return of mining with a server is ***negative***!



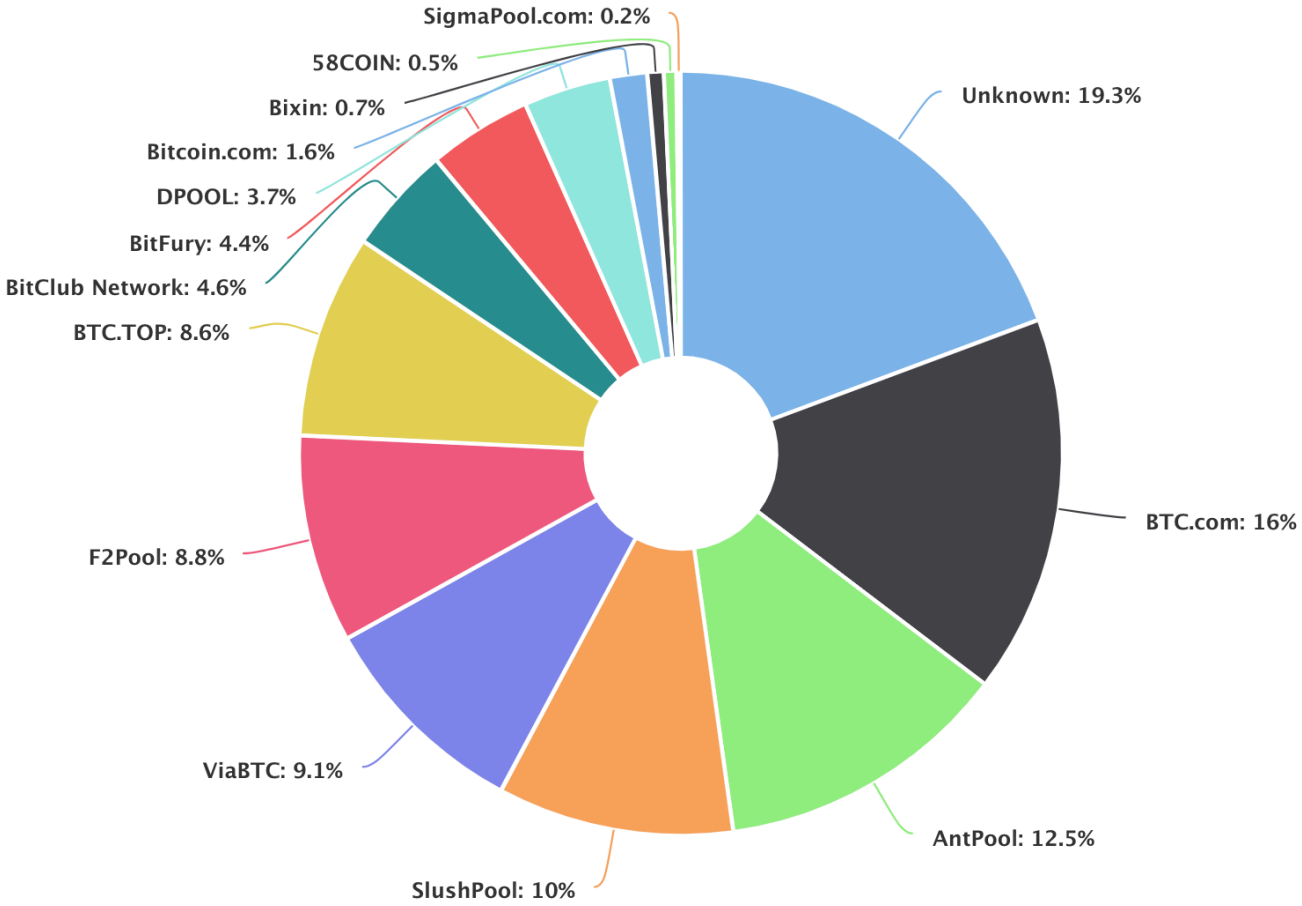
Biggest problem for individual miners: **variance!**

Reduce Variance: Mining Pools

- Pool manager forms POW input x and sends to miners
  - Coordinates value of c to avoid duplicating work
- Pool manager collects fees and distributes profits to everyone



Hashrate: Mining Pools (2019)



% share roughly equals probability of finding next block.

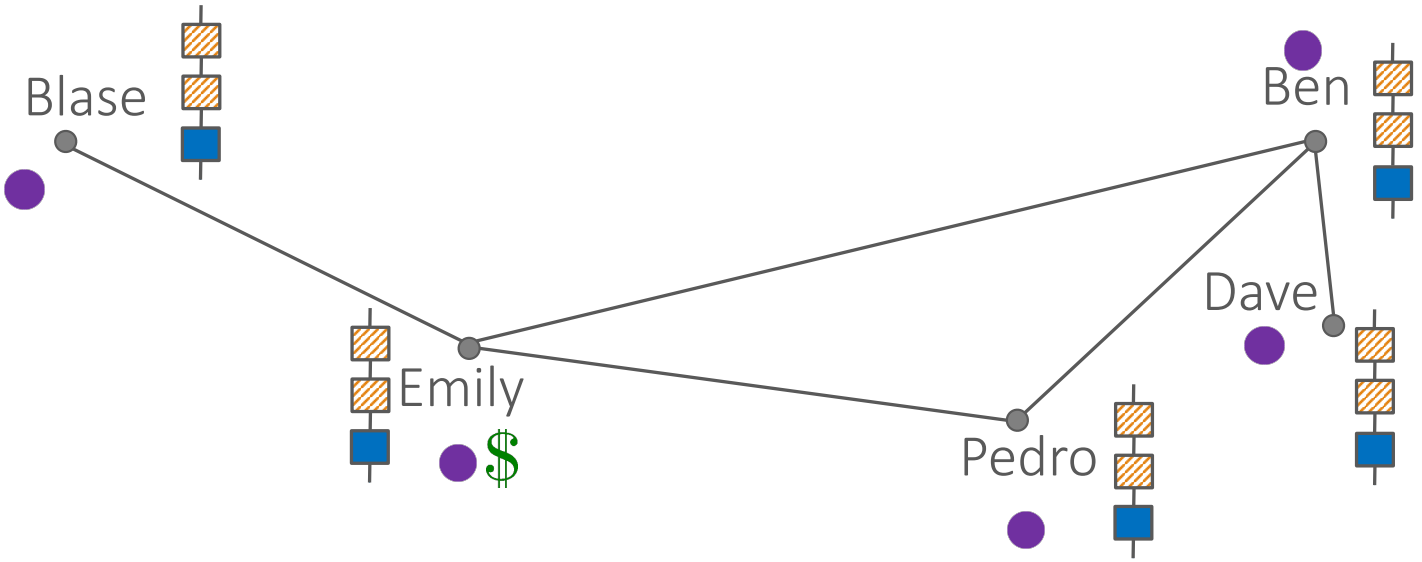
Lecture 5 Outline

1. Evolution of mining and mining pools in Bitcoin
2. **Attacks on and using mining pools**
3. Sybil Attacks and Eclipse Attacks

Bitcoin Mining

Blockchain

Fairness: If Emily has 1/4th computation power, she gets 1/4 of total reward



51% Attacks

Suppose some agent has the majority of compute power on the network:

$$\Pr[\text{agent n solves POW first}] \approx \frac{\text{agent n's compute power}}{\text{network's total compute power}} \geq 0.51$$

- Successful attack can reverse committed transactions; double spend
  - Perform transaction; mine blocks ahead of public chain; announce longer chain without previous transaction
  - Effectively control what transactions are valid

Mining pool giant GHash.io reaches 50% of bitcoin hashing power

Posted by Steve Shanafelt at June 13, 2014 in Bitcoin Mining, Bitcoin Tech, Finance, News Co

pool giant GHash.io reaches 50% of bitcoin hashing power

Yesterday, mining pool giant GHash.io controlled fully half of the hashing power of the entire bitcoin network. For a distributed network that relies on trustless consensus, having the majority of the "voting" power controlled by a single mining pool is more than a little problematic. Most notably, it opens up the bitcoin netw
com

At t
it's p
dam
larg

GHASH.IO

Bitcoin mining pool GHash.IO is preve accumulation of 51% of all hashing power

GHash.IO, the worlds largest and most powerful mining pool, has entered 2014 with overall hashing power of over 40%, making it the #1 pool currently in the Bitcoin network.

The pool has gained significant hashing power due to the 0% pool fee, merged mining of alt coins, excellent real-time data presentation as well as quality 24/7/365 support service.

We have put a plan in place to see that 51% of all hashing power, will not be maintained by Ghash.IO by executing the following actions:

- We will temporarily stop accepting new independent mining facilities to the Ghash.IO pool.
- We will implement a feature, allowing CEX.IO users to mine bitcoins from other pools. So when they purchase GH/s they can put it towards any pool they choose.

We will not be implementing a pool fee, as we believe the pool has to remain free.

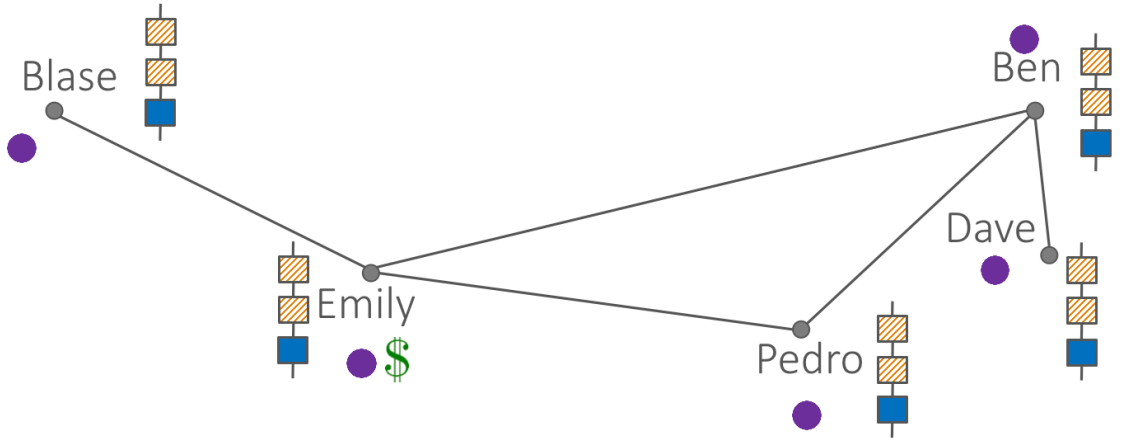
GHash.IO does not have any intentions to execute a 51% attack, as it will do serious damage to the Bitcoin community, of which we are part of. On the contrary, our plans are to expand the bitcoin community as well as utilise the hashing power to develop a greater bitcoin economic structure. If something happened to Bitcoin as a whole it could risk our investments in physical hardware, damage those who love Bitcoin and we see no benefit from having 51% stake in mining.

Recall the Fairness Assumption

Bitcoin Mining

Blockchain

Fairness: If Emily has 1/4th computation power, she gets 1/4 of total reward



Selfish Mining (Eyal, Sirer 2014)

- Selfish mining: approach to gain profit > your share of hash power
  - Leverage advantage in hash power, hash ahead in private chain (hidden)
  - Only release hidden blocks as necessary to “kill” competing chains
  - Make other miners *waste* computation time

Majority Is Not Enough: Bitcoin Mining Is Vulnerable. In Financial Cryptography and Data Security (FC), March 2014

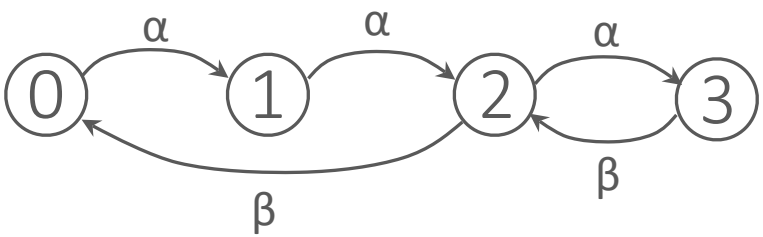
Selfish Mining

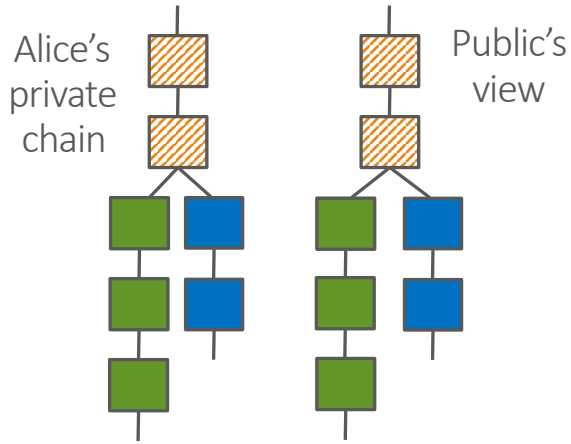
(in more detail)

γ: Alice's ability to win race conditions

Alice (α)	Public (β)

(α, γ): network model parameters



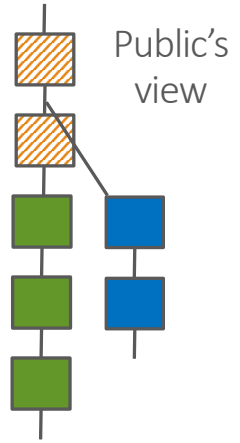


Selfish Mining

(in more detail)

Alice (α)	Public (β)

- Key to profit
  - Create non-computing, zero-power miners as sensors
  - Disseminate attacker's block faster —> win
  - Anyone can launch attack in unmodified Bitcoin



Eyal/Sirer proposed fix
When observing competing branches of same length:
propagate both, randomly choose 1 to mine.
Attack requires minimum 25% mining power.

Other, more complicated attacks exist, but beyond scope of this class.

Detection and Countermeasures?

- No clear-cut detection method (AFAIK), but can infer based on …
  - Number of orphaned blocks (from honest miners or attackers)
    Challenge: orphan blocks silently pruned/discarded;
    also need comprehensive viewpoints for accurate count
  - Timing of successive blocks & deviation from expectation of randomness
    Challenge: statistical measure and takes time
- Countermeasures by attackers
  - Cloak identity / addresses, masquerade as multiple pools
  - Can just release 1 block to compete in network (less $, less detectable)

Lecture 5 Outline

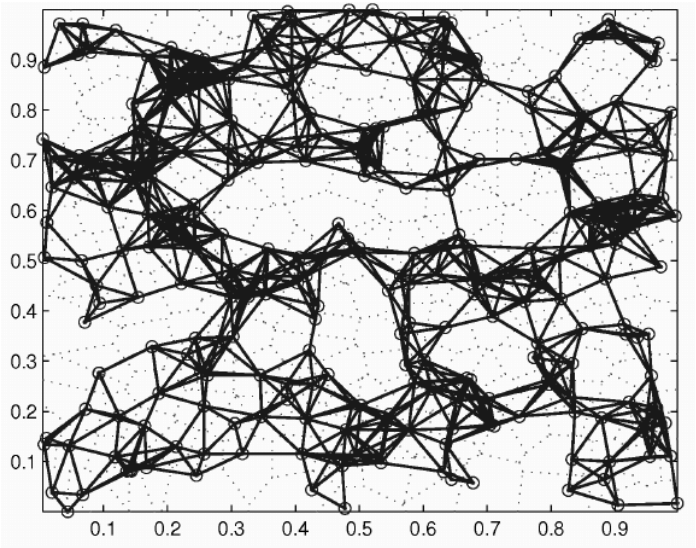
1. Evolution of mining and mining pools in Bitcoin
2. Attacks on and using mining pools
3. **Sybil Attacks and Eclipse Attacks**

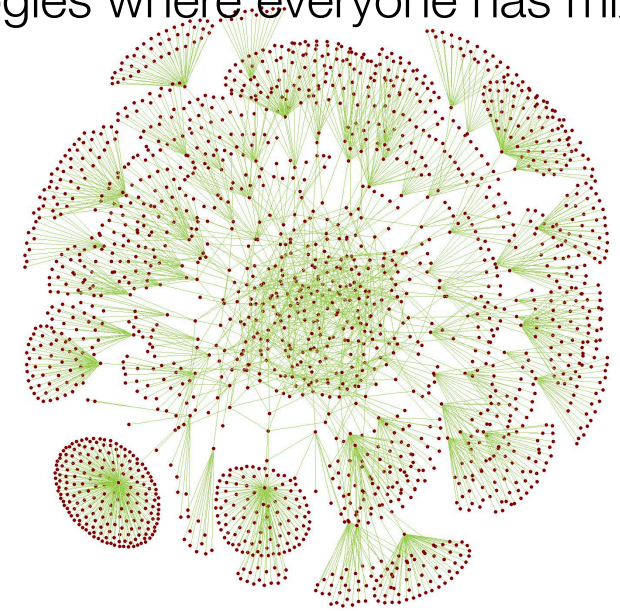
What's Old is New Again

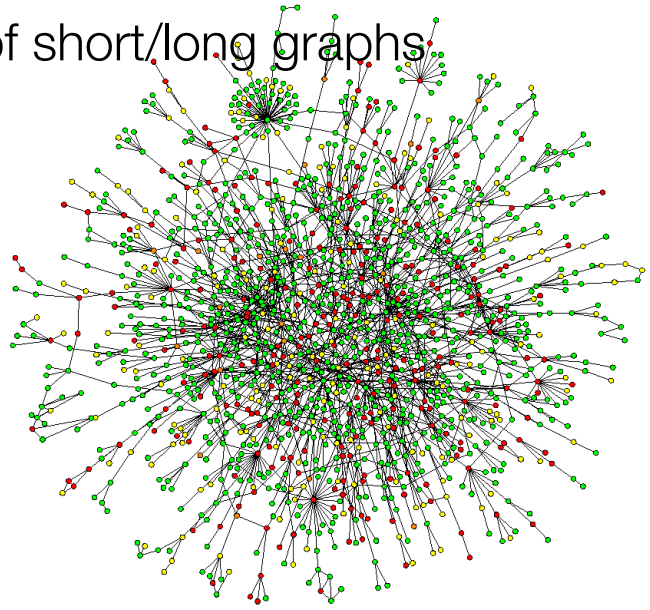
- Bitcoin and most cryptocurrencies reside on peer-to-peer networks
  - P2P networks subject of much research between 1990's to 2000's
  - Network topology analysis, data dissemination latency, robustness to attacks
- Many concerns & attacks applicable to cryptocurrency networks
  - We'll focus on two big ones today: Sybil & Eclipse

First, Network Topolgy

- Flat, planar graphs horrible for disseminating data
  (slow, prone to congestion at barbell-like choke points)
- Single hierarchy better for latency, but not resilience/congestion
- Best: small world topologies where everyone has mix of short/long graphs

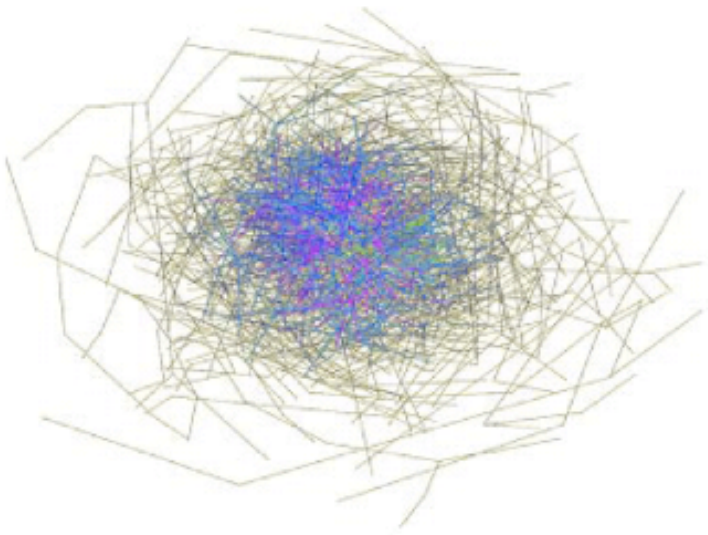




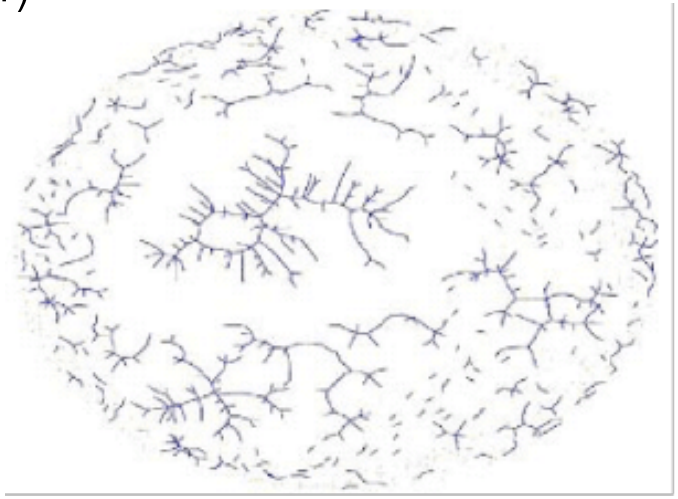


Resilience to Failures and Attacks

- Studies (Barabasi) show dichotomy of resilience for “scale-free networks”
  - Resilient to random failures, but not attacks
- Here’s what it looks like for Gnutella (2001)

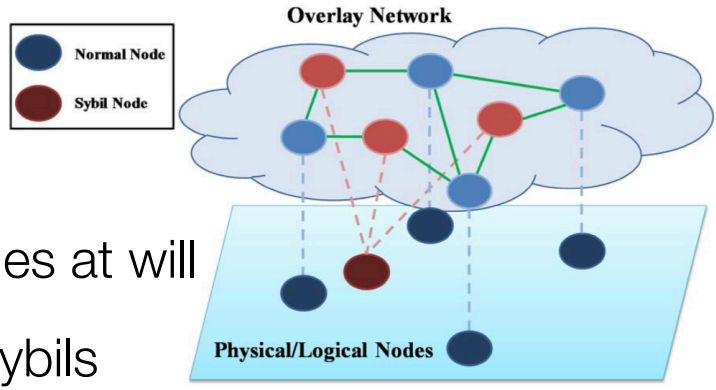


1771 Peers in Feb, 2001



After removing 30% of peers

Sybil Attack



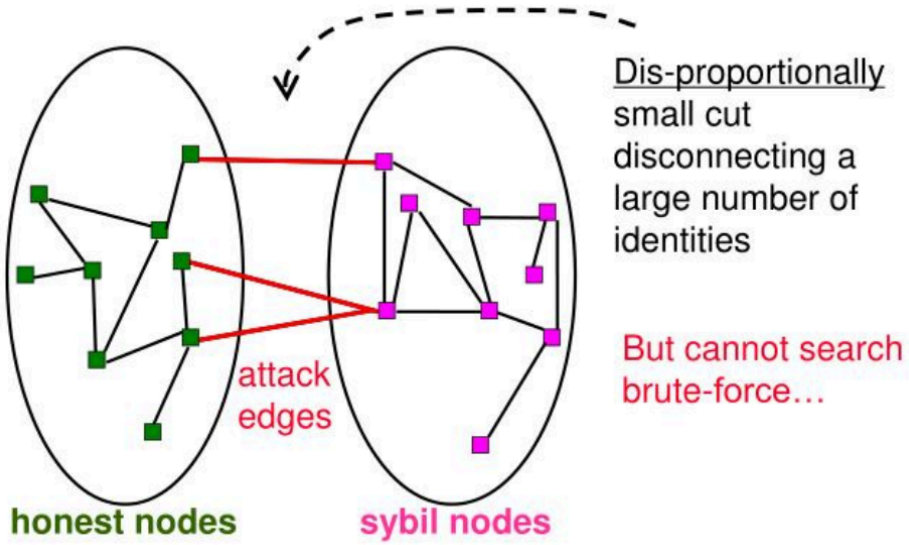
- First identified by John Douceur, MSR, IPTPS 2002
  - Online identities are cheap
  - With sufficient resources, attack can create identities at will
  - No naive quantitative algorithm is robust against Sybils
    - Consensus, majority voting, resource scheduling, lottery scheduling
- Applicable to many contexts
  - Peer-to-peer networks; Mobile ad-hoc networks; Anonymous routing networks (Tor); Online social networks

Targets of Sybil Attacks

- Reputation and review systems
  - eBay, Yelp: fake accounts to give larger volume of + or - reviews
- Internet polls/discussion
  - Reddit, Twitter: multiple accounts generate more votes, astroturfing
- Google page-rank
  - Many sites that cross-link to target, increase ranking in search results
- Mobile apps and communities
  - Waze: software to control large # of users, generate traffic; track users
- Security mechanisms (mostly) all vulnerable to resource attacks
  - Bandwidth, disk space, computational resources, accounts

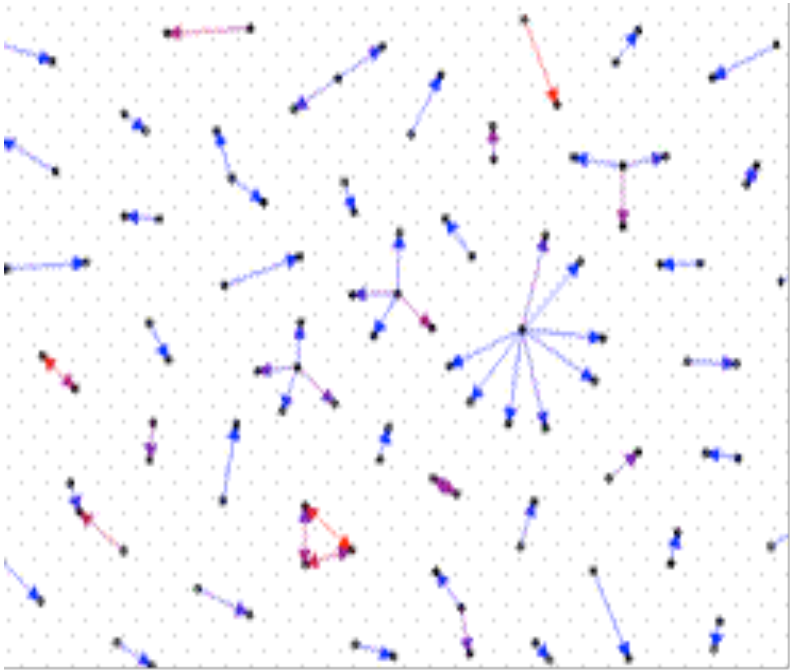
Proposed Defenses

- Trusted authority
  - Requires all nodes to authenticate (out of band mechanism? public keys?)
  - Dourceur, 2002: without centralized authority, Sybil attacks always possible
  - Tradeoffs between trust and privacy: e.g. CyWorld in South Korea
- Resource testing
  - Test computational power, disk, IP addresses, bandwidth, storage
    e.g. CAPTCHA, cryptographic puzzles
  - Adversary may have more resources
- Leverage out-of-band fixed resources:
  e.g. social relationships



Sybil Attacks in the Wild

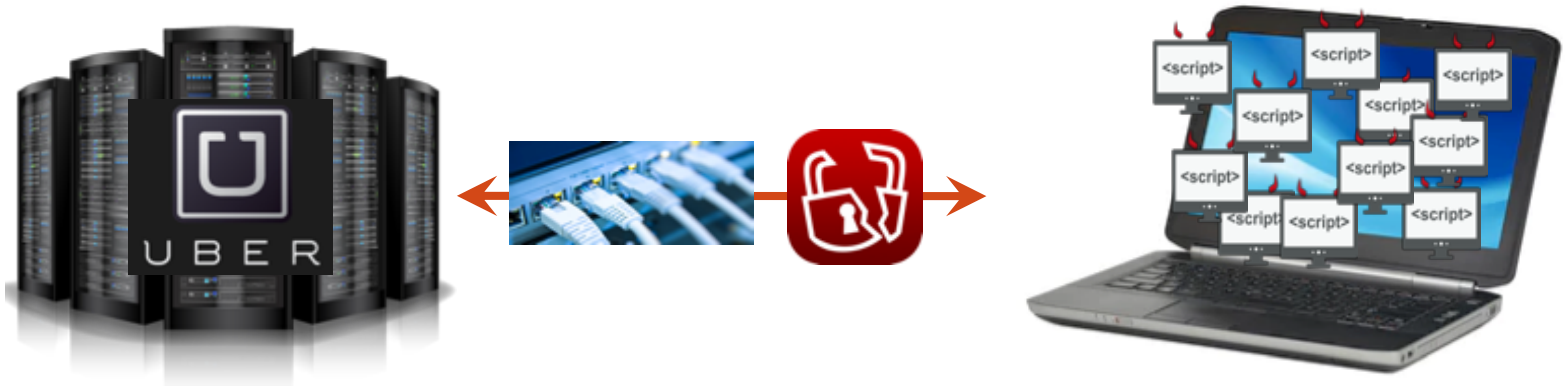
- Empirical study on Maze data-sharing network, 2007 (2M users, 13TB/day traffic)
  - Conducted w/ collaborators from MSR & Peking University
- Why collusion?
  - Collusion gets you points in Maze (incentive system)
  - Spawn dummy users/identities for free
- Collusion detectors
  - Duplicate traffic across links
  - Pair-wise mutual upload behavior
  - Peer-to-machine ratio of clients
  - Traffic concentration degree



Duplicate transfer graph→100 links w/ highest duplicate transfer rates

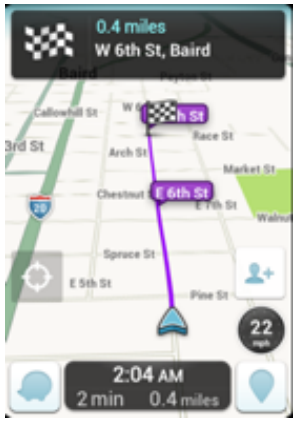
Sybil Attacks on Waze

- Use software to mimic mobile devices
  - Interact with unsuspecting mobile services, e.g. Uber, Tinder, Waze, …
  - Use resources to defeat identity checks
    CAPTCHAs, SMS/2FA, IMEI numbers

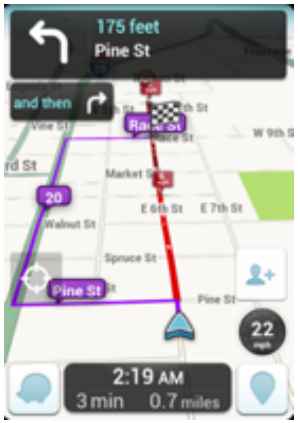


Sybils on Waze cont.

- Reroute users
- “Create” traffic jams, accidents



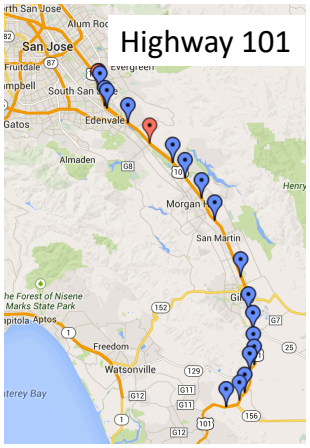
Before



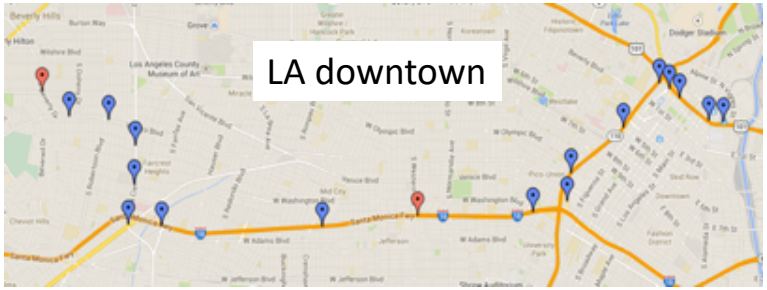
After

- Track precise GPS location of users
- Works at scale of cities

Fast moving target user



Extremely dense user population



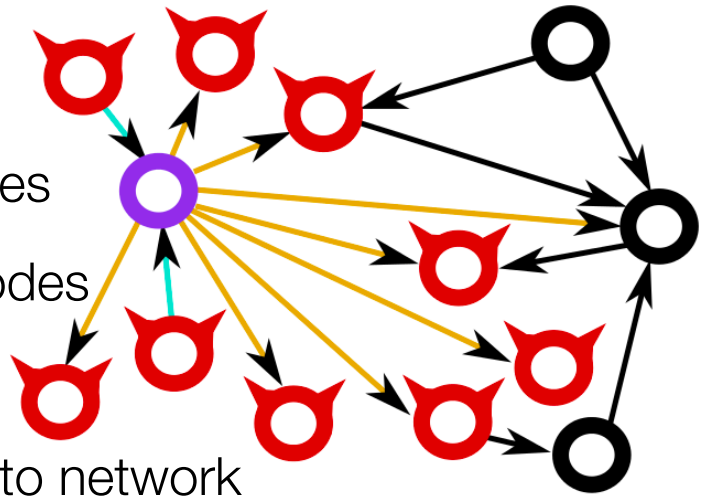
Sybil Attacks on Bitcoin

- “Mostly” addressed through Proof of Work
  - Computational power at scale requires serious $ resources
  - Still not unreasonable for powerful attackers, e.g. nation-states

- Disproportionally more effective on smaller-volume blockchains
  - What happens in Monero if you control 1/2 the nodes in the network?

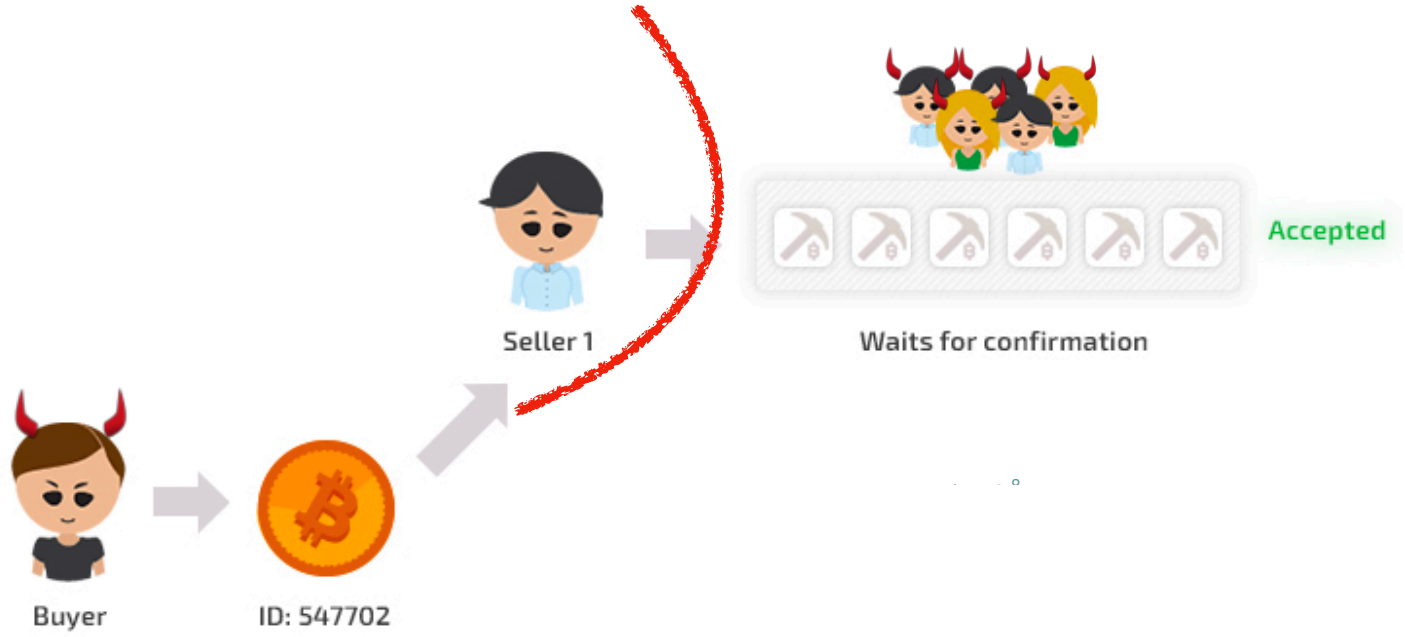
The Eclipse Attack

- Attack focuses on network topology
  - “Block view of network” for victim
  - New nodes discover neighbors via bootstrap nodes
    - Exploit any bias for “fresh” or “resource rich” nodes
  - Attacker infiltrates & dominates neighbor set
- Outcome: control/filter/alter interactions from victim to network
  - Controls/engineer block races
  - 0-confirmation double-spend; N-confirmation double spend



***Eclipse Attacks on Bitcoin’s Peer-to-Peer Network*,** Hellman et al, USENIX Security 2015

Double Spending Attack



Key Takeaways

- Consensus & PoW are not a panacea
  - Cryptocurrency protocols still constrained by network
  - Vulnerable to attacks at lower layers, Sybils, Eclipse, BGP rerouting…
  - security is ***not*** a solved problem

- Community action is *sometimes* responsive
  - Eclipse defenses integrated into Bitcoin and Ethereum protocols
  - Reaction to stronger attacks: emotional responses
    Read comment thread on Gun Sirer's blog on selfish-mining for example

http://hackingdistributed.com/2013/11/04/bitcoin-is-broken/