Cristina Nita-Rotaru

7680: Distributed Systems

Bitcoin security

Acknowledgements

- Based on slides from David Evans, Aviv Zohar, Lefteris Kokoris-Kogias, Ittay Eyal

Bitcoin Protocol

- Each P2P node runs the following algorithm:
  - New transactions are broadcast to all nodes.
  - Each node (miner) collects new transactions into a block.
  - Each node works on solving proof-of-work (PoW) for its block
    - Use computational resources
  - When a node finds a solution, it broadcasts the block to all nodes.
  - Nodes accept the block only if all transactions are valid (digital signature checking) and coins not already spent (check transactions from public ledger).
  - Nodes express their acceptance by working on creating the next block in the chain
    - If multiple valid blocks are available, choose the longest chain and include transactions from discarded blocks in the queue
    - Include the hash of the accepted block as the previous hash.

Nodes eventually reach global consensus on all transactions

Bitcoin security

- Protection again *invalid transactions* (forgery)
  - Cryptographic (digital signature)
- Protection against *modification of blockchain* (remove or modify old transactions)
  - Cryptography (collision-resistant hash functions and digital signatures)
- *Non-repudiation of transactions*
  - Based on blockchain
- Protection against *double spending*
  - Enforced by consensus (correct majority)
  - One of the transactions (either one) will be eventually accepted
- Protection against *Sybil attacks*
  - PoW cryptographic puzzles
  - Assume that adversary does not control majority of CPU resources

Bitcoin: Security issues

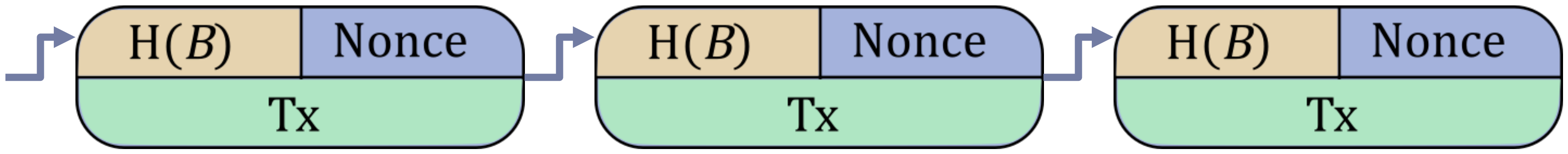
- Consensus algorithm:
  - Is majority enough?
  - Can blocks be removed?
- P2P network:
  - What are the reliability and network connectivity requirements?
  - Are any of the attacks in P2P relevant in this context?

1: Selfish miners

Majority is not Enough: Bitcoin Mining is Vulnerable

Ittay Eyal, and Emin Gün Sirer

Mining

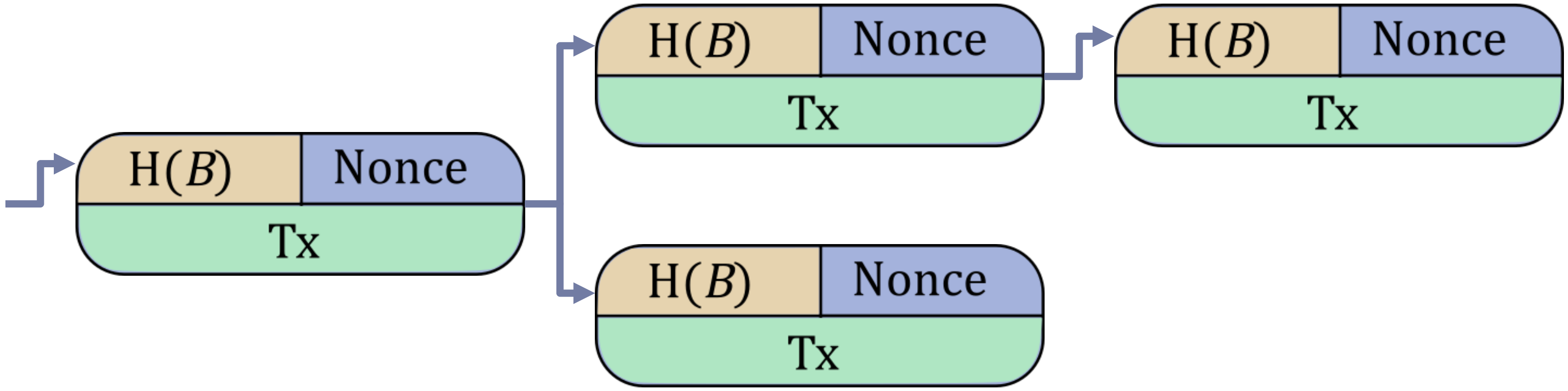


Why do we need miners?

Conflicting Blocks

Fork: multiple miner create blocks with the same preceding block.

- Longest chain wins with random tie-breakings.
- Accidental bifurcation happens once every 60 blocks.



Consensus

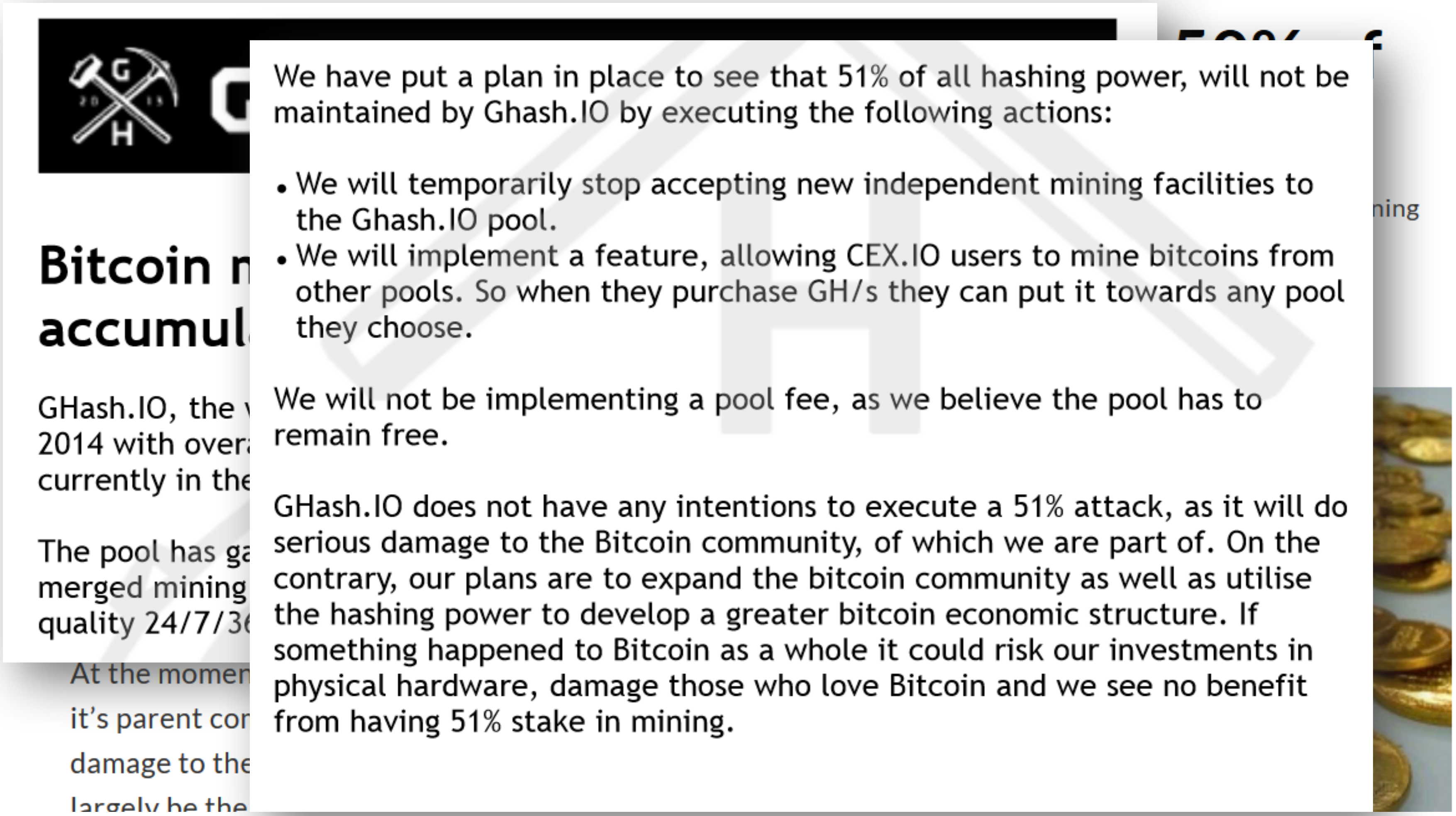
Majority of hashing power has voted for transactions on longest chain.

- It is costly to increase voting power
- Players are not motivated to cheat

The 51% attack!

If any party controls majority of hashing power, they can:

- Undo the past
- Deny mining rewards
- Undermine the currency



Bitcoin m
accumul

GHash.IO, the
2014 with over
currently in the

The pool has ga
merged mining
quality 24/7/36

At the momen
it's parent cor
damage to the
largely be the

ning

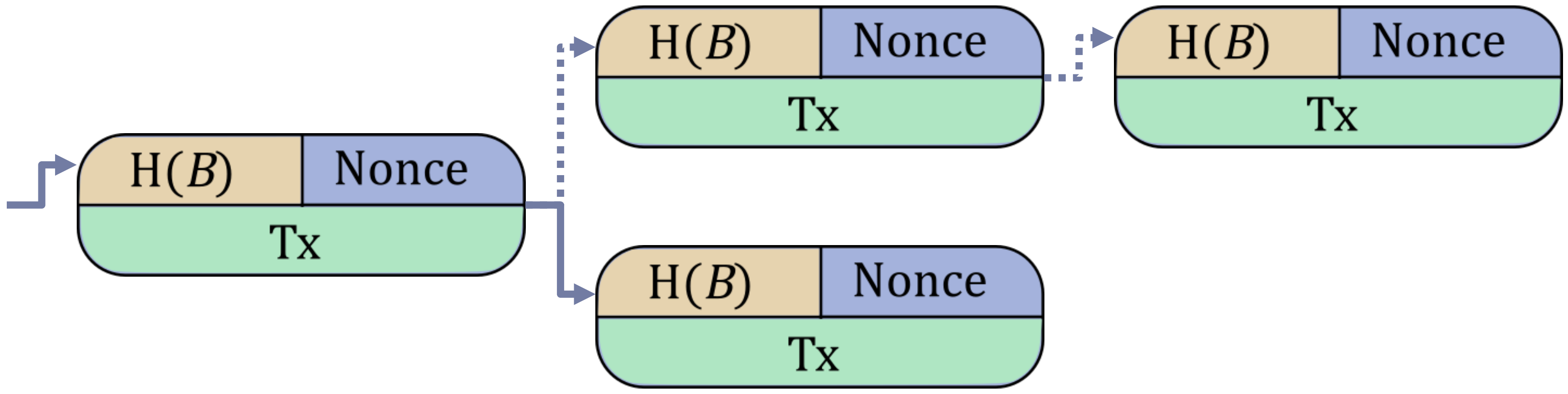
We have put a plan in place to see that 51% of all hashing power, will not be maintained by Ghash.IO by executing the following actions:

- 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.

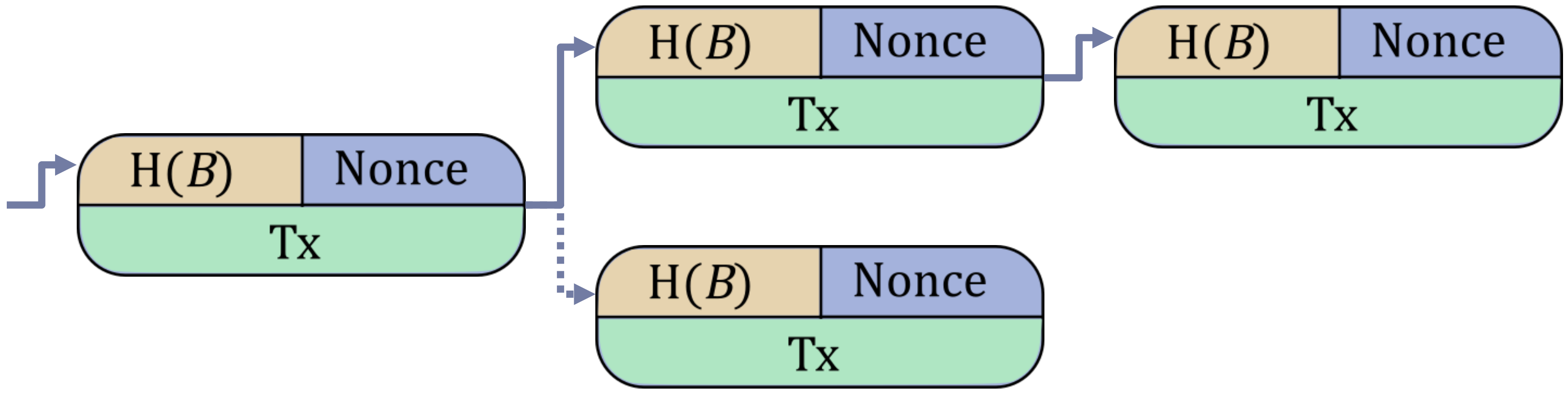
We will not be implementing a pool fee, as we believe the pool has to remain free.

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.

I'll keep these blocks for myself!



I'll keep these blocks for myself!



```
if we gain a lead:
  withhold blocks
  mine on private chain
else if lead shrinks, but is still at least 2:
  reveal blocks to keep abreast with public chain
else if lead drops below 2:
  reveal all blocks
  mine on public chain
```

Worries

“Rational miners will prefer to join the selfish miners, and the colluding group will increase in size until it becomes a majority. At this point, the Bitcoin system ceases to be a decentralized currency.”

Majority is not Enough: Bitcoin Mining is Vulnerable

Ittay Eyal, and Emin Gün Sirer

Detecting selfishness

- Orphaned blocks
  - i.e. valid **blocks** which are not part of the main **chain**:
    - occur naturally when two miners produce **blocks** at similar times
    - can be caused by an attacker (with enough hashing power) attempting to reverse transactions.
- Timing hints

More at: "*How to detect selfish miners*" by Ittay Eyal, and Emin Gün Sirer, http://hackingdistributed.com/2014/01/15/detecting-selfish-mining/

2: Block persistence

Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing
Authors:
Eleftherios Kokoris Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford, *École Polytechnique Fédérale de Lausanne (EPFL)*

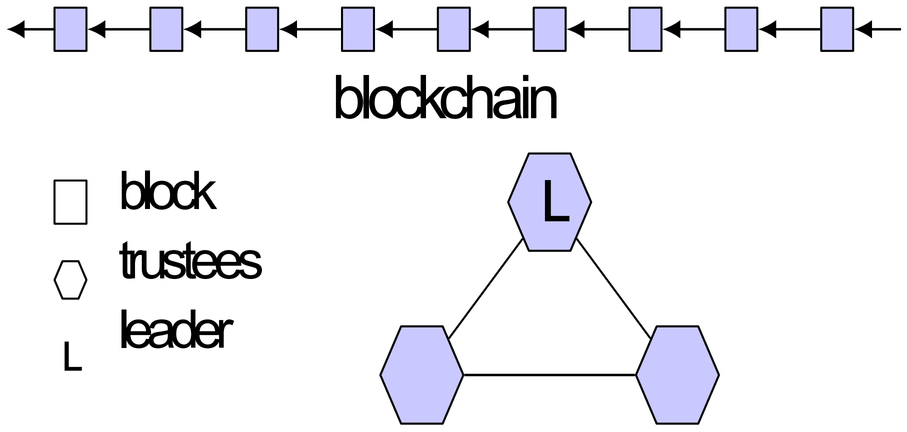
Block persistence

- A transaction is confirmed when it is buried "deep enough" 6 blocks, arbitrary number
- In Bitcoin there is no verifiable commitment of the system that a block will persist
  - Clients rely on probabilities to gain confidence.
  - Probability of successful fork-attack decreases exponentially

Strawman Design: PBFTCoin

- **3f+1** fixed "trustees" running PBFT* to withstand **f** failures
- Non-probabilistic strong consistency
  - Low latency

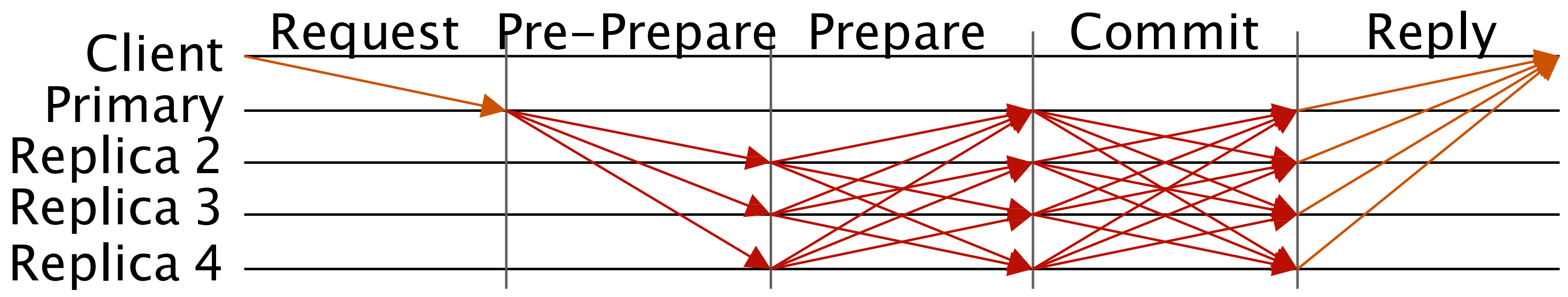
- No forks/inconsistencies
  - No double-spending



*Practical Byzantine Fault Tolerance [Castro/Liskov]

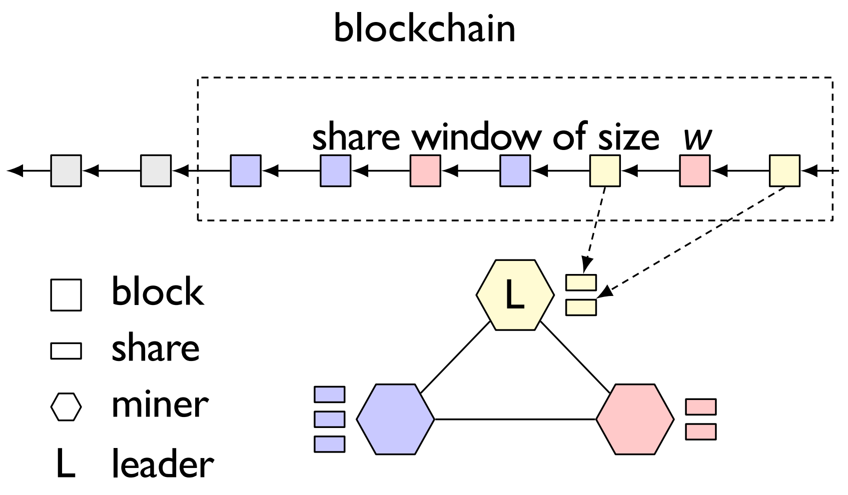
Strawman Design: PBFTCoin

- Problem: Needs a static consensus group
- Problem: Scalability
  - $O(n^2)$ communication complexity
  - $O(n)$ verification complexity
  - Absence of third-party verifiable proofs (due to MACs)



Opening the Consensus Group

- PoW against Sybil attacks
- One share per block
  - % of shares $\propto$ hash-power
- Window mechanism
  - Protect from inactive miners



From MACs to Signing

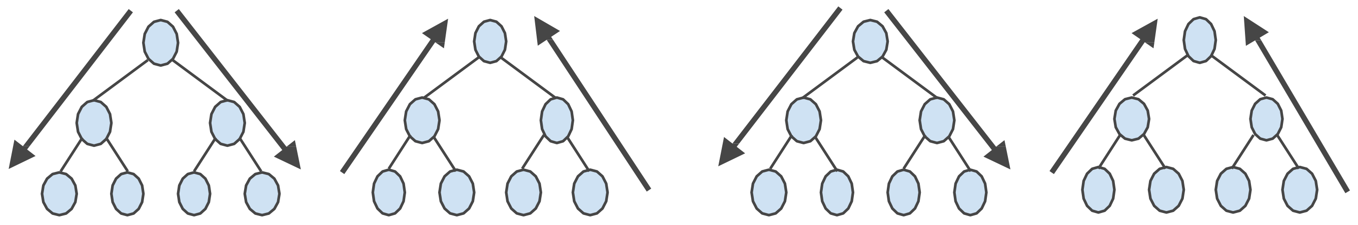
- Substitute MACs with public-key cryptography
  - ECDSA provides more efficiency
  - Third-party verifiable
  - PoW Blockchain as PKI
  - Enables sparser communication patterns (ring or star topologies)

From MACs to Collective Signing

- Can we do better than O(n) communication complexity?
  - Multicast protocols transmit information in O(log n)
  - Use trees!!
- Can we do better than O(n) complexity to verify?
  - Schnorr multisignatures could be verified in O(1)
  - Use aggregation!!
- Schnorr multisignatures + communication trees = Collective Signing [Syta et all, IEEE S&P'16]

CoSi

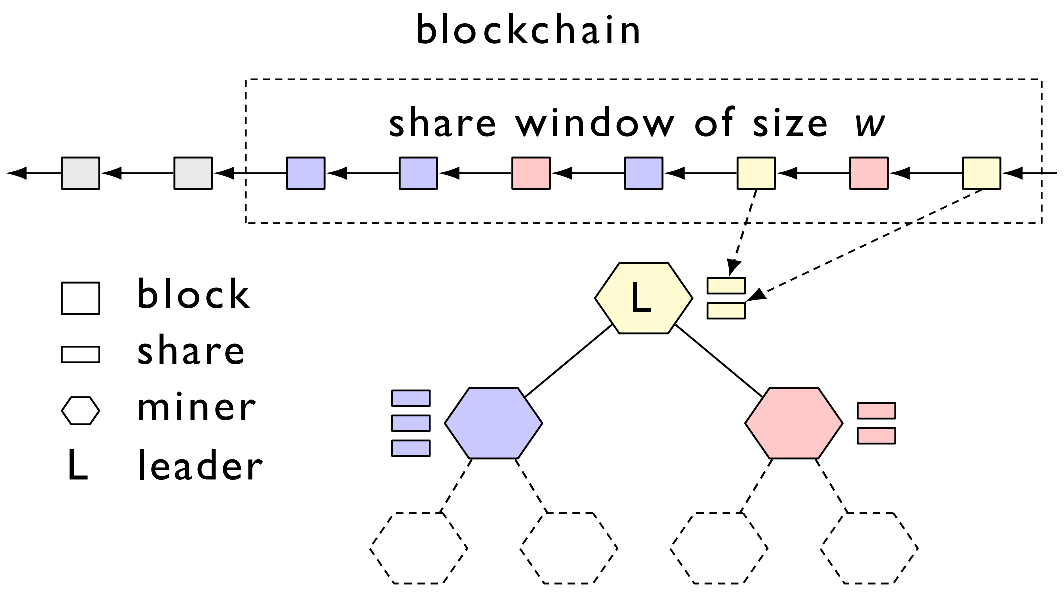
- Efficient collective signature, verifiable as a simple signature
  - 80 bytes instead of 9KB for 144* co-signers (Ed25519)



* Number of ~10-minute blocks in 1-day time window

Discussion

- CoSi is not a BFT protocol
- PBFT can be implemented over two subsequent CoSi rounds
  - Prepare round
  - Commit round



Problem Statement

1. In ~~Bitcoin~~ ByzCoin there is ~~no~~ a verifiable commitment of the system that a block will persist
2. Throughput is limited by forks
   - Increasing block size increases fork probability
   - Liveness exacerbation

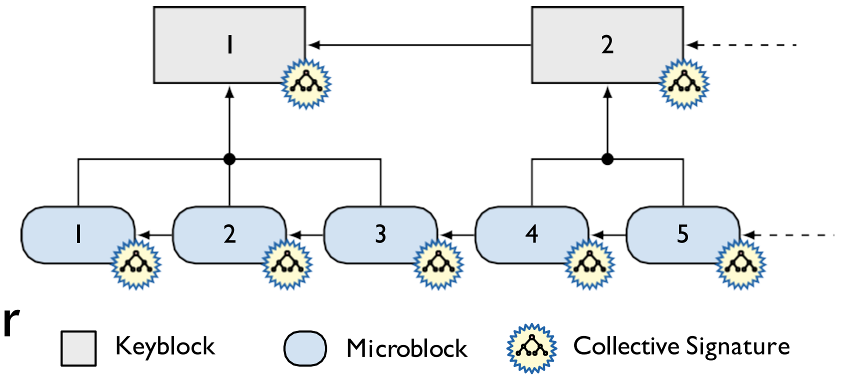
Bitcoin-NG [Eyal et all, NSDI ’16]

- Makes the observation that block mining implement two distinct functionalities
  - Transaction verification
  - Leader election
- But, Bitcoin-NG inherits many of Bitcoin’s problems
  - Double-spending
  - Leader is checked after his epoch ends

Decoupling Transaction Verification from Leader Election

Two type of blocks:

- Key blocks:
  - PoW & share value
  - Leader election
  - Contains public key used for future microblocks
  - Leader wins 40% of the transactions's revenue
- Microblocks:
  - Validating client transactions
  - Issued by the leader



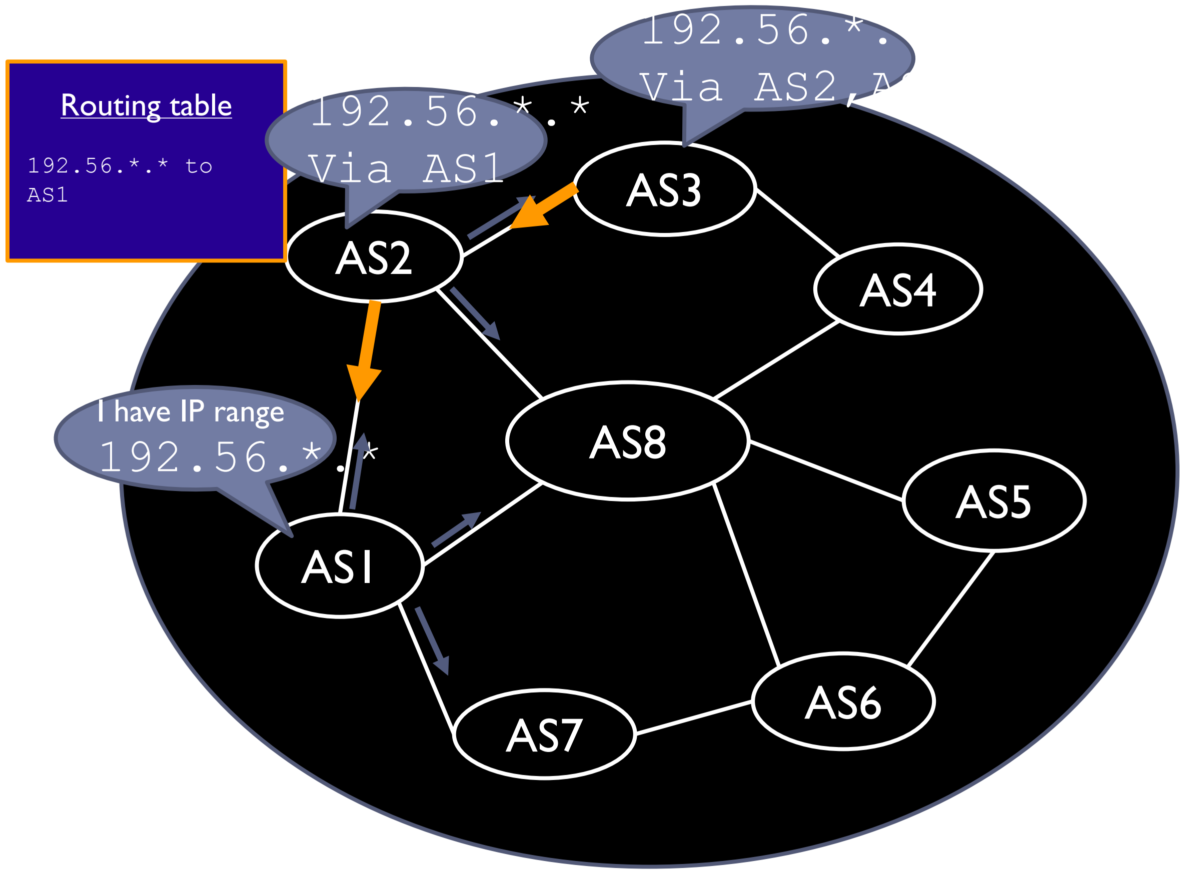
3: Eclipse attacks

Eclipse Attacks on Bitcoin’s Peer-to-Peer Network
Ethan Heilman Alison Kendler Aviv Zohar Sharon Goldberg

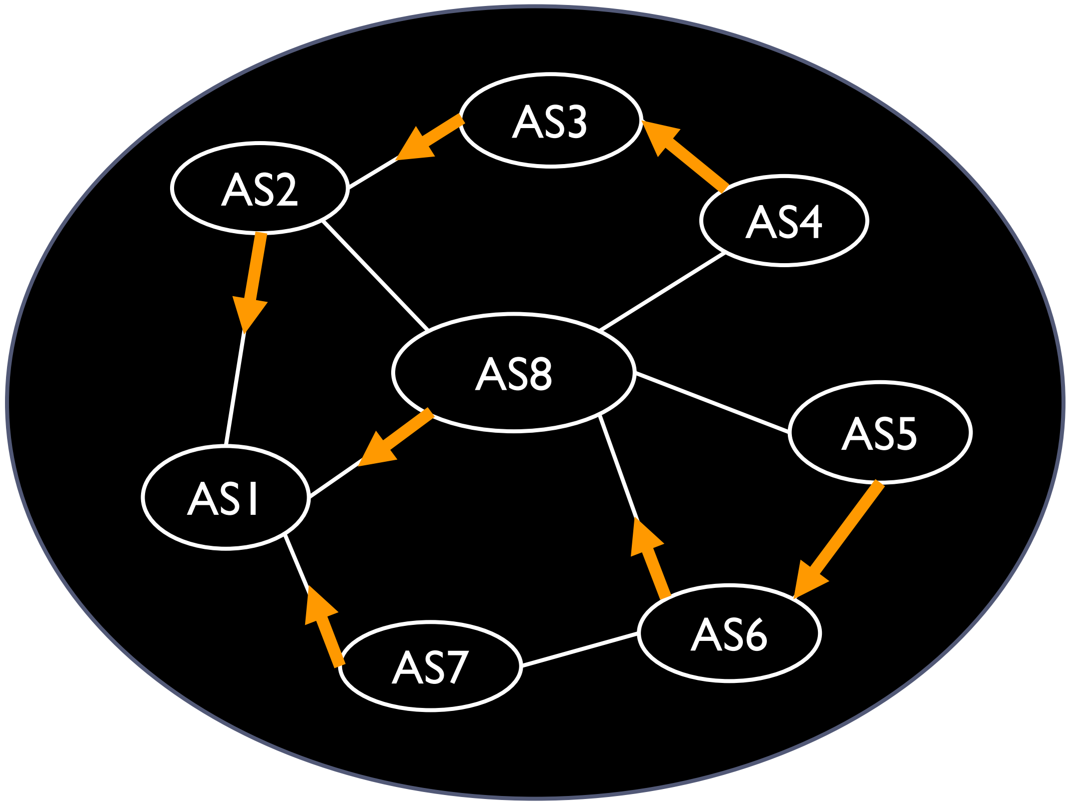
Eclipse attacks

- Eclipse attack: an attacker isolates the victim from the rest of the peers, i.e. controls all of the victim's incoming and outgoing connections
- Attackers
  - On-path attackers
  - Off-path attacker
- Implications for bitcoin:
  - Attacker can then filter the victim's view of the blockchain
  - Force the victim to waste compute power on obsolete views of the blockchain
  - Use the victim's compute power for its own purposes

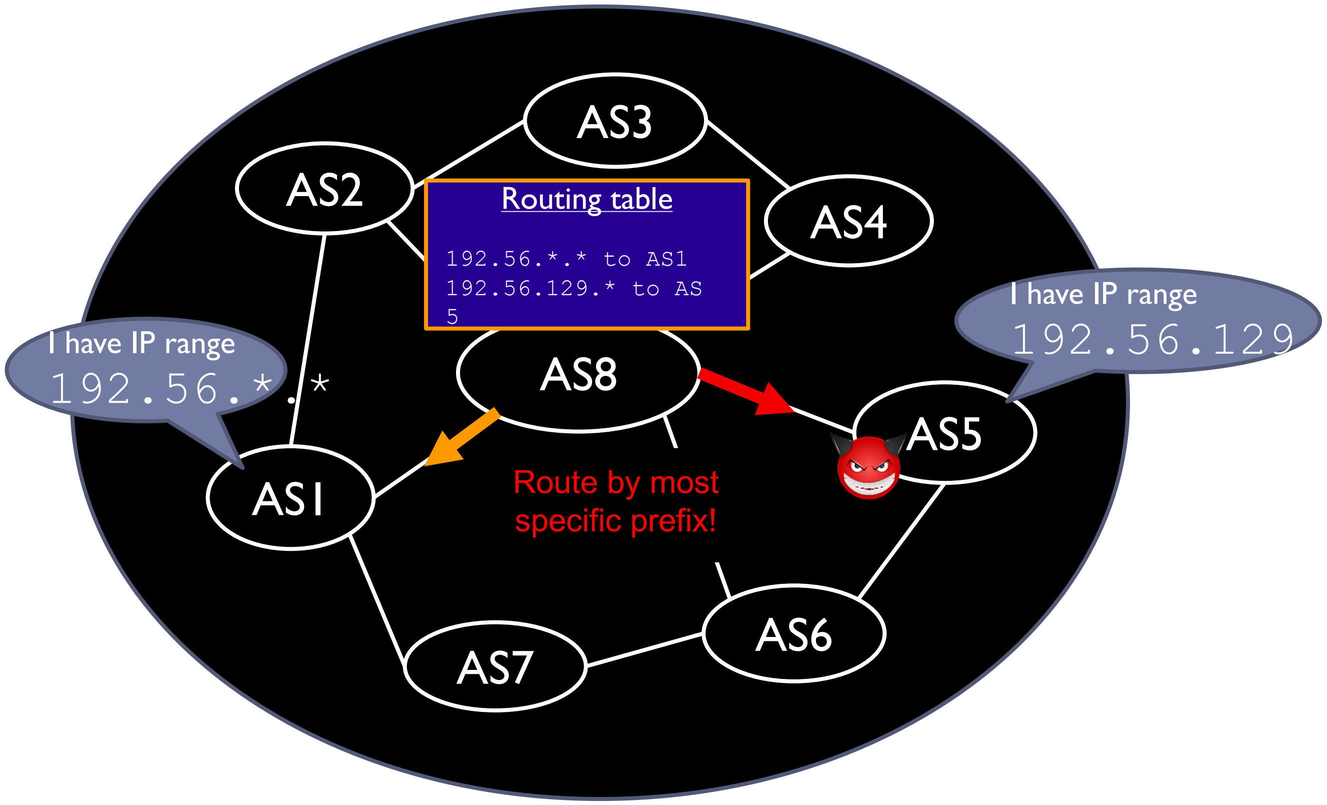
BGP and Routing



BGP and Routing



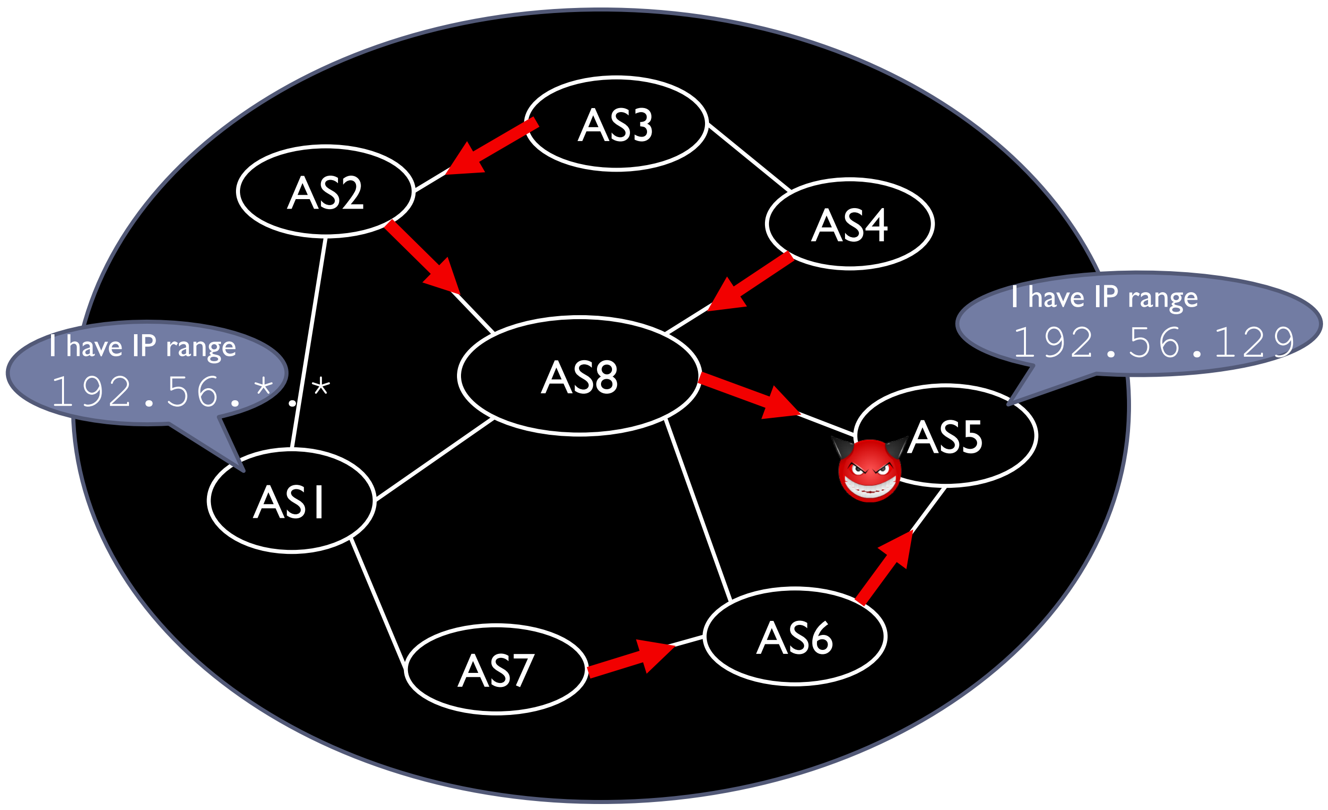
Prefix Hijacking


AS3
AS2
AS4
AS8
AS5
AS1
AS7
AS6
I have IP range 192.56.*.*
I have IP range 192.56.129
Route by most specific prefix!

Routing table

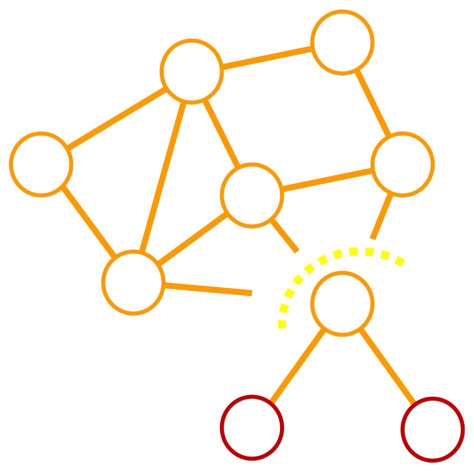
```
192.56.*.* to AS1
192.56.129.* to AS
5
```

Prefix Hijacking

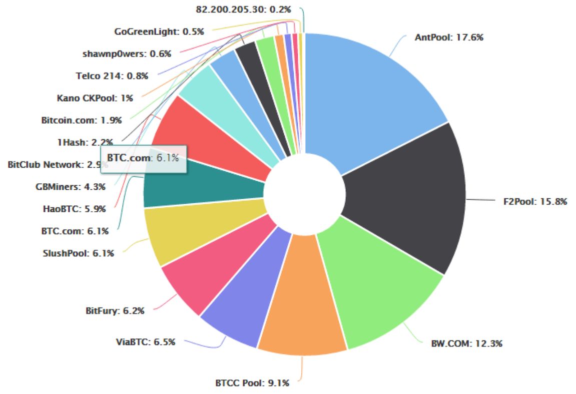


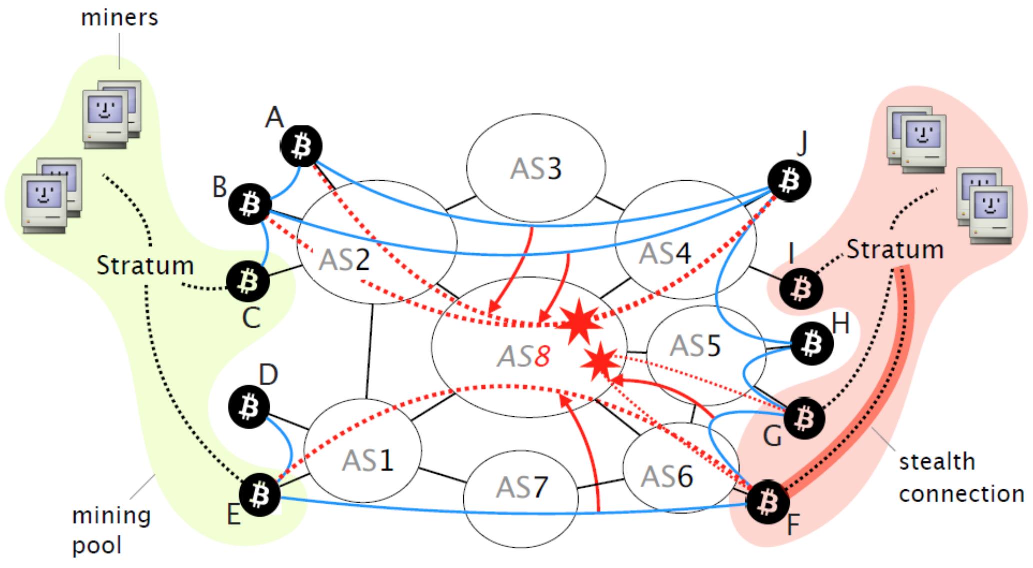
Consequences of disrupting connectivity

- Transactions cannot be sent (DoS)
- Pool rewards can be stolen
- Transactions on one side of the network are reversed
  - Miners lose revenue
  - Double spending attacks against merchants
- Mining power subverted to attack
  - double spend
  - selfish mining
  - Censorship via empty blocks



Mining pools



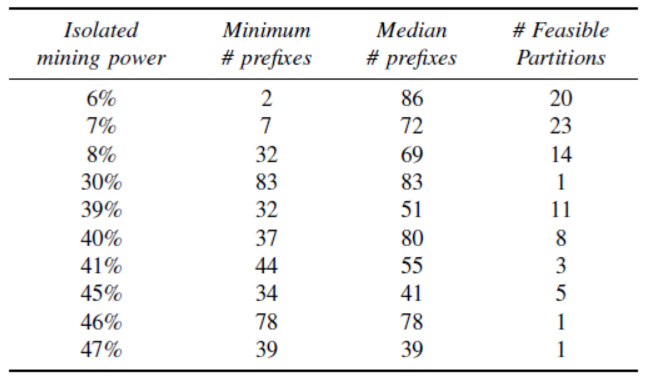


Attack 1: Partitioning Bitcoin

- Deduce gateway nodes for pools
  - Stratum servers
  - Block propagation data
- Combine with routing data

Factors that aid attacker:

- Mining power is held by few nodes
- Only 7% of nodes are advertised in /24 prefixes



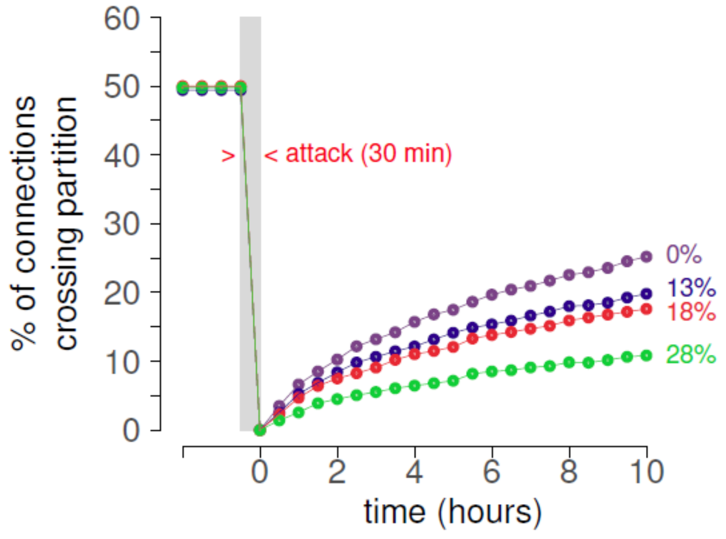
Isolated mining power	*Minimum # prefixes*	*Median # prefixes*	*# Feasible Partitions*
6%	2	86	20
7%	7	72	23
8%	32	69	14
30%	83	83	1
39%	32	51	11
40%	37	80	8
41%	44	55	3
45%	34	41	5
46%	78	78	1
47%	39	39	1

TABLE IV: This table lists all partitions that can be created based on our inferred topology. The leftmost column indicates the portion of mining power contained within the isolated set and the rightmost the number of different combinations of pools that could form it.

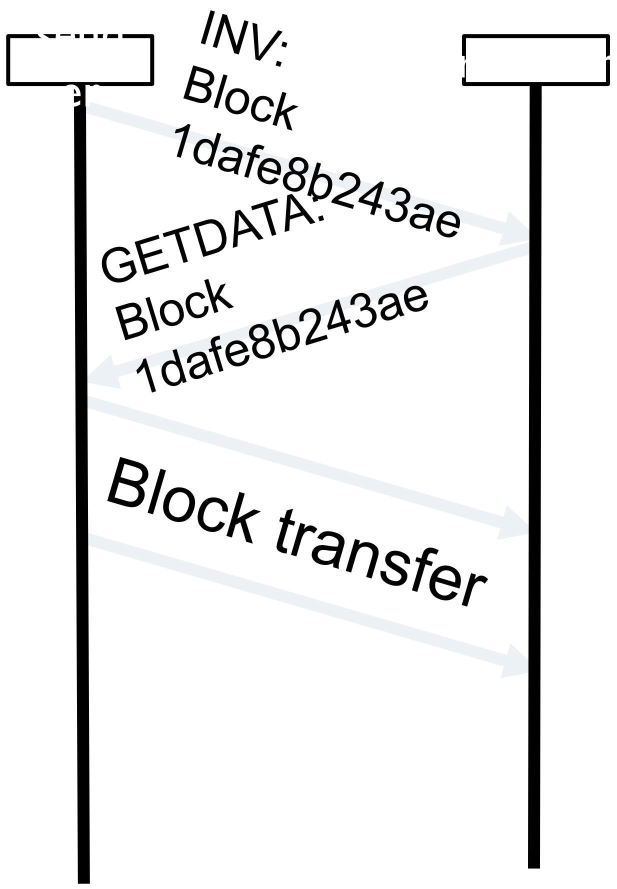
Partitions need to be perfect

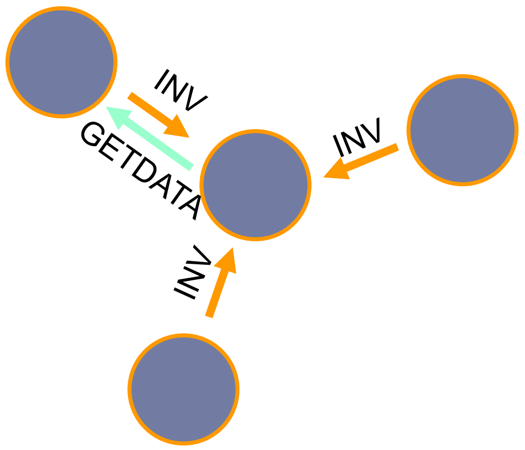
- 1050 bitcoind nodes running on VMs on emulated network.
  - With churn (as measured on network)

- Connections return slowly
- BUT a few connections suffice.



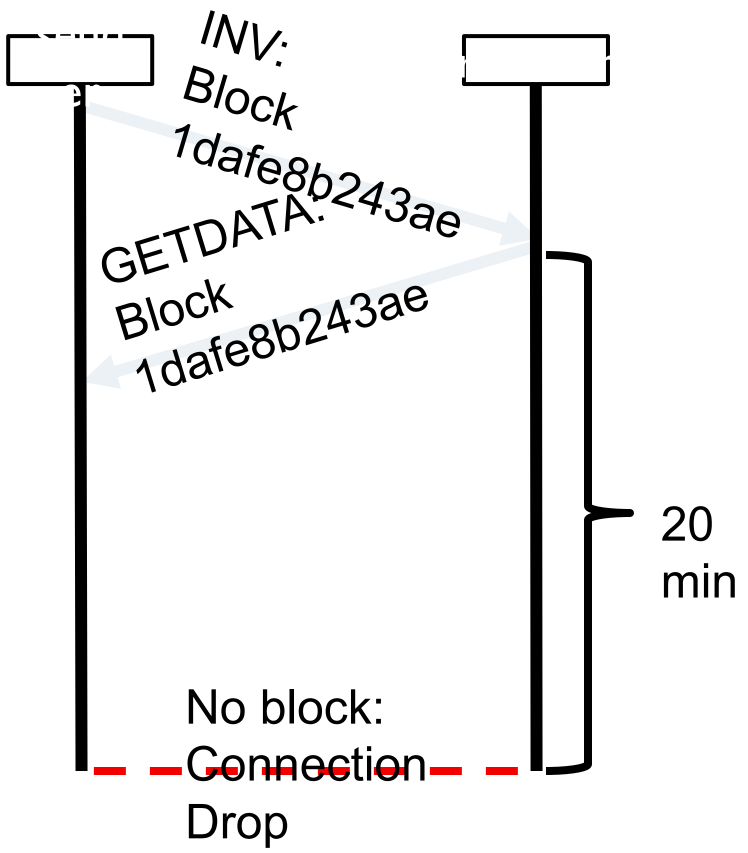
Blocks Propagation Mechanics



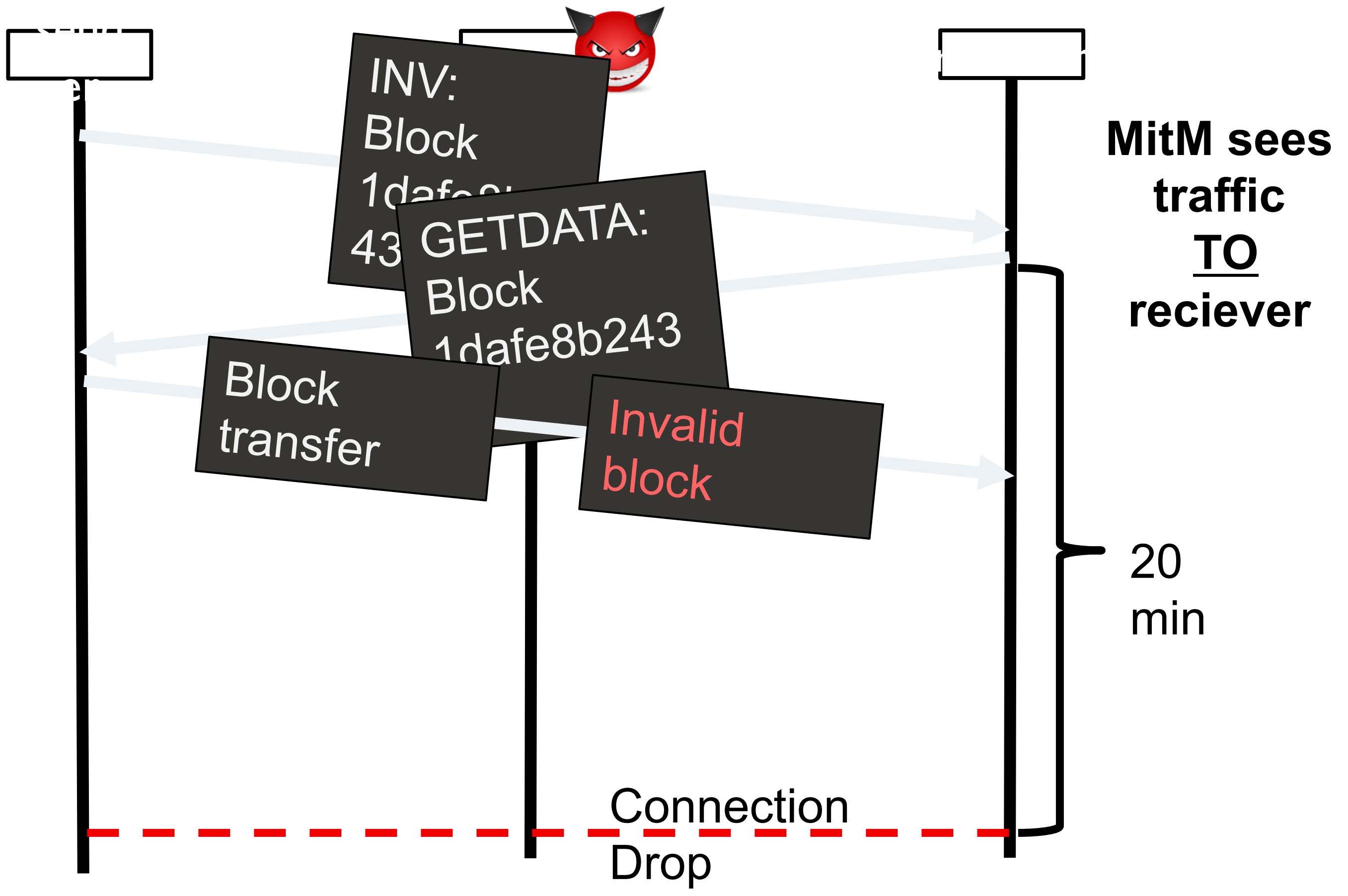


Traffic is not encrypted!

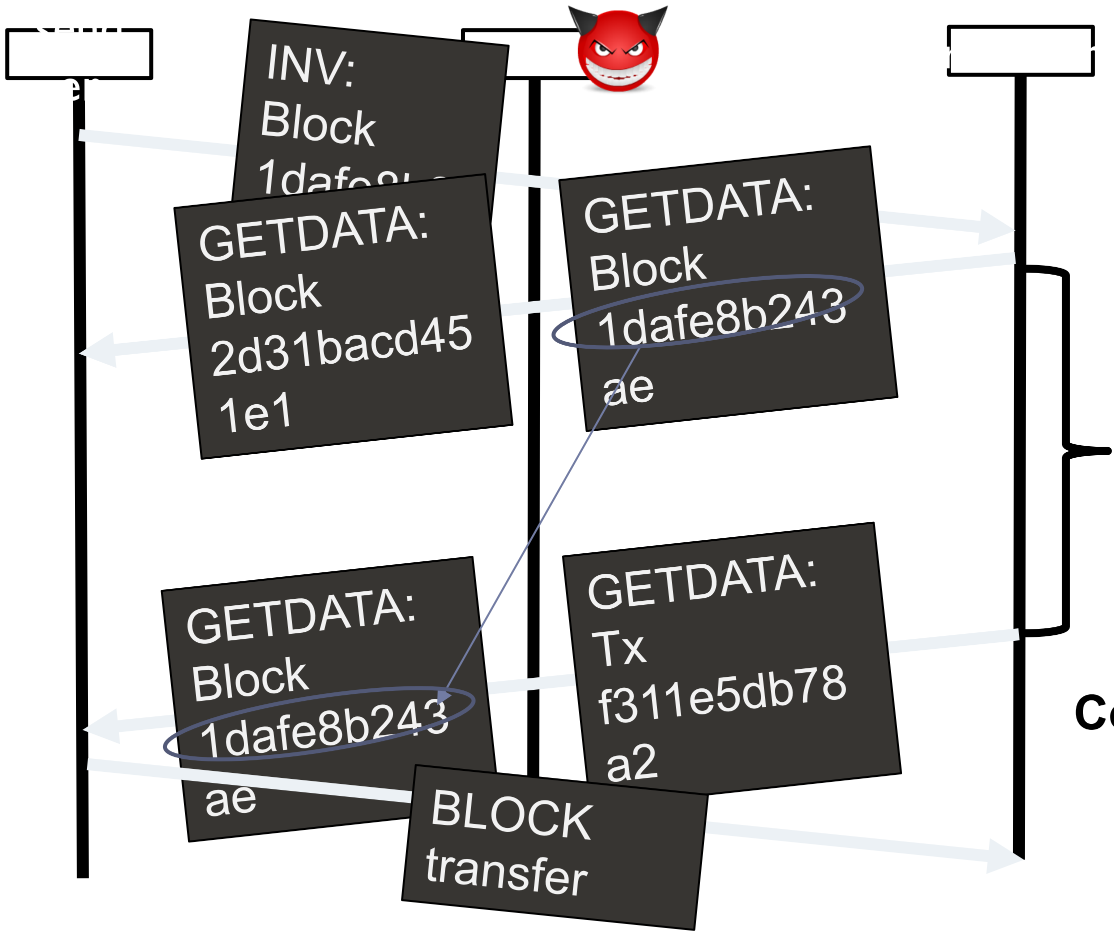
Blocks Propagation Mechanics



Attack 2a: MitM block delay attack



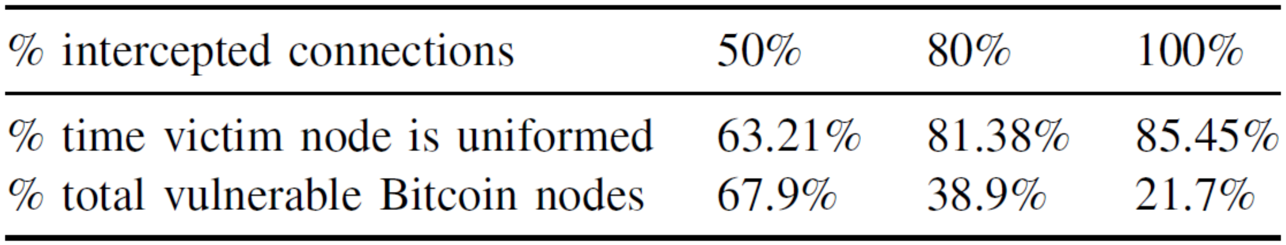
Attack 2b: MitM block delay attack



MitM sees traffic <u>FROM</u> reciever

Connection not lost. Repeat attack!

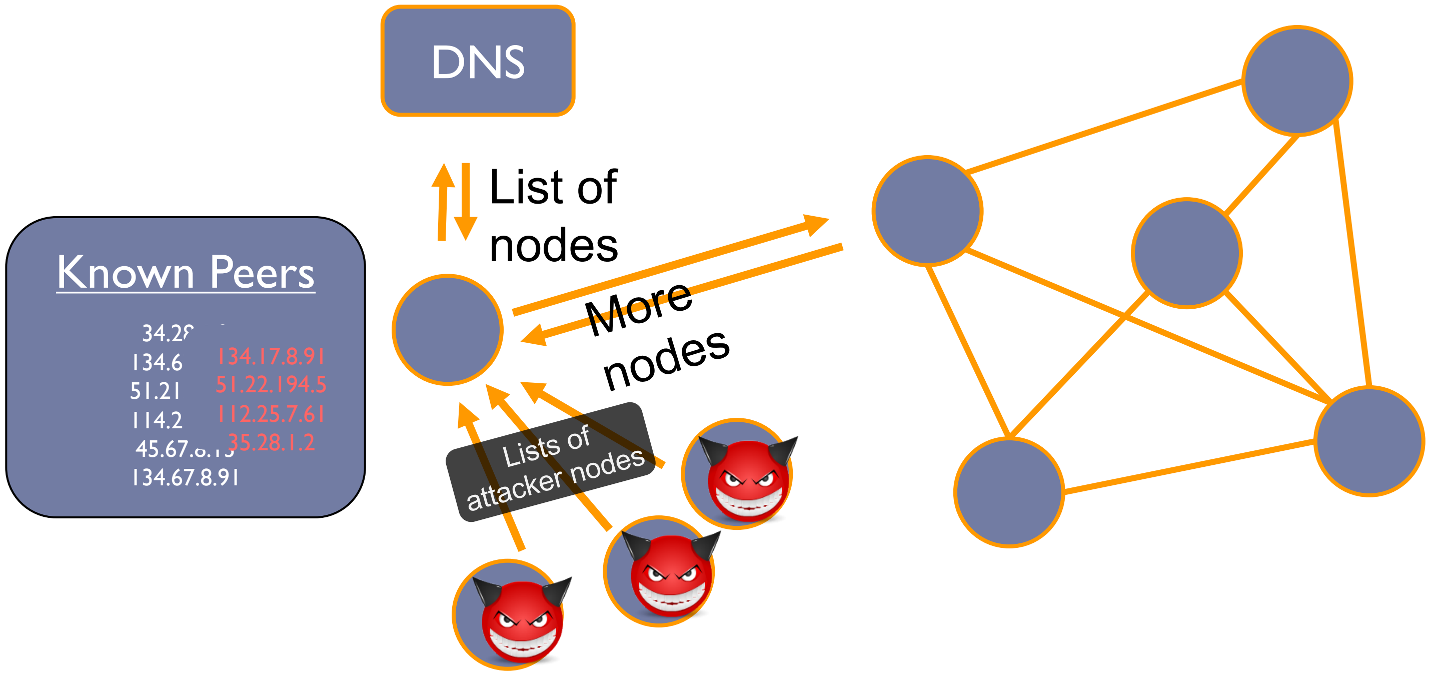
- We performed this MitM attack on our own node
- Passive AS (no hijacking)



% intercepted connections	50%	80%	100%
% time victim node is uniformed	63.21%	81.38%	85.45%
% total vulnerable Bitcoin nodes	67.9%	38.9%	21.7%

- Uninformed node wastes mining power
- Susceptible to 0-conf attacks

Eclipse attacks



Summary

- Bitcoin is considered secure as long as nodes can communicate
- Communication is easily disrupted
- Mitigation techniques in the papers
    - Much more needed!