eclipse and routing attacks on peer-to-peer networks

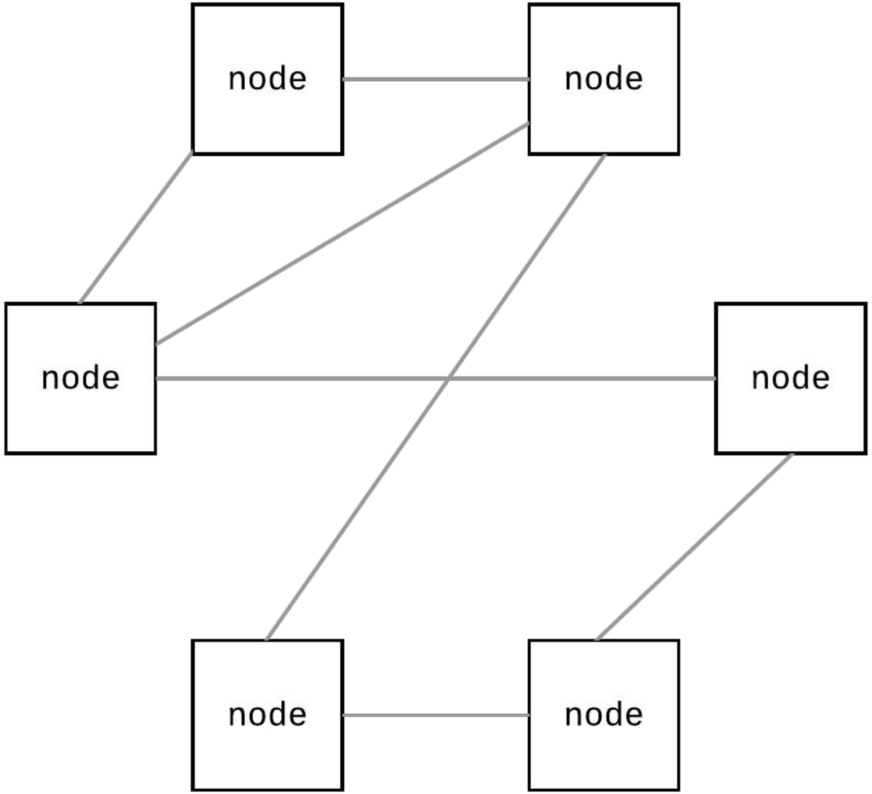
Péter Garamvölgyi

outline

- **distributed ledgers**
- eclipse attacks
- routing attacks
    - partitioning attacks
    - delay attacks

Bitcoin - setup

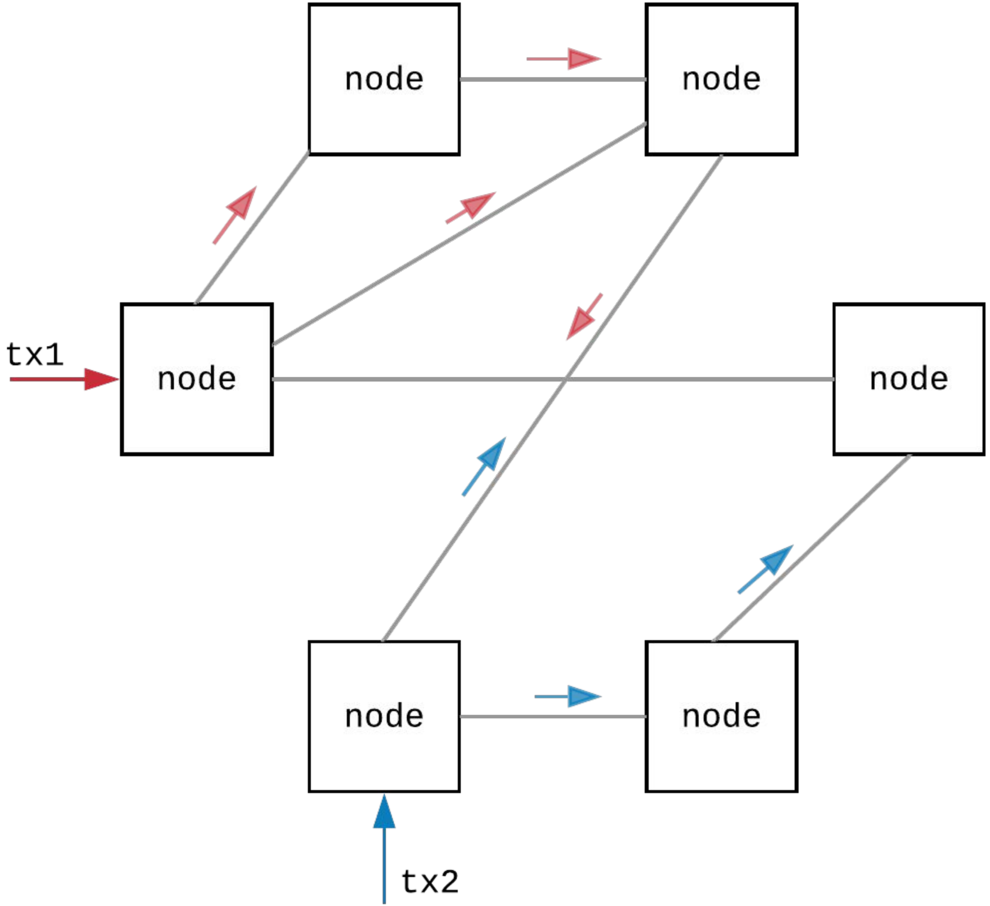
- p2p network of independent nodes



Bitcoin - setup

- p2p network of independent nodes

- async transactions
- gossip protocol

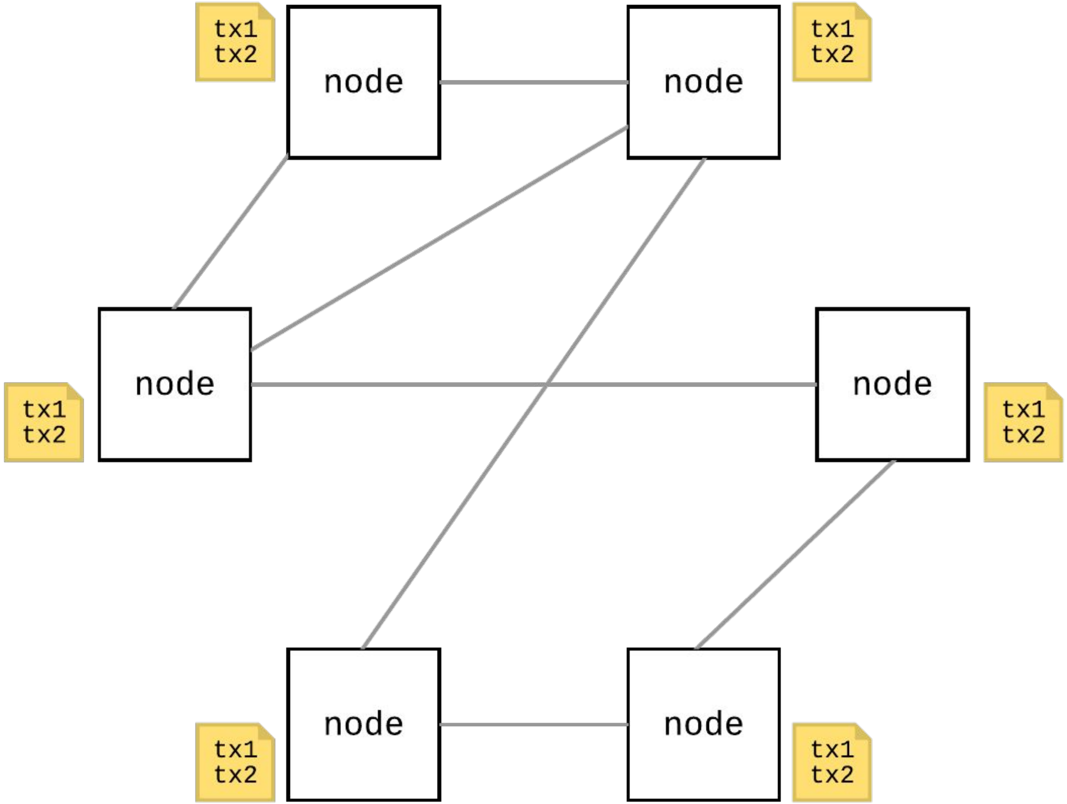


Bitcoin - setup

- p2p network of independent nodes

- async transactions
- gossip protocol

- matching ledgers*



* in practice, this is not always an exact match

the challenge of distributed ledgers

come up with an **algorithm for each node** so that they **reach consensus** on

1. which transactions are **valid**?
2. what is the (partial) **order** of the transactions?

honest nodes should end up having the same ledger.

malicious nodes should not be able to break the system.

outline

- distributed ledgers
- **eclipse attacks**
- routing attacks
    - partitioning attacks
    - delay attacks

the Bitcoin p2p protocol

- p2p gossip network on TCP:8333
- 8 outgoing, up to 125 incoming connections by default
- no encryption or integrity checks
- message-based protocol
  - ➜ ADDR — "I know about these nodes: ..."
  - ➜ INV — "I have these blocks/transactions: ..."
  - ➜ GETDATA — request a single block or transaction by hash
  - ➜ BLOCK — send a block in response to GETDATA

https://en.bitcoin.it/wiki/Protocol_documentation

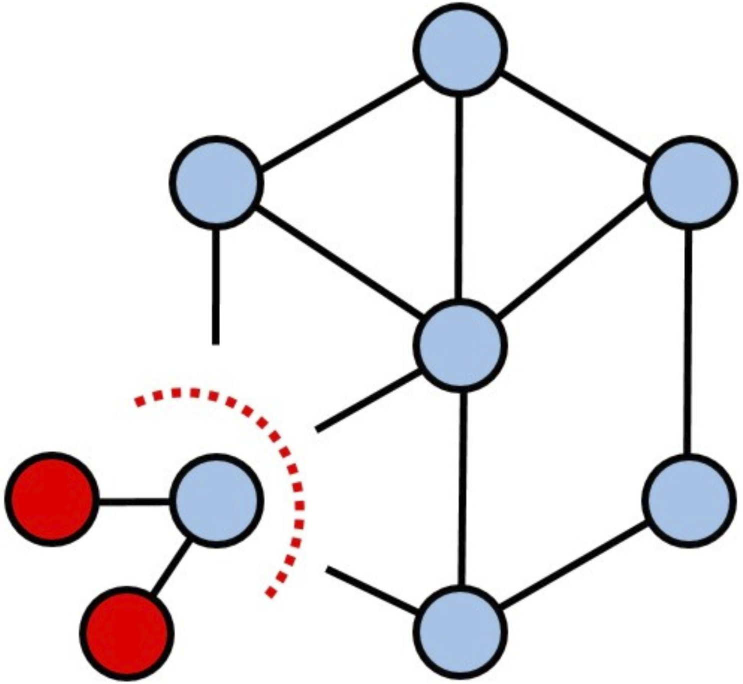
the Bitcoin p2p protocol

- keeping track of peers
  - **tried table**: addresses with previous connection
  - **new table**: addresses from peers (no connectivity check)
- peer selection
  - randomly select table
  - randomly select address (bias for fresher addresses)
  - connect or try again

https://en.bitcoin.it/wiki/Protocol_documentation

eclipse attack - overview

1. fill **tried table** with attacker addresses
   – use unsolicited incoming connections
2. fill **new table** with trash addresses
   – use unsolicited ADDR messages
3. wait for victim to **restart**
   – outages, software updates, DDoS
4. occupy all **incoming** connections



Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. (2015). Eclipse Attacks on Bitcoin's Peer-to-Peer Network

eclipse attack

- relevance
  - **botnet attack**: 4600 addresses
  - **infrastructure attack**: 16,000 addresses in 63 groups

- countermeasures
  - **feeler connections**: check addresses in new table
  - **anchor connections**: persist some connections between restarts
  - ban large unsolicited ADDR messages
  - anomaly detection

Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. (2015). Eclipse Attacks on Bitcoin's Peer-to-Peer Network

outline

- distributed ledgers
- eclipse attacks
- routing attacks
    - **partitioning attacks**
    - delay attacks

BGP hijacking

- ASes may announce IP ranges they do not own

    e.g. AS wants to attract traffic sent to 100.0.0.0/16

        (a) announce 100.0.0.0/16

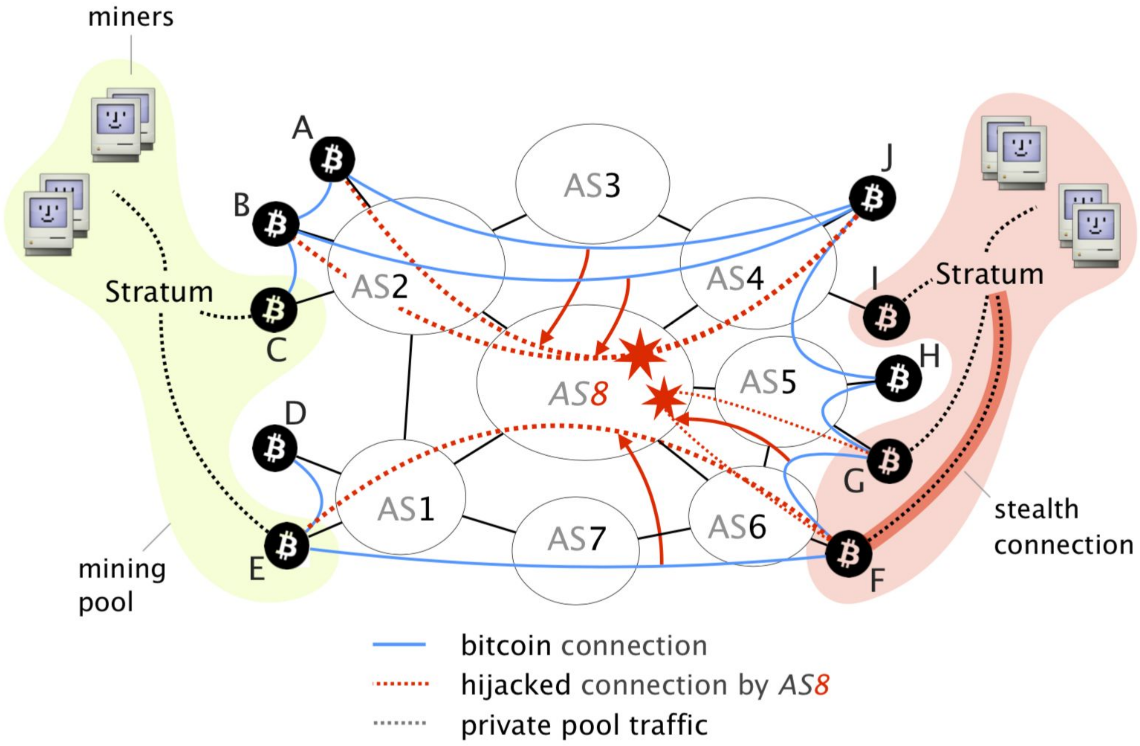
        (b) announce a more specific range, e.g. 100.0.0.0/17, 100.0.128.0/17

    announcements more specific than /24 are usually dropped

partitioning attack - overview

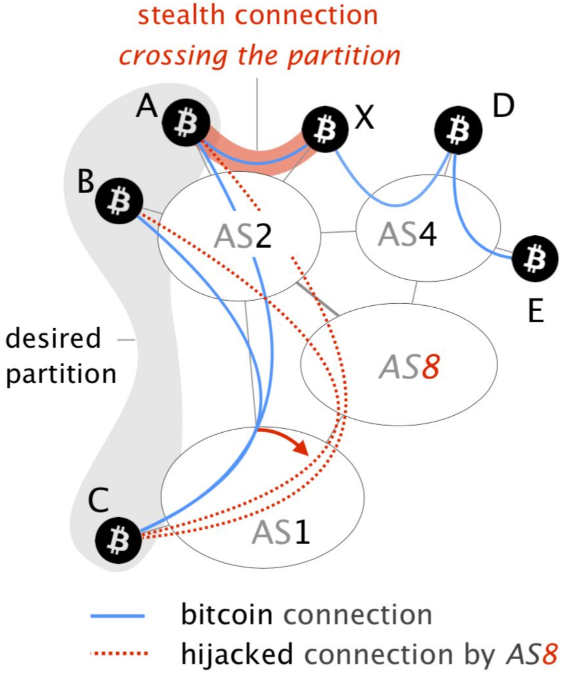
`P = {A, B, C, D, E, F}`

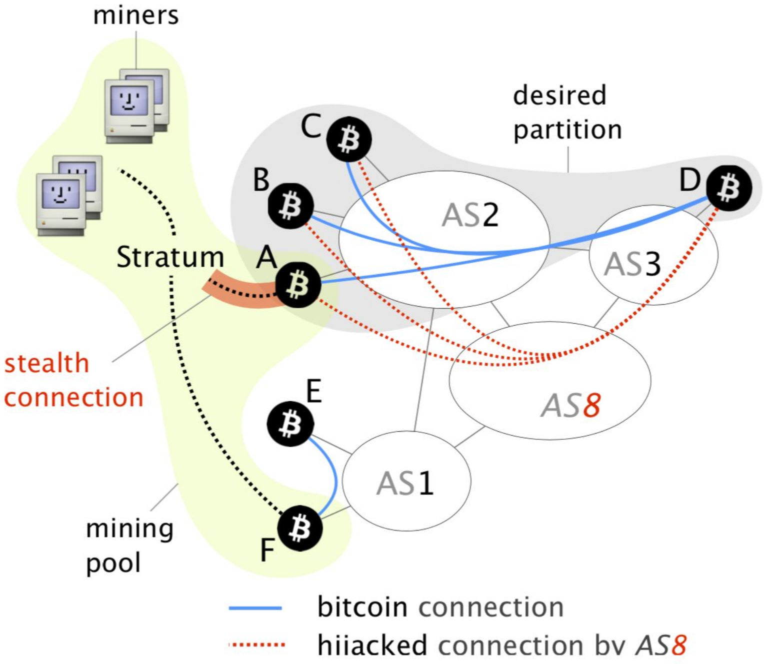
1. divert traffic
2. identify relevant packets
3. drop packets
4. isolate leaks



Apostolaki, M., Zohar, A., & Vanbever, L. (2017). Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. 2017

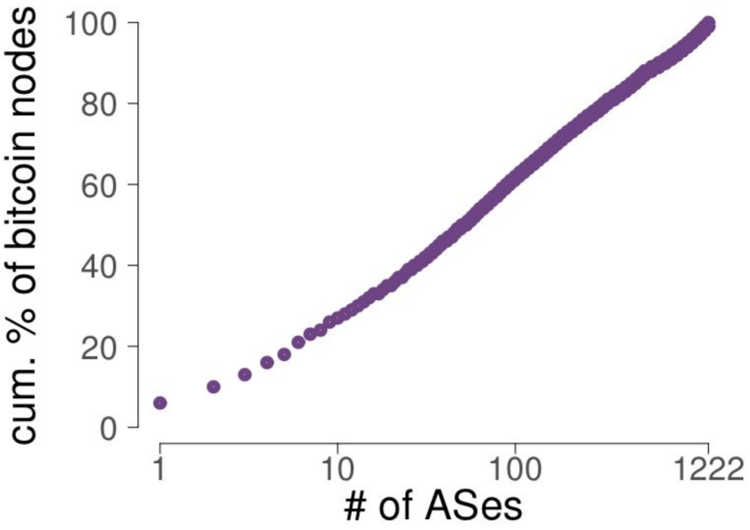
isolating leaks

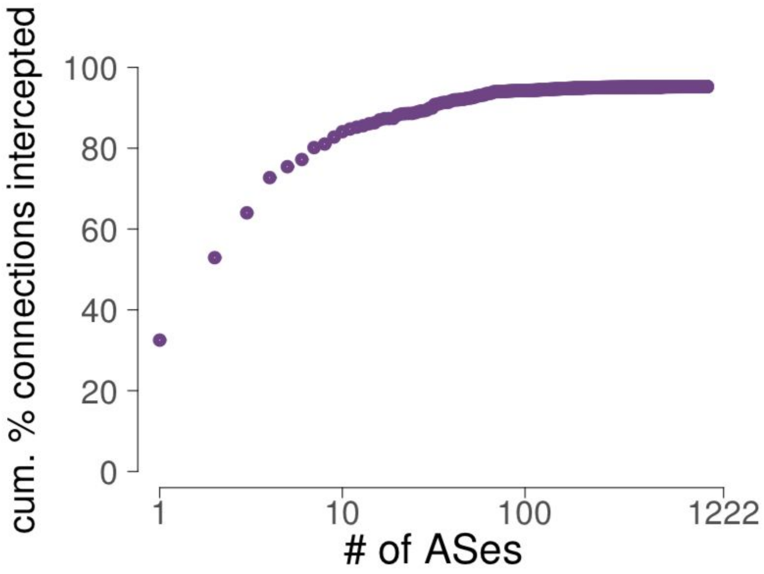




Apostolaki, M., Zohar, A., & Vanbever, L. (2017). Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. 2017

partitioning attack - relevance





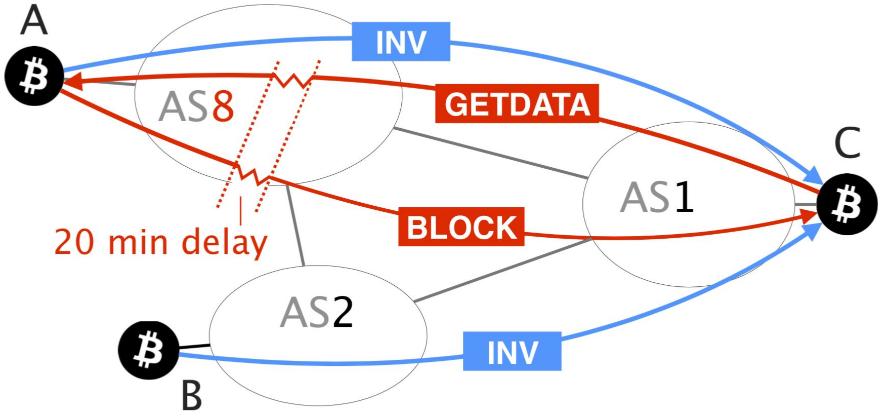
Apostolaki, M., Zohar, A., & Vanbever, L. (2017). Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. 2017

outline

- distributed ledgers
- eclipse attacks
- routing attacks
    - partitioning attacks
    - **delay attacks**

delay attack - overview

- slow down block propagation
- tamper with traffic in a way that
    1. prevents node from receiving correct information
    2. but keeps connection alive

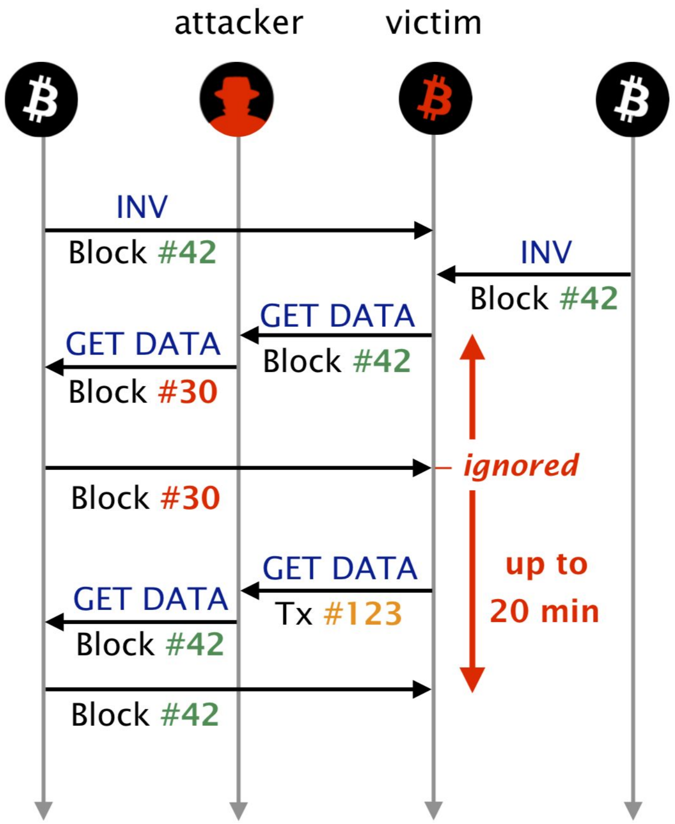


Apostolaki, M., Zohar, A., & Vanbever, L. (2017). Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. 2017

delay attack - overview

a. intercept outgoing connection (block #42)
   - ➜ change block hash in GETDATA to #30
   - ➜ victim gets wrong block (#30); keeps waiting
   - ➜ change another GETDATA to #42 this time
   - ➜ victim gets #42 with a large delay

- why not just drop?
- why change the second time?



Apostolaki, M., Zohar, A., & Vanbever, L. (2017). Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. 2017

delay - impact (single node)

% 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%

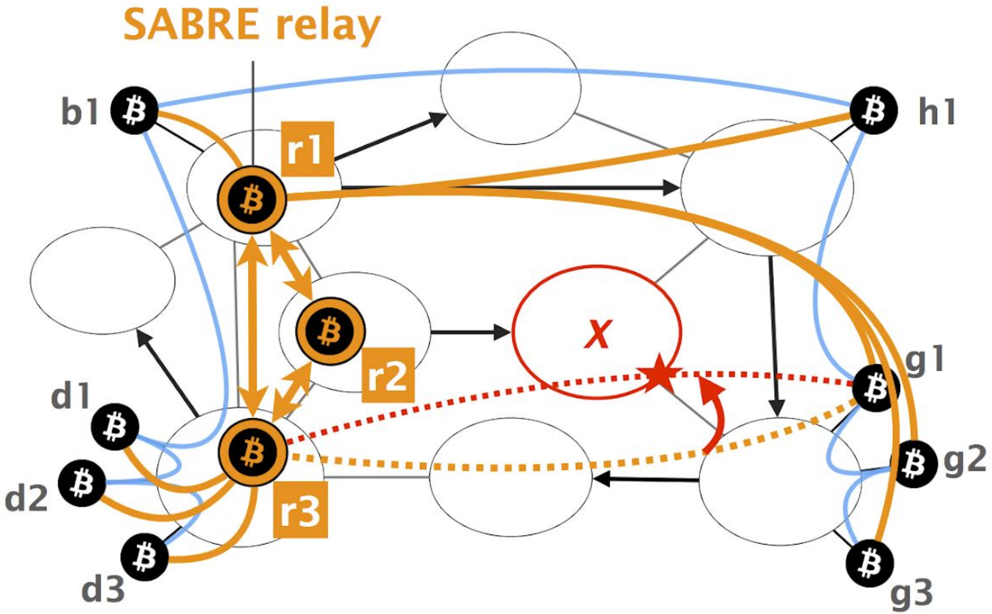
Apostolaki, M., Zohar, A., & Vanbever, L. (2017). Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. 2017

countermeasures

- short-term
  - ➡ increase diversity of node connections, prefer /24
  - ➡ use traceroute, check BGP traffic, detect anomalies
  - ➡ prevent a single AS from appearing in all paths
- long-term
  - ➡ use encryption and/or integrity checks (BIP-151)
  - ➡ use port negotiation or randomized port
  - ➡ request blocks on multiple connections

Apostolaki, M., Zohar, A., & Vanbever, L. (2017). Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. 2017

SABRE



Apostolaki, M., Marti, G., Müller, J., & Vanbever, L. (2018). SABRE: Protecting Bitcoin against Routing Attacks

references

Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. (2015). Eclipse Attacks on Bitcoin's Peer-to-Peer Network
https://eprint.iacr.org/2015/263.pdf

Apostolaki, M., Zohar, A., & Vanbever, L. (2017). Hijacking Bitcoin: Routing Attacks on Cryptocurrencies
https://btc-hijack.ethz.ch/files/btc_hijack.pdf

Apostolaki, M., Marti, G., Müller, J., & Vanbever, L. (2018). SABRE: Protecting Bitcoin against Routing Attacks
https://arxiv.org/pdf/1808.06254

Bitcoin Wiki
https://en.bitcoin.it/wiki