Announcement

- Mini-project
  - Due: 16 April 23:59 (Sun)

- Take-home Exam 2
  - Plan to publish 7 April 2023 (Fri)
  - Due: 24 April 2023 23:59 (Mon)
  - Covers Week 7-12
  - Lecture materials as well as papers.
  - Open book.
  - No discussion or collaboration is allowed.

Blockchain Security

CS 5321 Week 12

Materials are provided by

Dr. Muoi Tran

https://www.comp.nus.edu.sg/~muoitran/

Outline

- Cryptocurrency/blockchain ***overview***
- ***Security*** research of blockchain ***networks***
- ***Partitioning attacks*** against Bitcoin peer-to-peer networks
- ***Summary***

Cryptocurrency/blockchain ***overview***

Cryptocurrency: digital currency with strong cryptography and decentralization guarantees

Cryptocurrency

Digital currency

- Digital assets (e.g., coins)
- Allow online transactions

Cryptography

- Proof of coin ownership and ownership transfer
- No double-spending

Decentralization

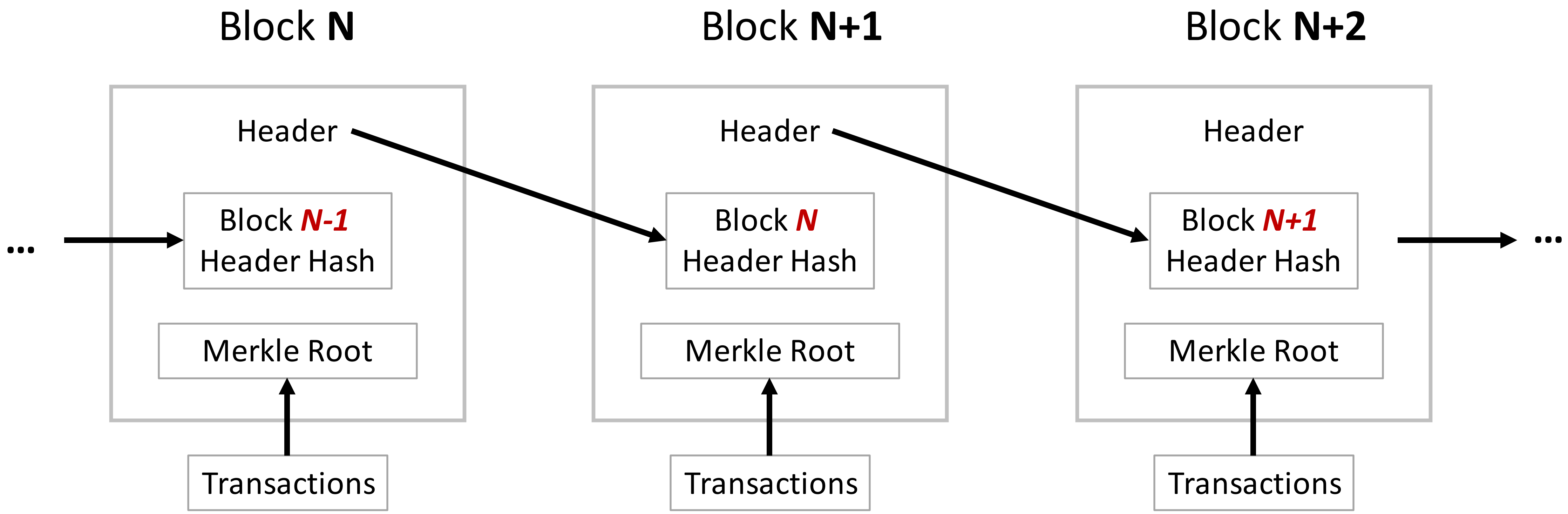
- No centralized authority
- Global transaction database is available to everyone

Cryptocurrency relies on ***blockchain*** technologies for decentralization

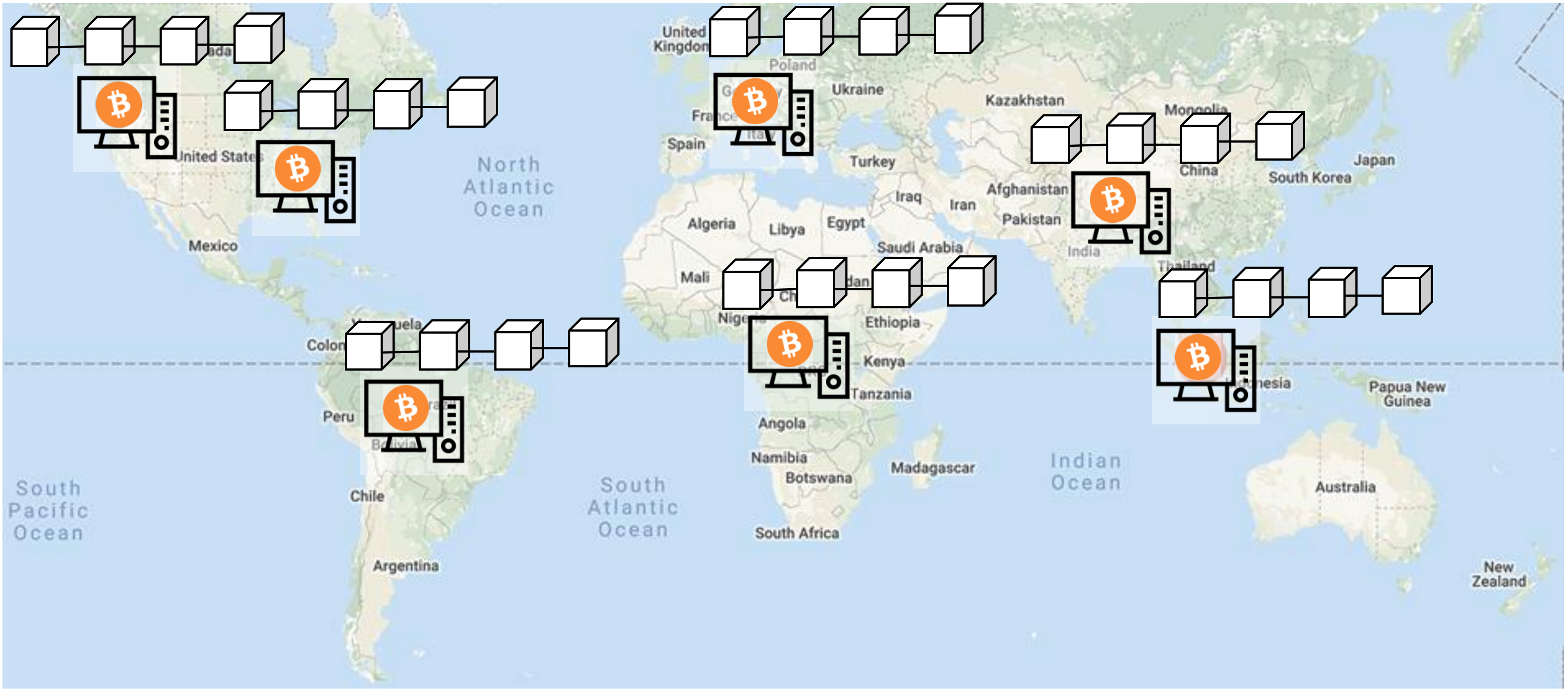
- Digital currencies with cryptography foundations are ***not new***
  - ✓Chaum82: blind signatures for e-cash
  - ✓Chaum85: retroactive double spender identification
  - ✓Camenisch05: compact offline e-cash

- ***Bitcoin***: the ***first*** cryptocurrency
  - ✓Fully decentralized digital currency system
  - ✓Released by Satoshi Nakamoto in 2008
  - ✓Innovation: the ***blockchain***

Blockchain is a ***linked list*** of ***blocks*** of all transactions



The blockchain is maintained by many ***distributed nodes***



Nodes follow ***pre-defined consensus*** to agree on a ***single*** blockchain

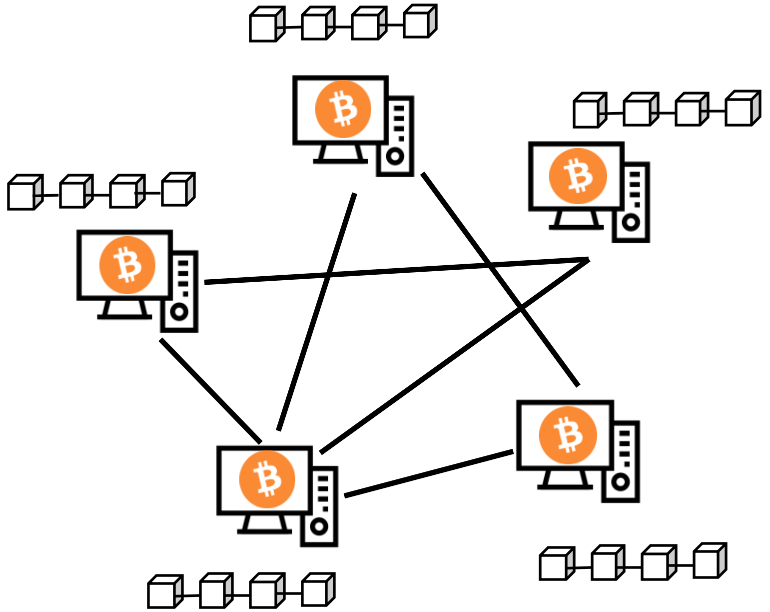
- ***All nodes*** can generate transactions:
  - ✓***Unconfirmed*** transactions are not in any block
  - ✓***Confirmed*** transactions are included in a block

- ***Miners*** generate new blocks from unconfirmed transactions
  - ✓Must own some ***powers*** (e.g., <u>computational</u> (***PoW***) or <u>stake</u> (***PoS***))
  - ✓Incentives: new coins + transaction fees

- Nodes follow ***consensus*** rules:
  - ✓Only ***valid*** blocks and transactions are stored
    - ✓Valid block must contains proof of work
  - ✓Only ***one block*** at any height is accepted (e.g., ***longest*** chain in Bitcoin)

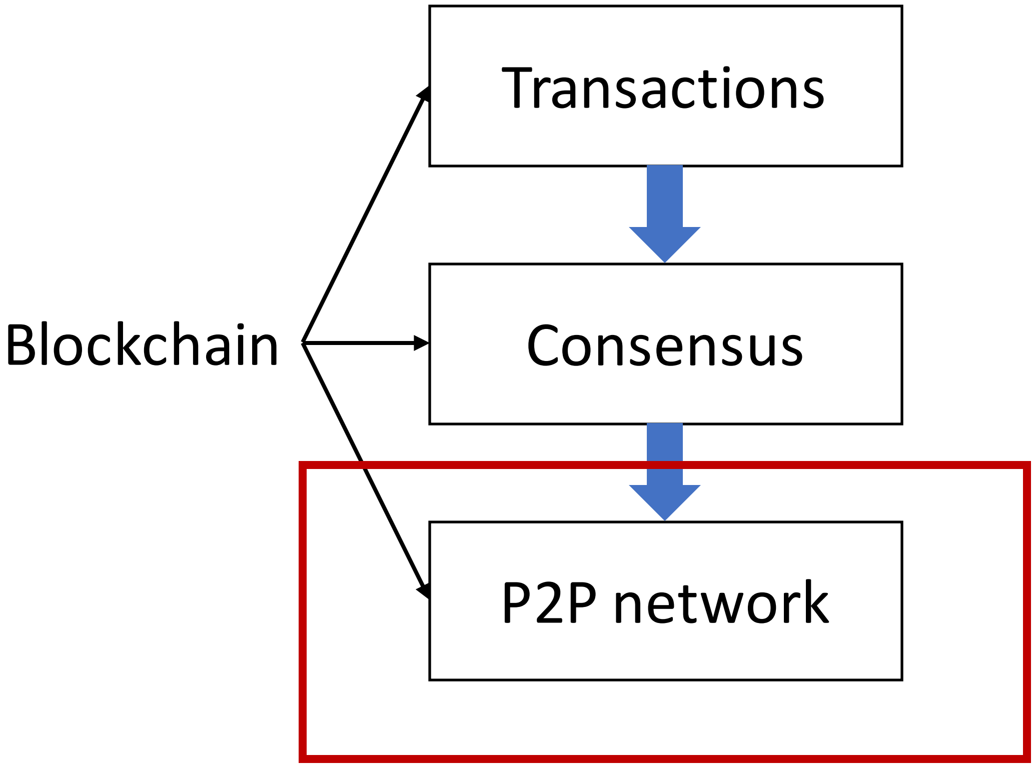
  => a ***single*** blockchain will eventually remain

Blockchain relies on underlying ***peer-to-peer networks*** for data propagation



- Nodes connect via a ***P2P network***
- New transactions/blocks are propagated via ***gossiping*** protocols
- Upon receiving a block /transaction, nodes ***verify*** before forwarding it
- P2P network can be ***permissionless*** or ***permissioned***

Why blockchain ***network security*** matters?



- Blockchain in ***three*** layers:
  - ✓Transactions/scripts
  - ✓Consensus/mining
  - ✓P2P networks
- Security guarantees of a layer ***depend*** on a ***lower*** layer:
  - ✓Example 1: a *valid* transaction may be *rejected* if malicious miners *censor* it
  - ✓Example 2: blockchain will have *two versions* if network is *split* into two

Security research of blockchain ***networks***

Topics in blockchain network security

Blockchain Networks

Block propagation

- Delayed delivery
- Selfish-mining attacks
- Incentives

Transaction propagation

- Denial-of-Service attacks
- Censorship attacks
- Privacy

Network P2P topology

- Peer discovery
- Mapping network
- Eclipse attacks

Block propagation: ***challenges***

- ***Delayed*** blocks lead to different blockchains (i.e., forks)
  - ✓ Non-attacks: block data is large, network latency, churn, …
  - ✓ MitM attackers purposedly delay block delivery

- ***Selfish mining***: Malicious miners do not release new blocks immediately to gain advantages in mining next block

- There is ***no incentives*** for ***non-miner*** nodes to propagate blocks => P2P network becomes more centralized

Transaction propagation: ***challenges***

- Transaction propagation can be abused for ***DoS attacks***
  - ✓Attackers ***flood transactions*** to the network, which are propagated by ***all nodes***
  - ✓Only require ***minimal transaction fees***

- Transactions can be ***censored*** or ***not propagated***
  - ✓Miners do ***not include*** valid transactions that they don't like in new blocks
  - ✓Miners ***do not propagate*** transactions with high fees to other miners

- The ***origin*** of the transaction can be ***deanonymized***
  - ✓ Transactions are ***grouped*** together and ***linked*** to the origin (e.g., IP address)

P2P network topology: ***challenges***

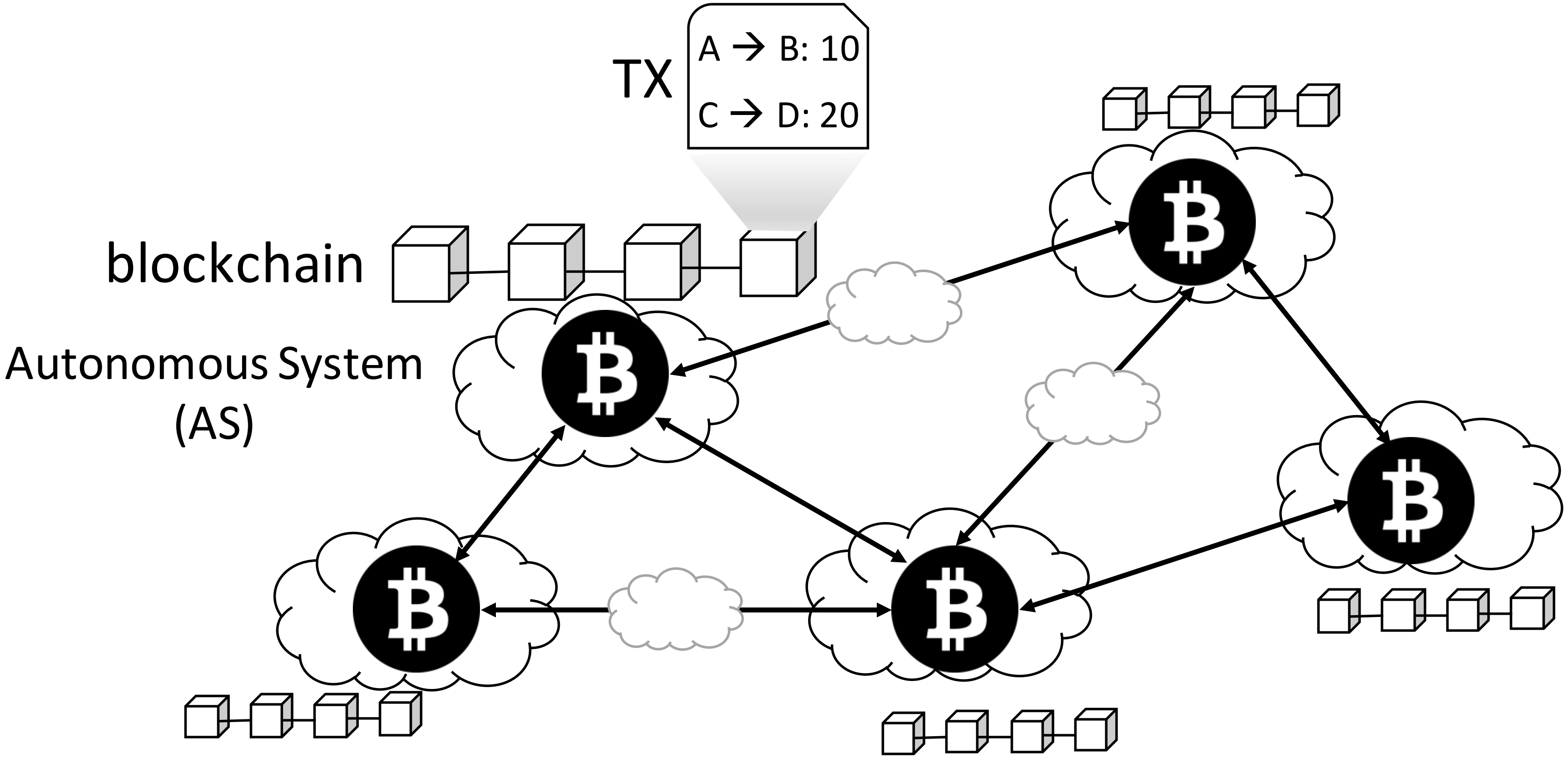
- How to ***discover*** other peers when a node ***first joins*** the network?
  - ✓E.g., Bitcoin bootstrap nodes are maintained by developers
  - ✓E.g., Ethereum uses Distributed Hash Table
  - $\Rightarrow$ ***not fully decentralized***, or ***no robustness guarantee***

- How to prevent ***mapping*** the full network topology?
  - ✓Network topology shows which node is connecting to which node
    => reveal the ***influenced nodes*** to the attackers

- ***Eclipse*** attacks split the P2P network:
  - ✓Consensus cannot be reached
  - ✓Users cannot operate on transactions

Partitioning attacks against Bitcoin peer-to-peer networks

Bitcoin P2P network under the hood

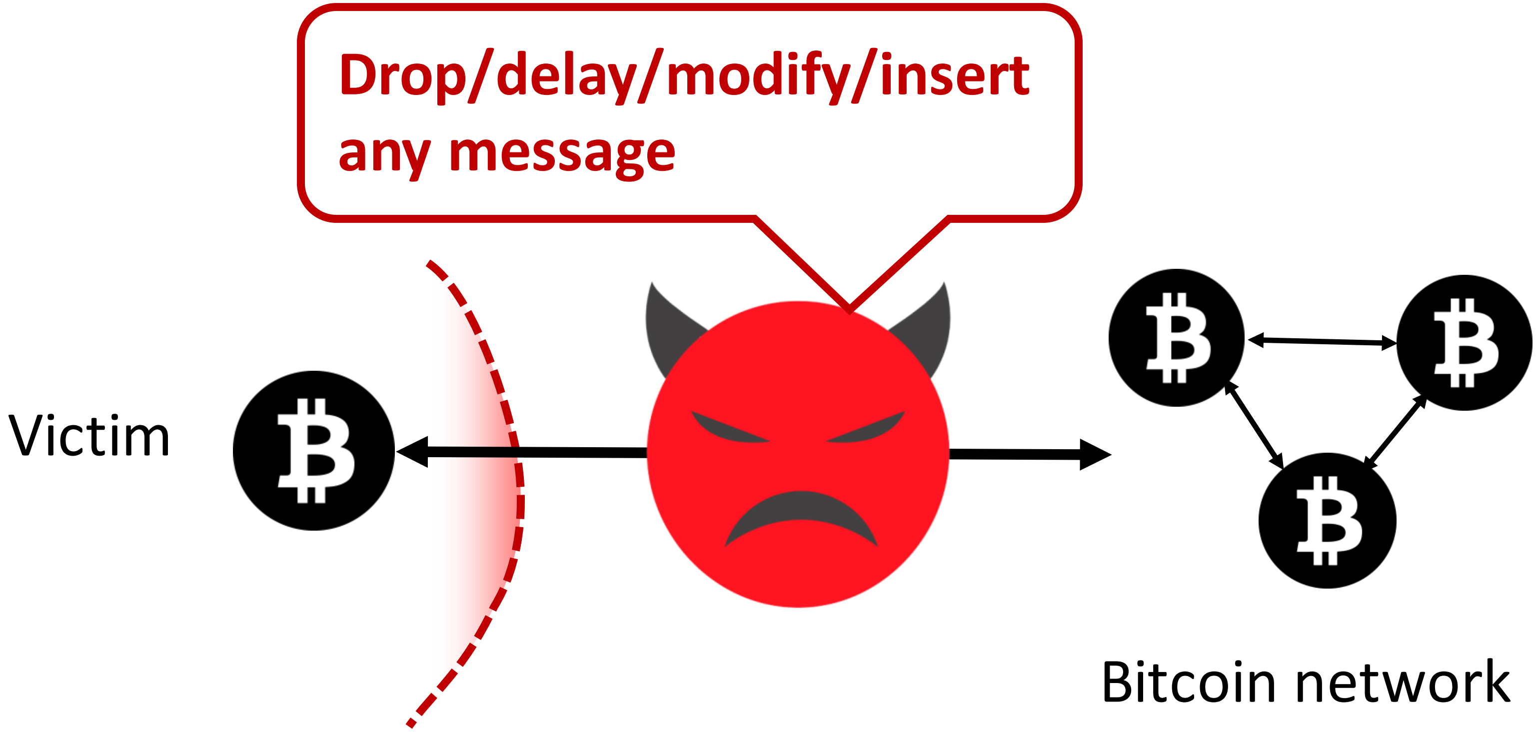


Bitcoin consensus rules

Peer-to-peer network

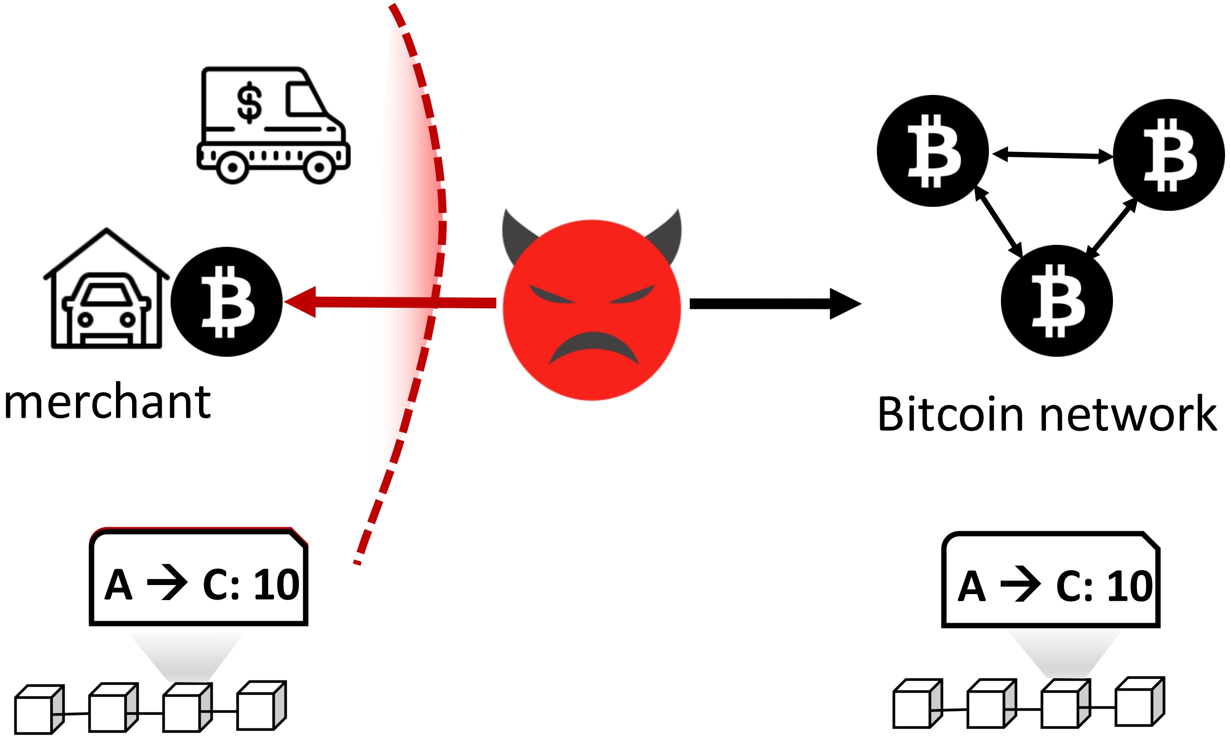
Routing network

Partitioning attacks against Bitcoin network



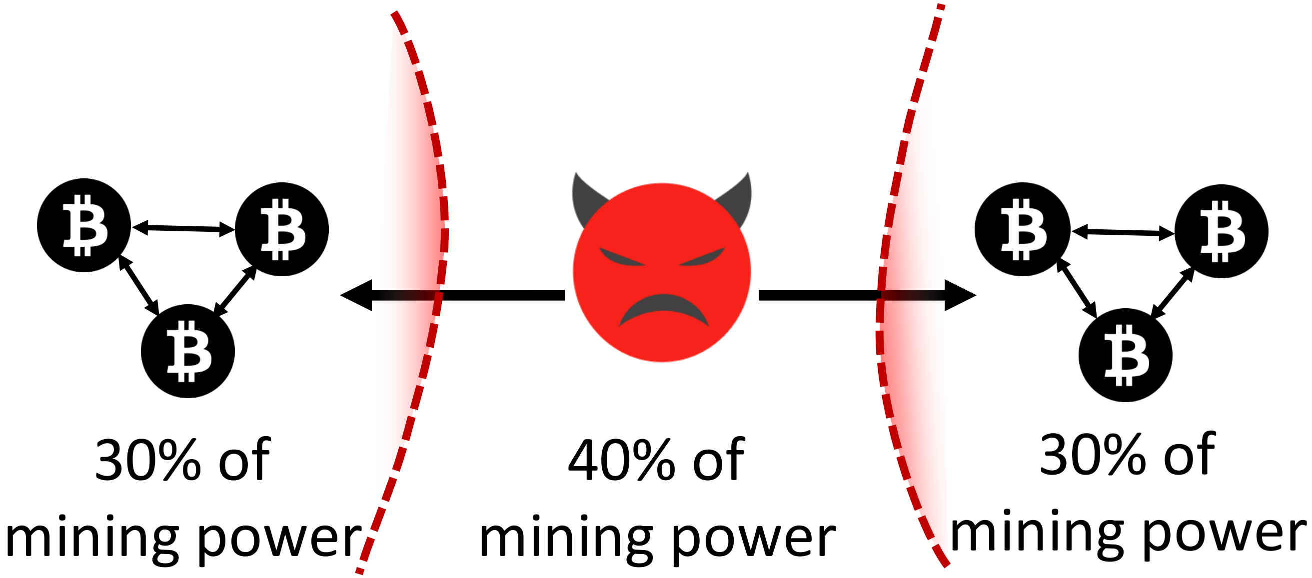
Partitioning attacks: isolate one or more nodes from the rest of network

Motivation of partitioning attacks



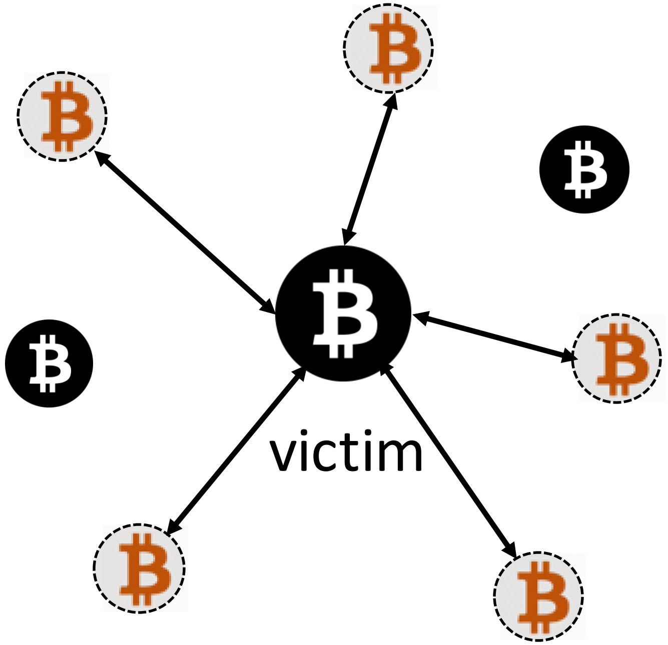
- Double-spending attack
  - ✓ Attacker pays merchant with **(A→B: 10)** transaction
  - ✓ Attacker pays herself with **(A → C: 10)** transaction
  - ✓ Mines N blocks to confirm the transaction
  - ✓ Rest of network accepts **(A → C: 10)** transaction and discards **(A→B: 10)**

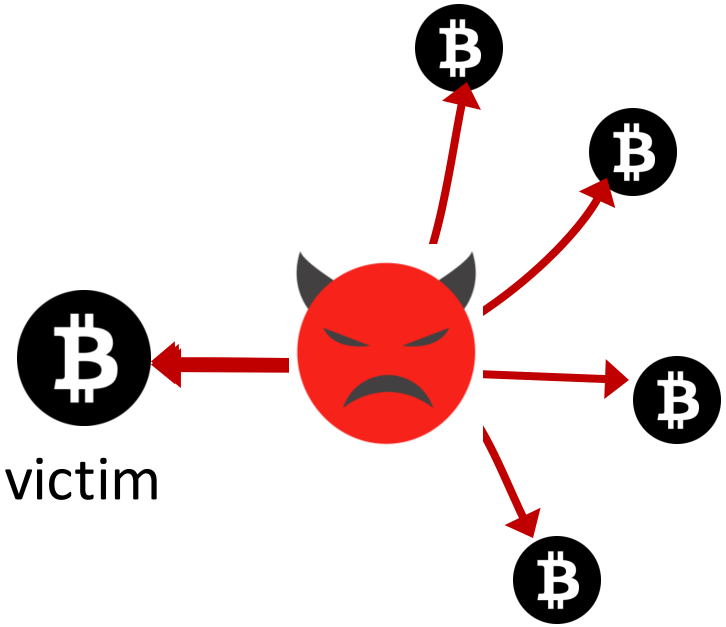
Motivation of partitioning attacks (cont.)



- 51% attack
  - ✓ Attacker outcompetes each partitioned miner, even with only 40% mining power.
  - ✓ Attacker's blockchain will become the longest chain
- Many other attacks:
  - ✓ Selfish-mining
  - ✓ Censoring transactions
  - ✓ Attack layer-2 protocols
  - ✓ …

How to partition a Bitcoin node?





- ***Strategy 1***: Influence victim to only connect to adversary-controlled peers (e.g., Eclipse, Erebus attack)
- ***Strategy 2***: Adversary places herself in all legitimate peering connections (e.g., Bitcoin hijacking attack)

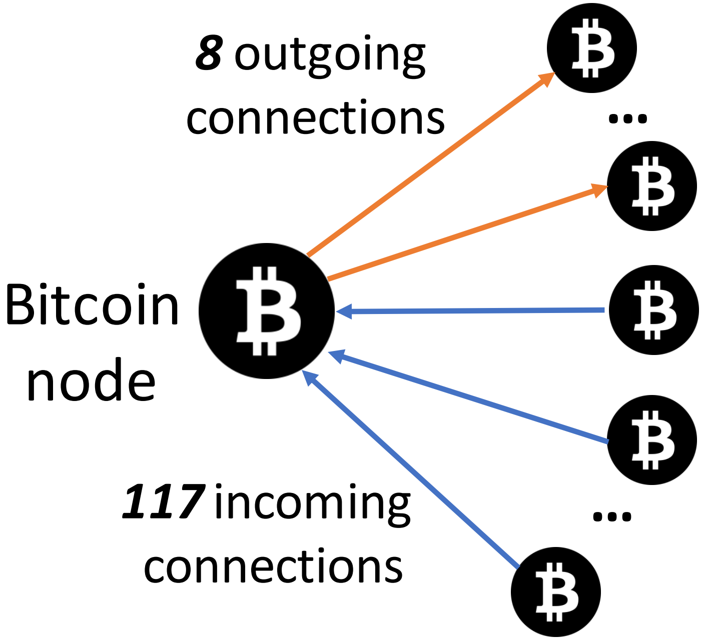
Eclipse attack

- Paper: "*Eclipse Attacks on Bitcoin's Peer-to-Peer Network*" by Heilman et al. [**USENIX Security 2015**]

- Threat model
  - ✓Attacker's capability: Control a botnet of ~3,000 IP addresses
  - ✓Attacker's goal: Influence all of victim's connections to be made to adversary-controlled bots

- Assumptions:
  - ✓Victim run Bitcoin version 0.9.3 or earlier
  - ✓Victim has a public IP address (i.e., not behind NAT, Tor, VPN, …)

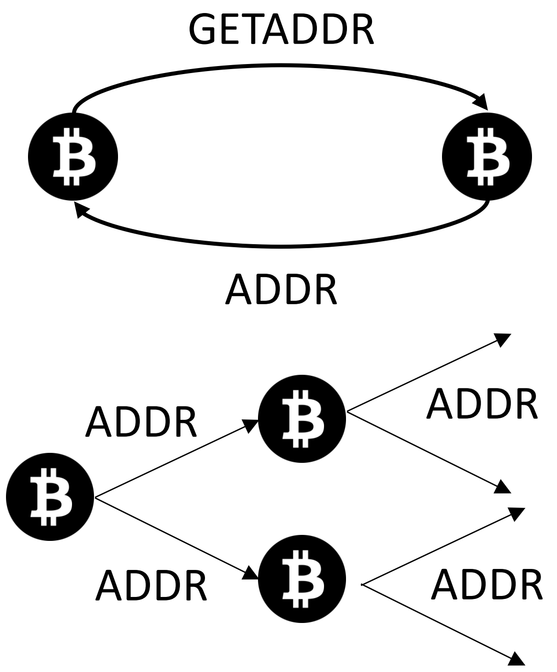
How Bitcoin nodes form the P2P network



- Bitcoin node establishes and maintains ***8 outgoing connections***
  - ✓Outgoing peers are selected from internal peer database (i.e., addrman)
- A node also accepts at most ***117 incoming connections***
  - ✓Only full nodes with public IP accept incoming connections
  - ✓Today network includes ~8K full nodes

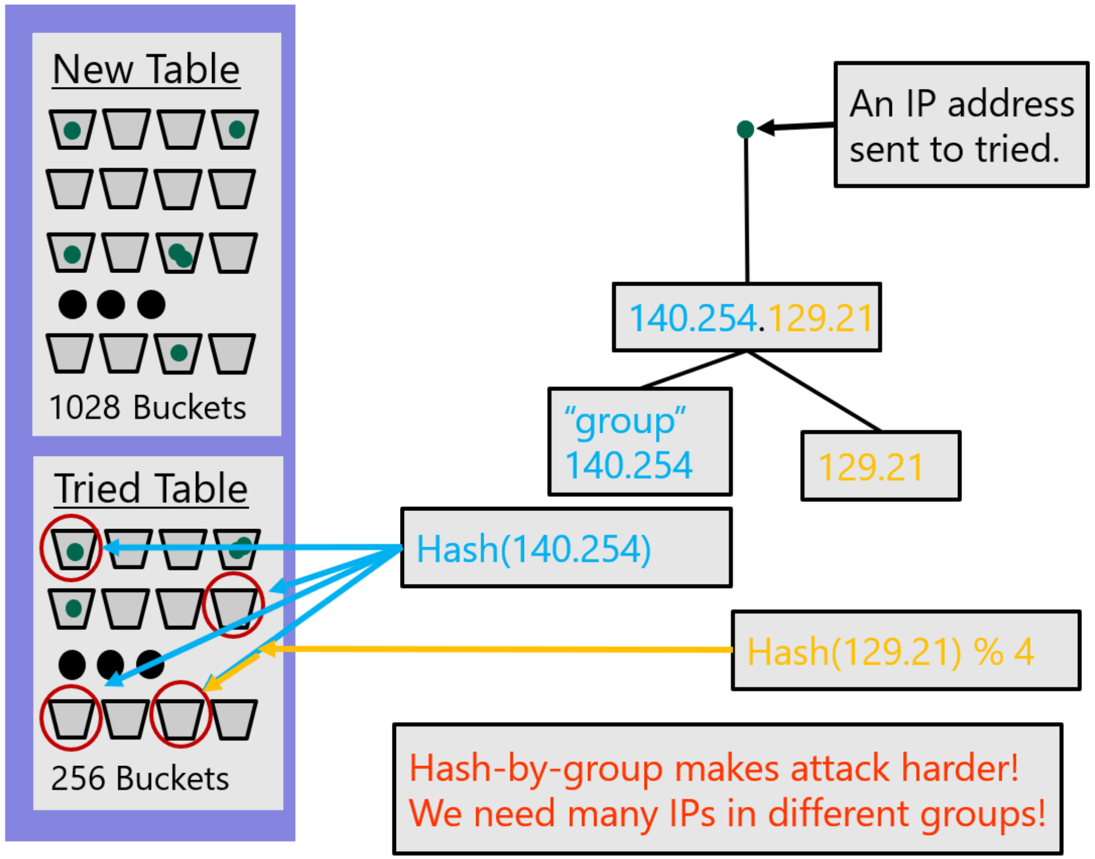
Address propagation

```
vSeeds.emplace_back("seed.bitcoin.sipa.be"); // Pieter Wuille
vSeeds.emplace_back("dnsseed.bluematt.me"); // Matt Corallo
vSeeds.emplace_back("dnsseed.bitcoin.dashjr.org"); // Luke Dashjr
vSeeds.emplace_back("seed.bitcoinstats.com"); // Christian Decker
vSeeds.emplace_back("seed.bitcoin.jonasschnelli.ch"); // Jonas Schnelli
vSeeds.emplace_back("seed.btc.petertodd.org"); // Peter Todd
vSeeds.emplace_back("seed.bitcoin.sprovoost.nl"); // Sjors Provoost
```



- Boot-strapping: New node gets a list of peers via DNS seeds or hard-coded IP addresses.
- Solicited **ADDR** messages: responses to a **GETADDR** request
- Unsolicited **ADDR** messages: relaying other's advertised IPs
- Each **ADDR** message contains up to 1,000 IP address.

Storing address in addrman



- addrman stores each IP with a timestamp:
  - ✓New table: IPs from ADDR messages
  - ✓Tried table: IPs of the peered nodes
- Adding an IP to Tried table:
  - ✓The /16 group determines 4 buckets
  - ✓The rest of the IP determines 1 out of 4 buckets
  - ✓If bucket is full, pick 4 random IPs and delete the oldest one (bitcoin eviction policy)

Attacking steps

- Fill the Tried table
  - ✓Simply connect to the victim node from botnet
- Fill the New table
  - ✓Flood unsolicited ADDR messages (contain 1,000 IPs each)
- Restart the victim node
- Make 117 incoming connections to the victim
- All selected outgoing peers are adversary IPs
  - ✓Outgoing peer is chosen from either New and Tried table
  - ✓Selected IP is biased toward “fresher” timestamps

Vulnerabilities exploited

- Vulnerability 1 (**Selection Bias**):
  - ✓ Attacker ensures its IPs are fresher, so they are more likely to be selected => Keep filling the New table to update the timestamp
- Vulnerability 2 (**Try-Try-Again**):
  - ✓ If an attacker IP replaces another attacker IP, she can resend the evicted IP and eventually replace an honest IP
- Vulnerability 3 (**Eviction Bias**):
  - ✓ Attacker ensures its IPs have recent timestamps to avoid being evicted
- IP diversity requirements:
  - ✓ Only 3,000 botnet IPs have sufficient /16 prefix diversity

Countermeasures to Eclipse attack

- **Random eviction:** No preference to newer IPs
- **Test before evict:** If existing IP is still reachable, do not evict it
- **Feeler connection:** Periodically test an IP in New table and move it to Tried if it is reachable
- **Larger table size:** Both tables are increased by 4 times
- **Remove direct IP inserting to Tried table:** Only IPs, that the node makes outgoing connection to, are stored.

- All above countermeasures are adopted in Bitcoin Core
  => Eclipse attack doesn’t work against the latest version

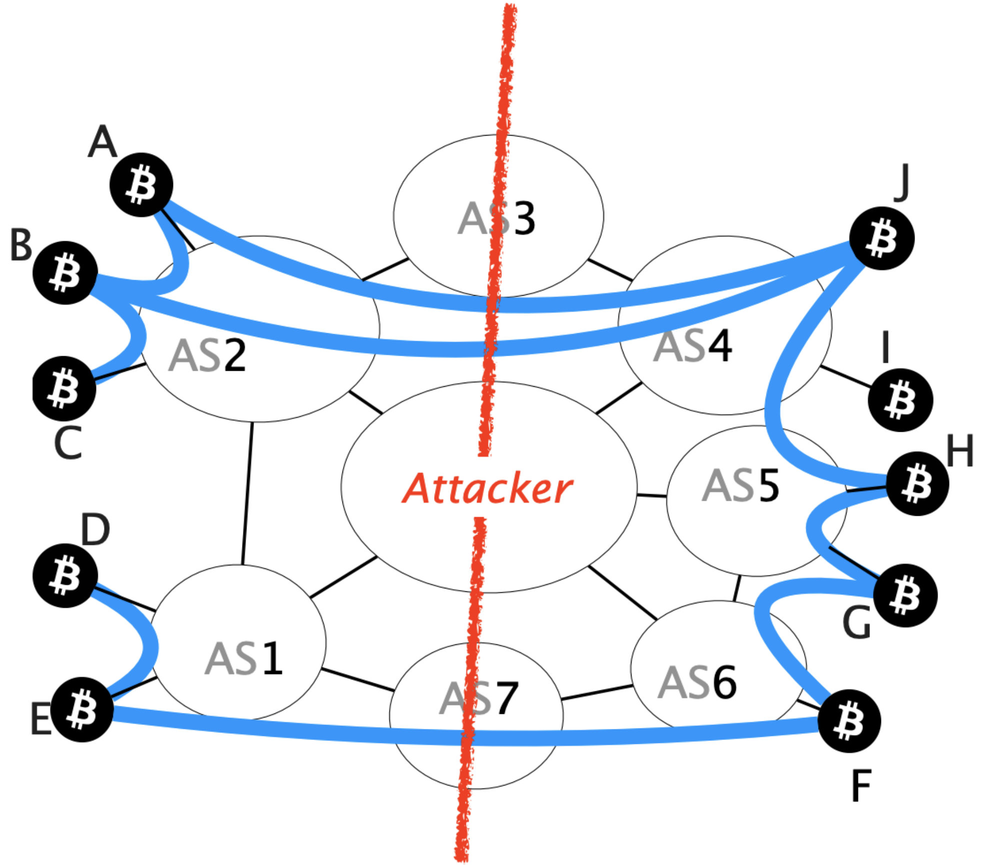
Bitcoin hijacking

- Paper "*Hijacking Bitcoin: Routing Attacks on Cryptocurrencies*" by Apostolaki et al. [IEEE S&P 2017]

- Threat model:
  - ✓Attacker's capability: Adversary is a network attacker (i.e., an ISP)
  - ✓Attacker's goals: intercepts all Bitcoin connections of the victim

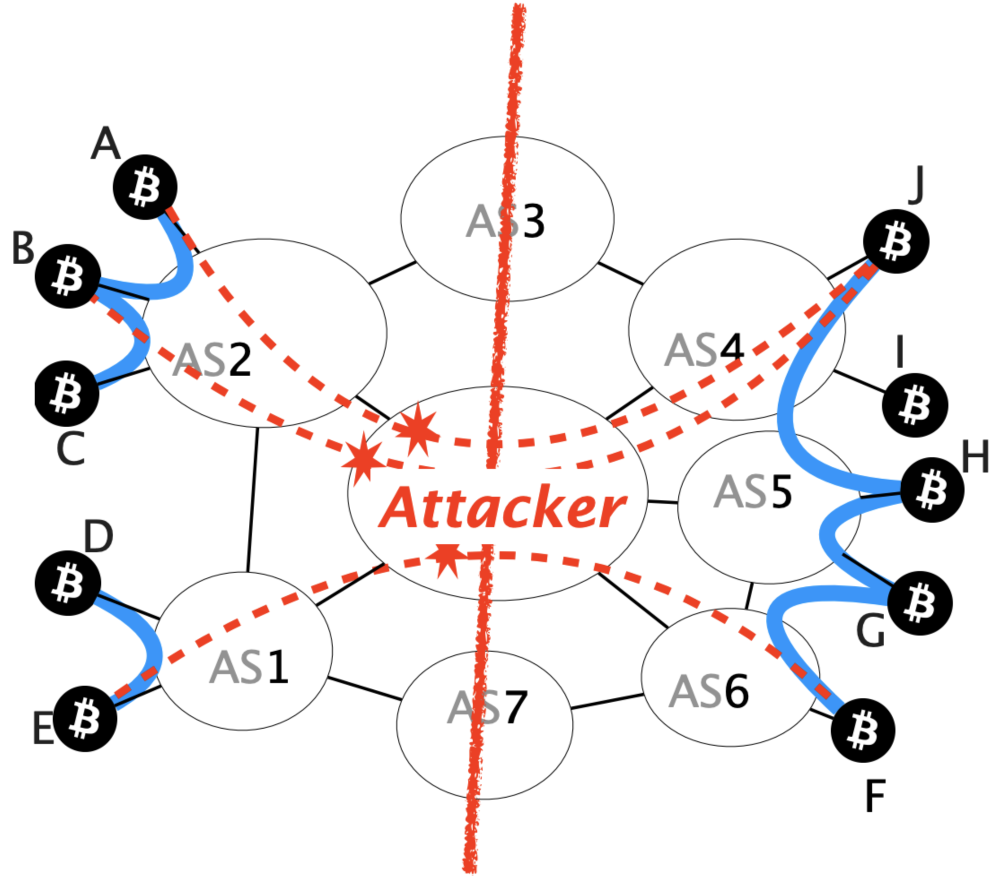
- Assumptions:
  - ✓Attacker can launch BGP hijacking attack (not all ASes can do!)
  - ✓Victim's IP belongs to a "hijack-able" prefix (i.e., prefix shorter than /24)

Attack steps



- Nodes of the left and the right side of the network communicate via Bitcoin connections.
- The attacker wishes to split the network into two disjoint components
- The attacker intercepts the traffic destined to the right nodes by performing BGP hijacks

Attack steps (cont.)



- After the hijack, all traffic sent from the left to the right side is forwarded through the attacker.
- The attacker can drop all Bitcoin traffic => partitions are created
- Some connections cannot be intercepted
  - ✓Nodes within same AS
  - ✓Miners in the same mining pool

Attack effectiveness and practicality

- Victim: **93%** Bitcoin nodes' IPs belong to prefixes shorter than /24
- Attacker: Hijack **<100 prefixes** can isolate up to 47% mining power
- ASes (e.g., large ISPs) **can** launch this Bitcoin hijacking attack
  - Question: Do they really launch this attack in practice?
  - **Yes,** but attack is quickly **detected!**

cisco BGPMON

HOME BLOG ABOUT US PRODUCTS AND SERVICES CLIENT PORTAL

The Canadian Bitcoin Hijack

Posted by Andree Toonk - August 12, 2014 - Hijack - No Comments

A few days ago researchers at Dell SecureWorks published the details of an attacker repeatedly hijacking BGP prefixes for numerous large providers such as *Amazon, OVH, Digital Ocean, LeaseWeb, Alibaba* and more. The goal of the operation was to intercept data between Bitcoin miners and Bitcoin mining pools. They estimated that $83,000 was made with this attack in just four months. The original post has many of details which we won't repeat here, instead will take a closer look at the BGP details of this specific attack. **Attack details** Our friends at Dell SecureWorks decided not to name the network from which the hijacks originated. As a result we won't name the exact Autonomous System either, instead we will suffice by saying that the originator of this hijack is a network operating in Eastern Canada. **Initial experiment** BGPmon

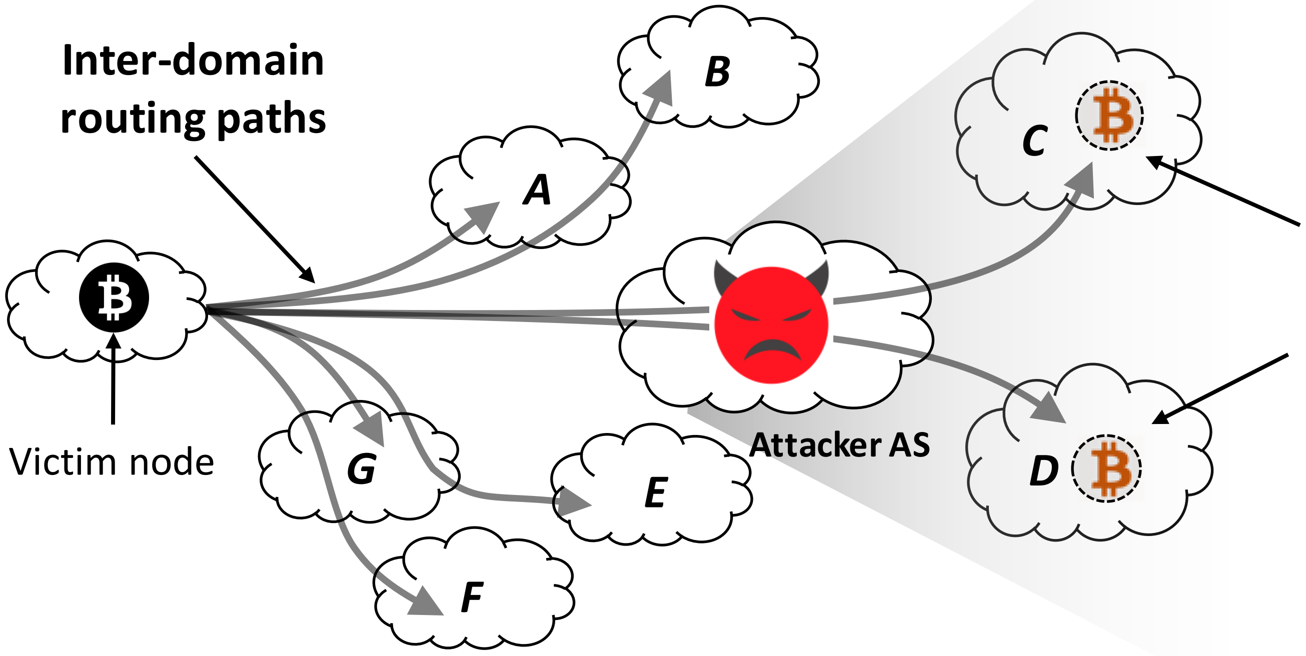
Erebus attack

- Paper "*A Stealthier Partitioning Attack against Bitcoin Peer-to-Peer Network*" by Tran et al. [IEEE S&P 2020]

- Threat model:
  - ✓Attacker's capability: Adversary is a network attacker (i.e., an ISP)
  - ✓Attacker's goals: Influence all of victim's connections to be made to adversary-controlled bots

- Assumptions:
  - ✓Victim has a public IP address (i.e., not behind NAT, Tor, …)
  - ✓Victim runs the latest Bitcoin ver. with Eclipse countermeasures

Idea: Using "***shadow***" IPs as attack resource

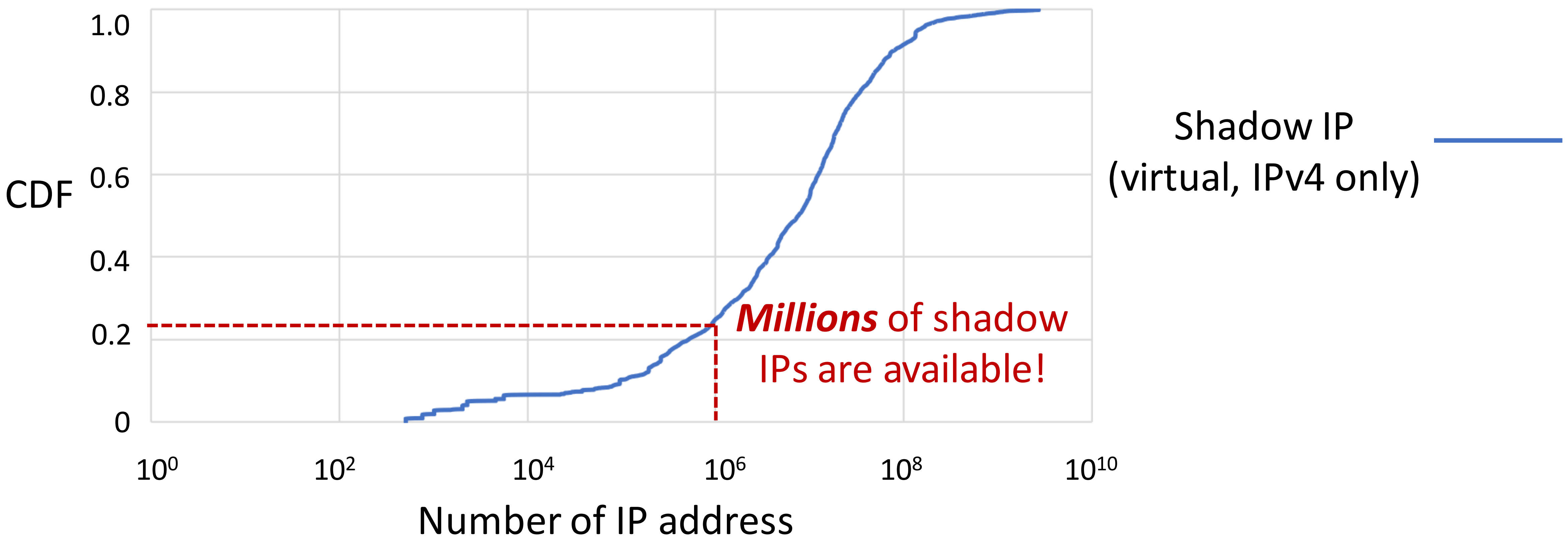


Observation: Traffic from victim node to ***any*** IP addresses at AS ***C*** and ***D*** would ***traverse attacker AS***

- ***Shadow IPs*** are valid IPs whose victim-to-IP routes include the attacker AS
- Attacker AS can ***spoof*** connections with victim node ***on behalf*** of shadow IPs

  => Attacker ***virtually*** controls shadow IPs

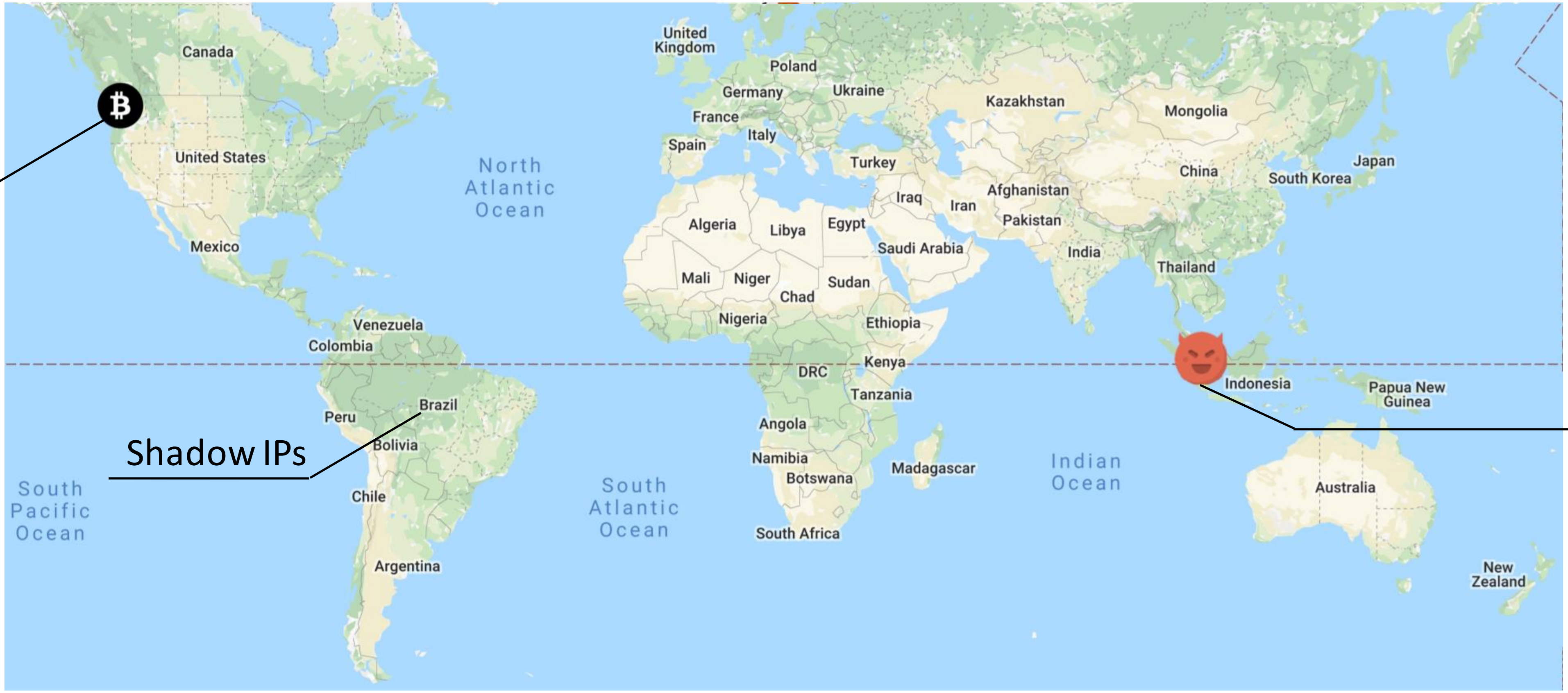
How many shadow IP addresses are available?



When attackers are ***top-100 ASes*** and victim are hosted at 100 random ASes

Shadow IPs are ***geographically well-distributed***

Victim node (e.g., Amazon)



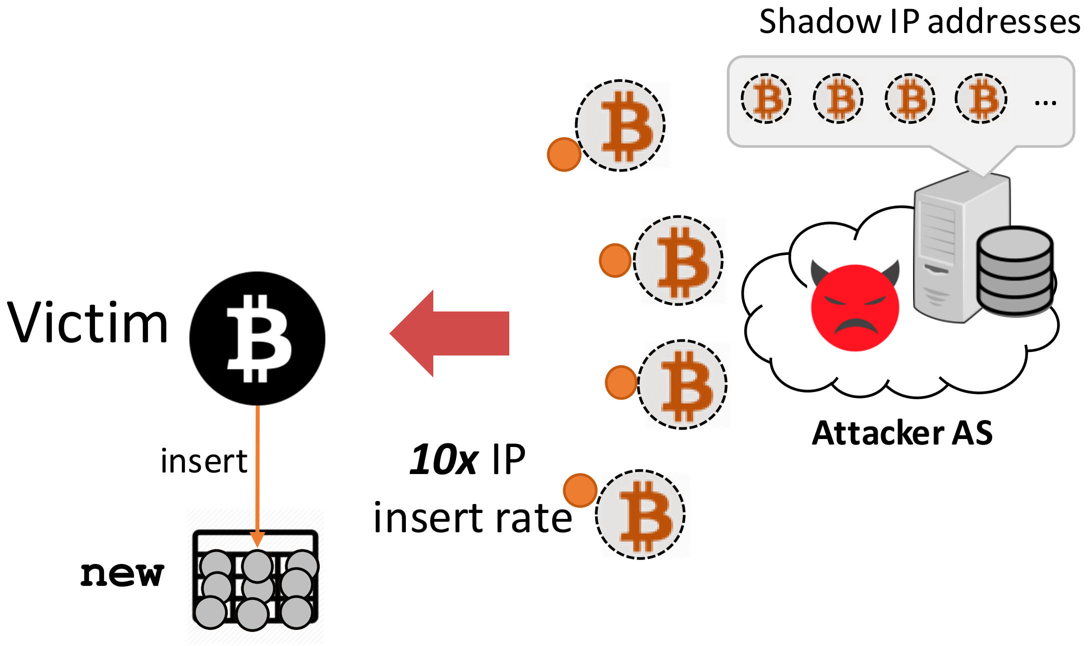
Shadow IPs

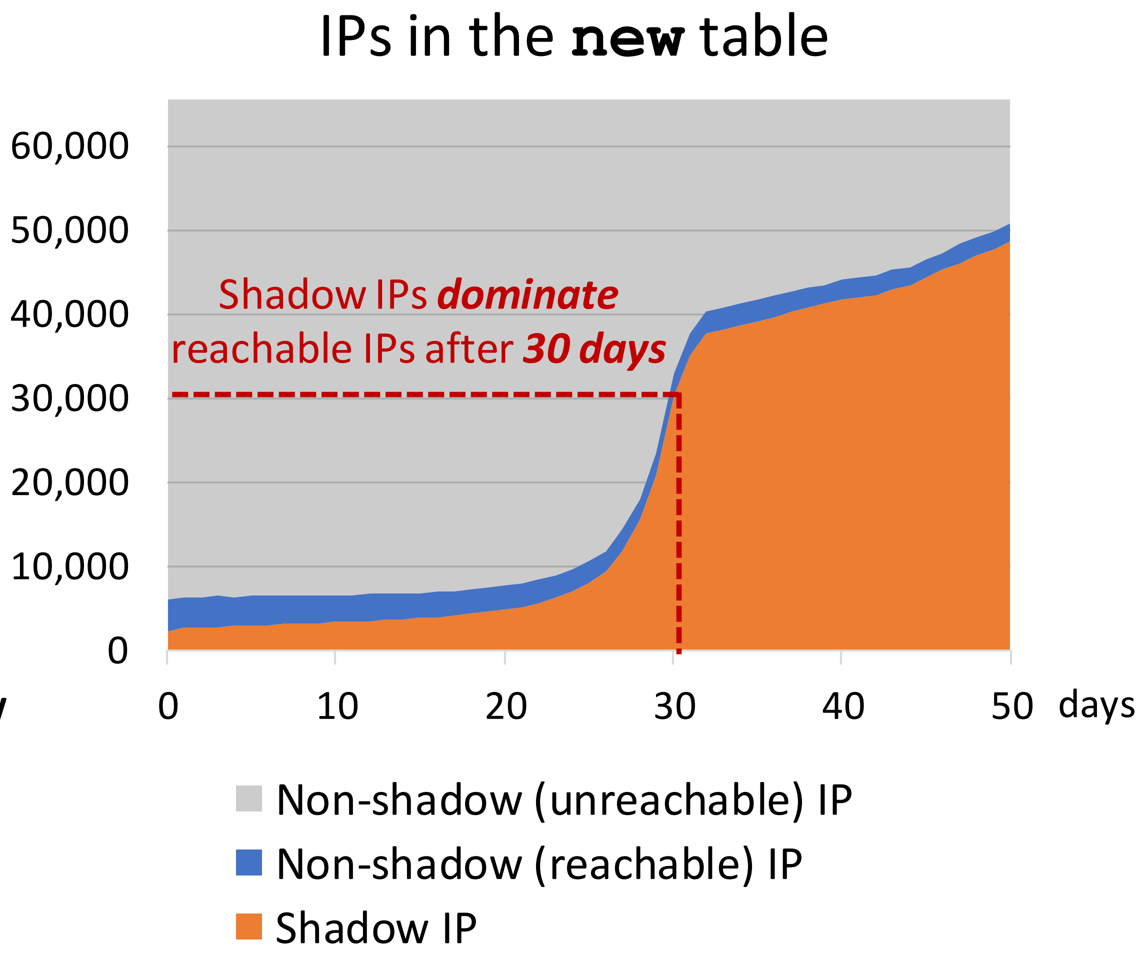
Attacker AS in Singapore

Shadow IPs are usually ***well-distributed*** across the world

=> look normal to cautious Bitcoin nodes

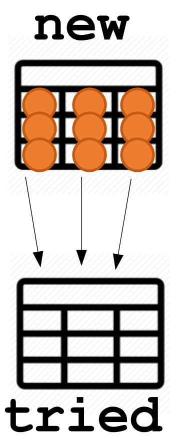
Dominate ***new*** table: advertise shadow IPs with ***high rate***





- Attacker fills the **new** table by sending shadow IPs with ***high rate*** to replace legitimate IPs
  - ✓e.g., an IP is deleted if it is not heard from peers in ***30 days***

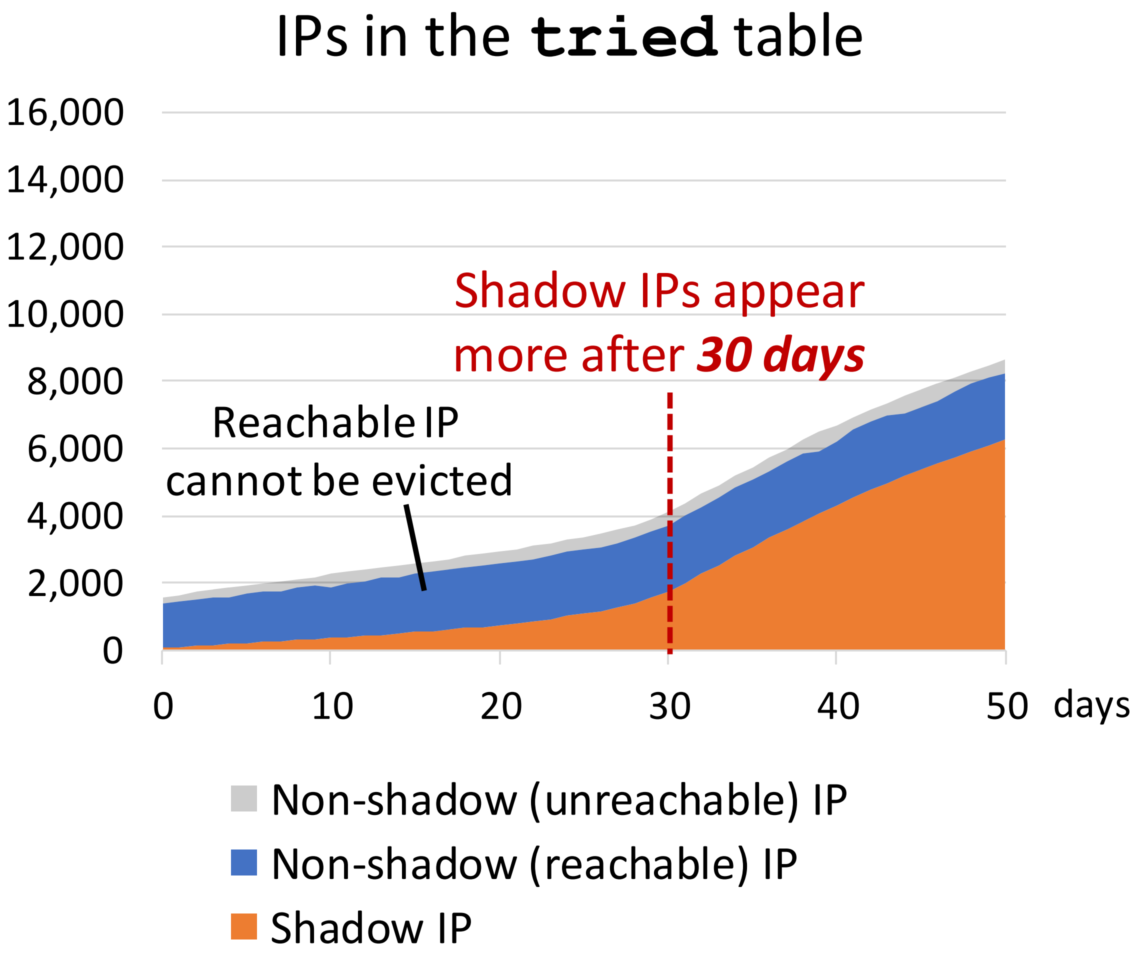
Dominate *tried* table: *patiently* wait for IP *trickle-downs*



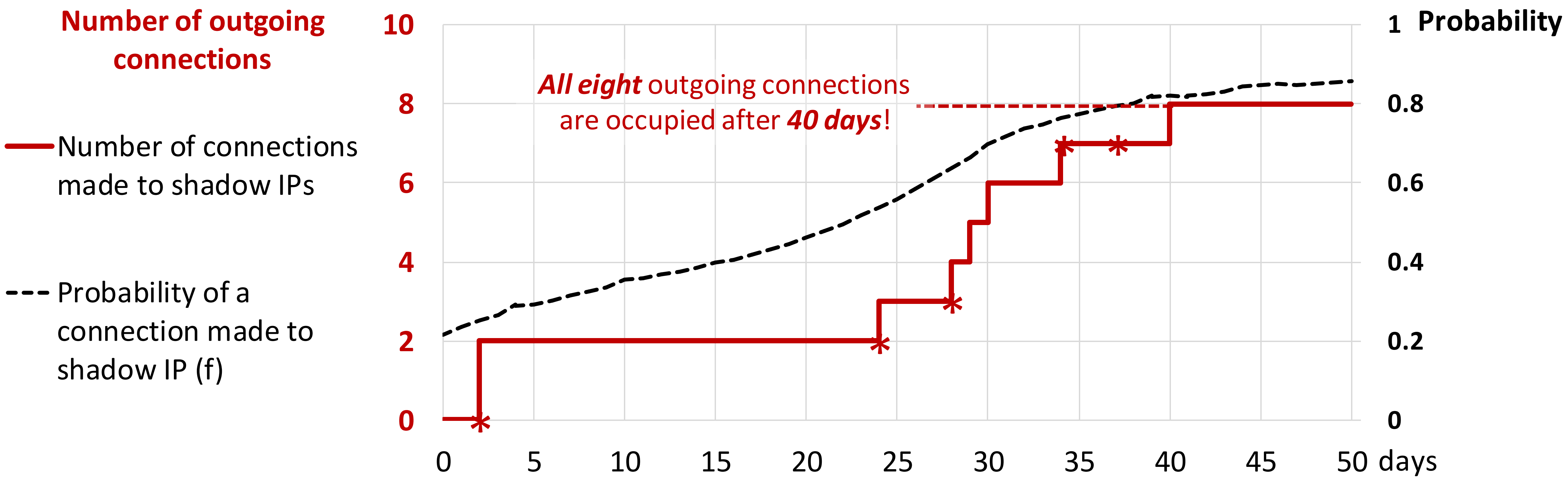
Trickle-down

Rate:
1 IP / 2 mins

- Attacker fills the **`tried`** table by ***patiently*** waiting for ***trickle-downs***
  - ✓e.g., one IP is moved from **`new`** to **`tried`** every ***two minutes*** via feeler connections



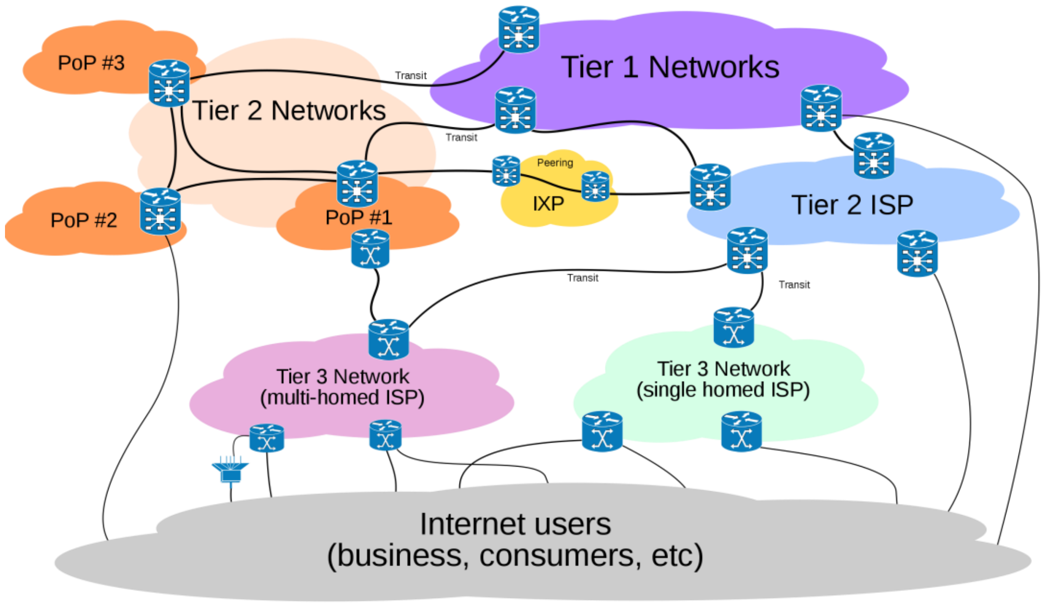
The Erebus attacker patiently isolates victim Bitcoin nodes in ***5 - 6 weeks***



- Attacker keeps sending ***low rate*** attack traffic (***520 bit/s*** or ***2 IP/s)*** until it controls ***all eight*** outgoing connections of the victim

Who can launch the Erebus attack?

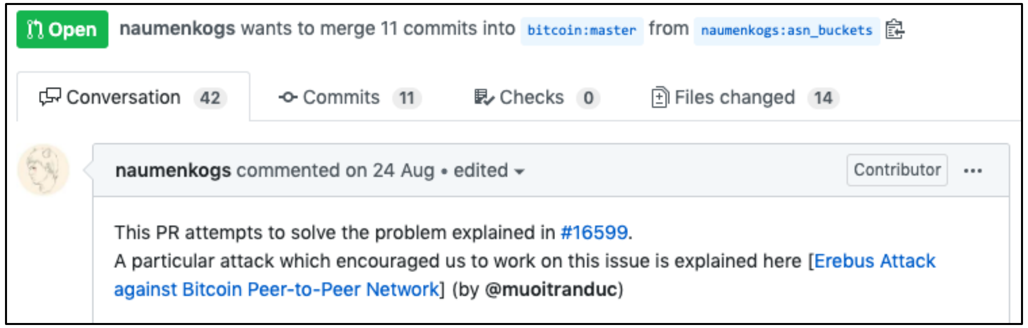
- To attack a targeted node, Erebus attacker needs:
  - ✓ ***millions*** shadow IP addresses
  - ✓ ***several weeks*** of attack execution
- ***All Tier-1*** networks
  - ✓ AT&T, CenturyLink, NTT, …
  - ✓ Can target ***any*** Bitcoin node!
- Many ***large Tier-2*** networks
  - ✓ Singtel, China Telecom, …
  - ✓ Can target most Bitcoin nodes!
- ***Nation-state*** adversaries
  - ✓ Some countries are believed to have direct control over their ISPs



The Erebus attack has a ***high impact***

EREBUS

- The Erebus attack works against ***latest*** Bitcoin core v0.18.1
  - ✓ ***Any public Bitcoin node*** can be targeted!
  - ✓ Complete mitigation is ***difficult***: ***No*** bugs was exploited but only Internet routing
  - ✓ Deterrence are ***being implemented*** by Bitcoin
- ***34 out of top-100*** cryptocurrencies are also potentially vulnerable to the Erebus attack
- Project webpage: https://erebus-attack.comp.nus.edu.sg/



…

Comparing three attacks

	Bitcoin hijacking attack	Erebus attack	Eclipse attack
Attack capabilities			
Network attacker?	Yes	Yes	No
Distributed computing resource required?	No	No	Yes
Route manipulation required?	Yes	No	No
Attack outcomes			
Effective against Bitcoin v0.18.0?	Yes	Yes	No
Attacks are visible and attributable?	Yes	No	No

Summary

- ***Blockchain*** is the biggest innovation in cryptocurrencies
- Blockchains rely on the ***security*** guarantees of the underlying ***networks***
- ***Partitioning attacks*** are great examples of how a network adversary can attack consensus and/or transactions layers
- ***Network security*** of blockchain is a ***promising*** research direction