SybilLimit: Secure Decentralized Reputation System

Petar Maymounkov
petar@protocol.ai

SybilLimit: Secure Decentralized Reputation System

Petar Maymounkov
petar@protocol.ai

- Based on SybilLimit (2010, Yu, et al.)
- Some simplifications, small improvements

Introduction

Context

In search of **robust** decentralized architectures:
– Open membership
– Surviving (malfunction, Sybil attacks, **Sybil infections**)
– Scalable
– Live (responsive)

For broadcast (gossipsub), DHT (Kademlia) and in general.

Summary

+ Technical ideas are strong.

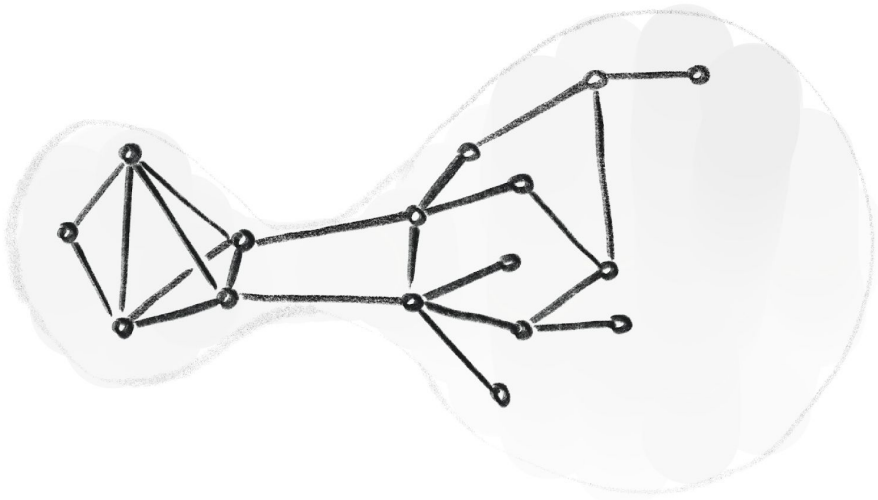
– Solution has a non-intuitive “user interface”.

– Solution not strong for consensus, which handles <25% infection.

+ I was able to reuse techniques to design a new provably robust decentralized system for broadcast and consensus.

Reputation system

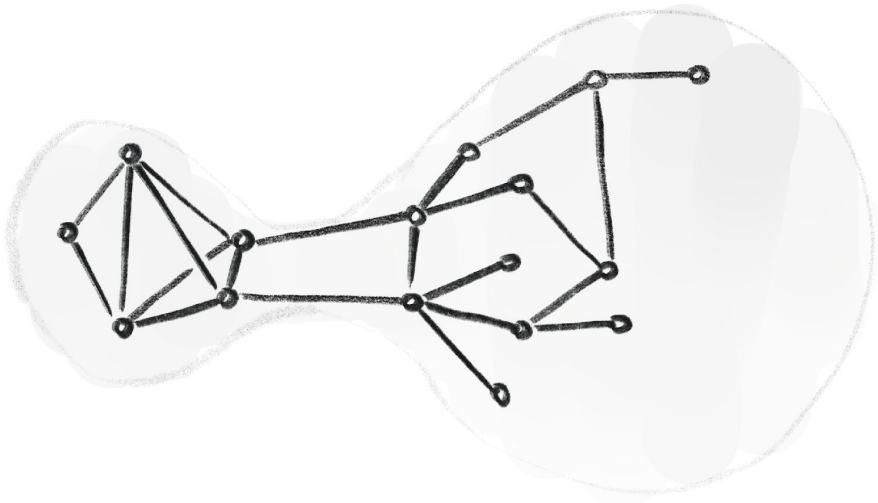
Motivated by the belief that “communities” are well-connected,
with sparse bottlenecks between “communities”.



Reputation system

What is a **“reputation system”**?

- Add/remove edge/vertex
- Query: How well **“connected”** are two nodes?

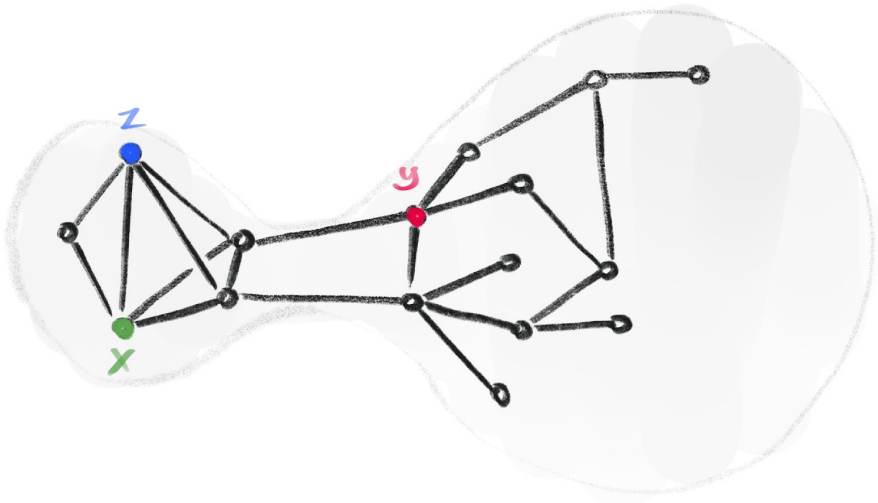


Reputation system

What is a **“reputation system”**?

- Add/remove edge/vertex
- Query: How well **“connected”** are two nodes?

$$\text{conn}(x, y) = \min_{\text{cut}} \frac{\text{edges across cut}}{\text{edges on side of x}}$$



Reputation system

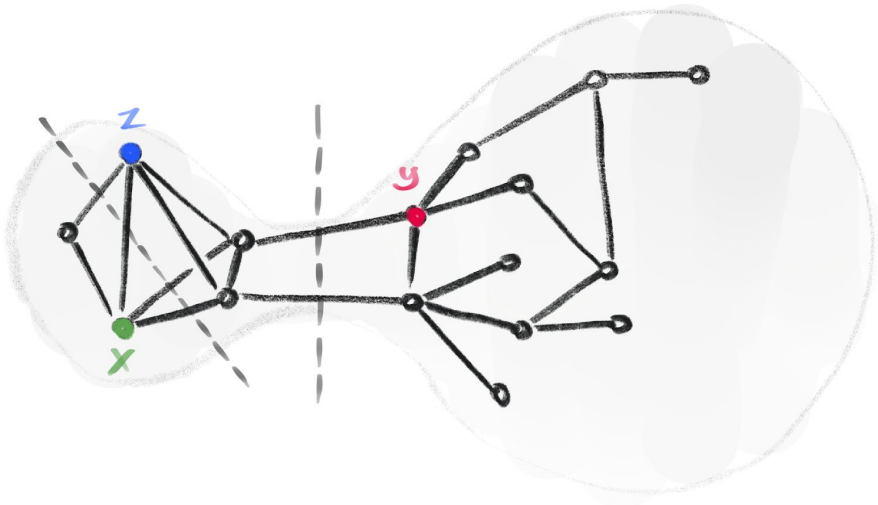
What is a **“reputation system”**?

- Add/remove edge/vertex
- Query: How well **“connected”** are two nodes?

$$\text{conn}(x, y) = \min_{\text{cut}} \frac{\text{edges across cut}}{\text{edges on side of x}}$$

$$\text{conn}(x, y) = \frac{2}{10} = 0.2$$

$$\text{conn}(x, z) = \frac{4}{5} = 0.8$$



Reputation system

What is a **“reputation system”**?

- Add/remove edge/vertex
- Query: How well **“connected”** are two nodes?

Diffusion = Distribution of the last vertex of a random walk.

Diffusions from well connected vertices **intersect** more than diffusions from badly connected ones.

Reputation system

What is a **“reputation system”**?

- Add/remove edge/vertex
- Query: How well **“connected”** are two nodes?

Diffusion = Distribution of the last vertex of a random walk.

Diffusions from well connected vertices **intersect** more than diffusions from badly connected ones.

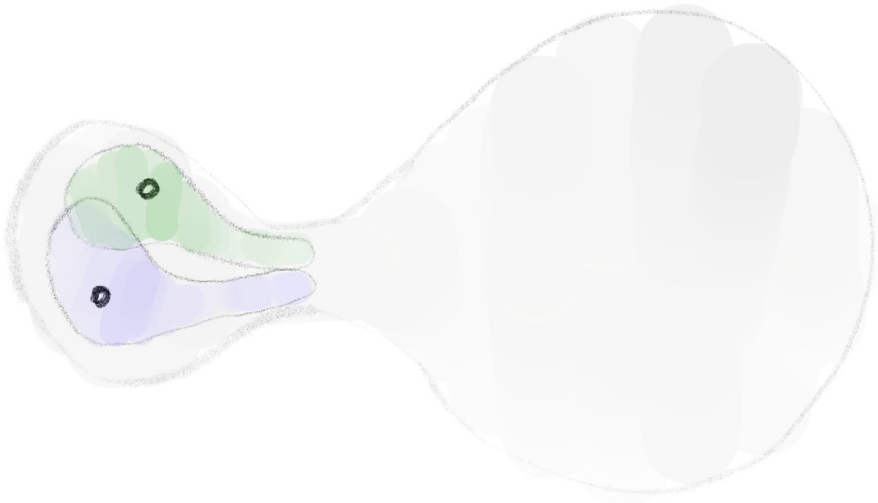
Reputation system

What is a **“reputation system”**?

- Add/remove edge/vertex
- Query: How well **“connected”** are two nodes?

Diffusion = Distribution of the last vertex of a random walk.

Diffusions from well connected vertices **intersect** more than diffusions from badly connected ones.



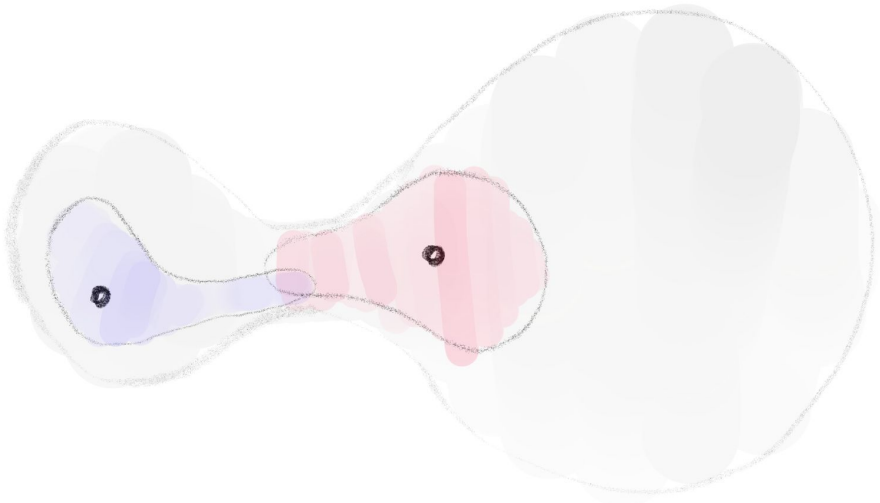
Reputation system

What is a **“reputation system”**?

- Add/remove edge/vertex
- Query: How well **“connected”** are two nodes?

Diffusion = Distribution of the last vertex of a random walk.

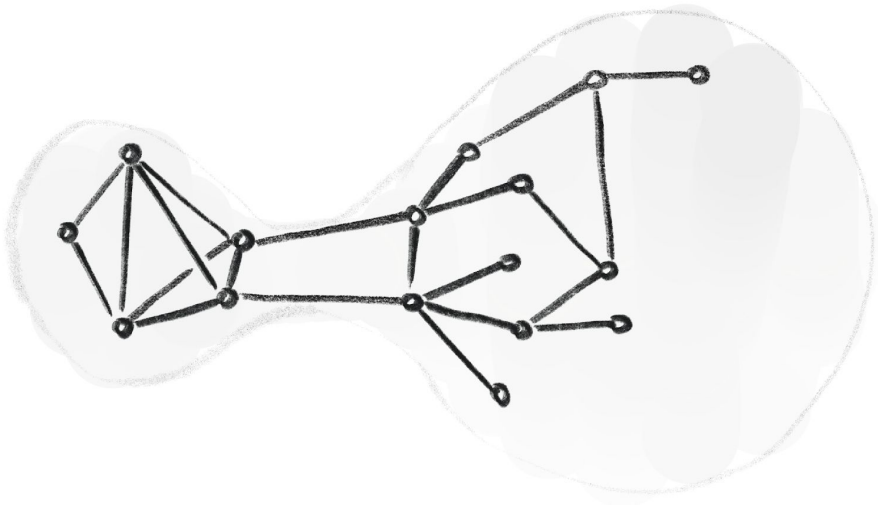
Diffusions from well connected vertices **intersect** more than diffusions from badly connected ones.



Decentralized reputation system

What is a **“decentralized reputation system”**?

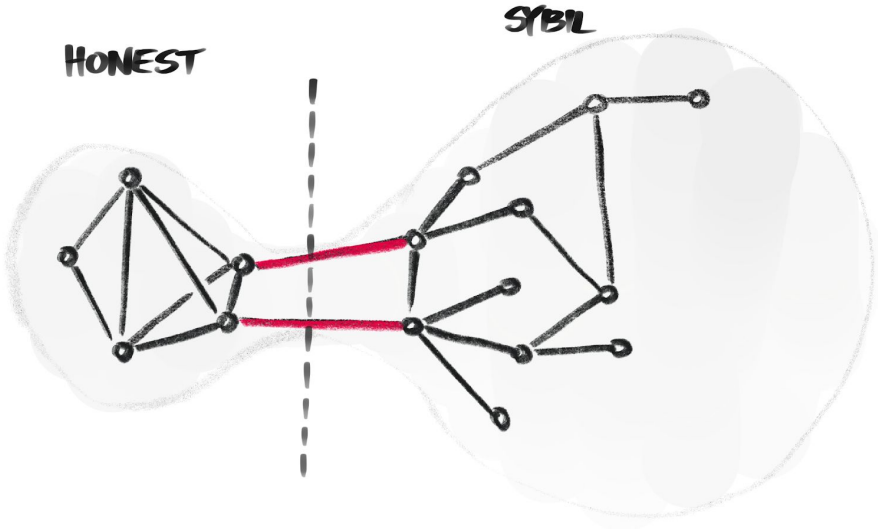
- Vertices are participants (compute nodes)
- Participants can add/remove **only incident** edges
- Query: Any two participant can compute their connectedness



Secure decentralized reputation system

What is a **“secure decentralized reputation system”**?

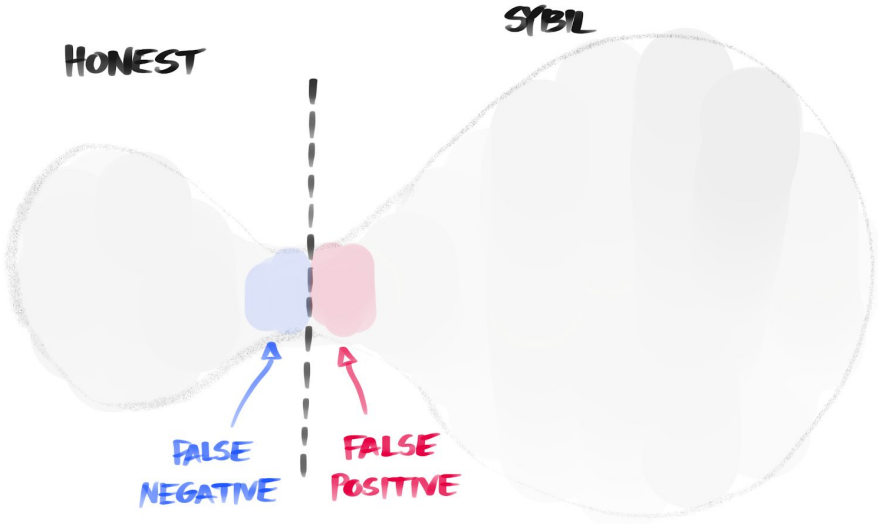
- Honest participants + unlimited amount of Sybil participants
- Honest “verifier” can check whether a “suspect” is well-connected
  - Query: *Is **my** connectedness to **you** below a threshold (e.g. %5)? Yes/no?*



Secure decentralized reputation system

What is a **“secure decentralized reputation system”**?

- Honest participants + unlimited amount of Sybil participants
- Honest “verifier” can check whether a “suspect” is well-connected
  - Query: *Is **my** connectedness to **you** below a threshold (e.g. %5)? Yes/no?*
- Size of the bottleneck affects the correctness of the answer



Experimental results

Vague sense of numbers:

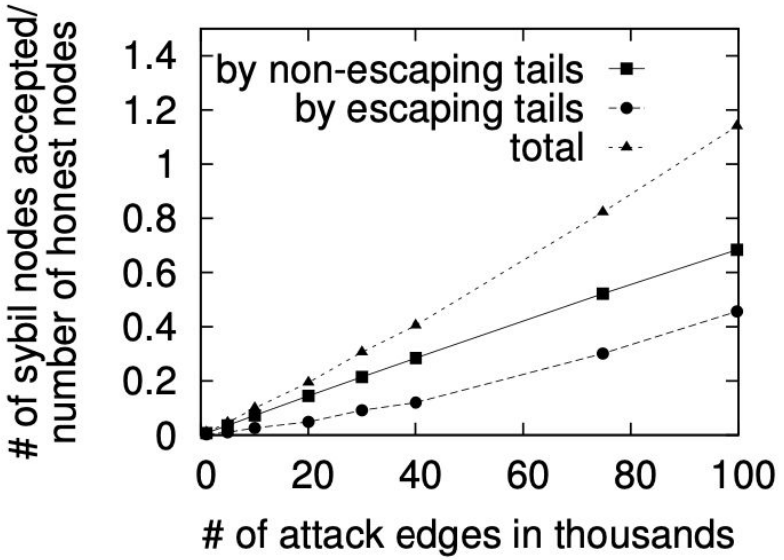
attack edges = 7% of honest nodes
infection = accepted sybils / accepted honest = 80%

} Consensus breaks after 25% infection

Caveats:

Topology-dependent. Using random walk diffusion (PageRank walk is better).

Data set	Friendster
Data set source	[33]
Date crawled	Nov-Dec 2005
# nodes	932,512
# undirected edges	7,835,974
w used in SybilLimit	10
r used in SybilLimit	8,000



Applications?

Applications vary based on what you model as a vertex/edge.

Social / subjective:

- Social network: edge = subjective trust (based on manual user input)

Socio-economic / subjective:

- Club with membership fees: edge = monetary transaction

User interface not intuitive?

Historical / objective:

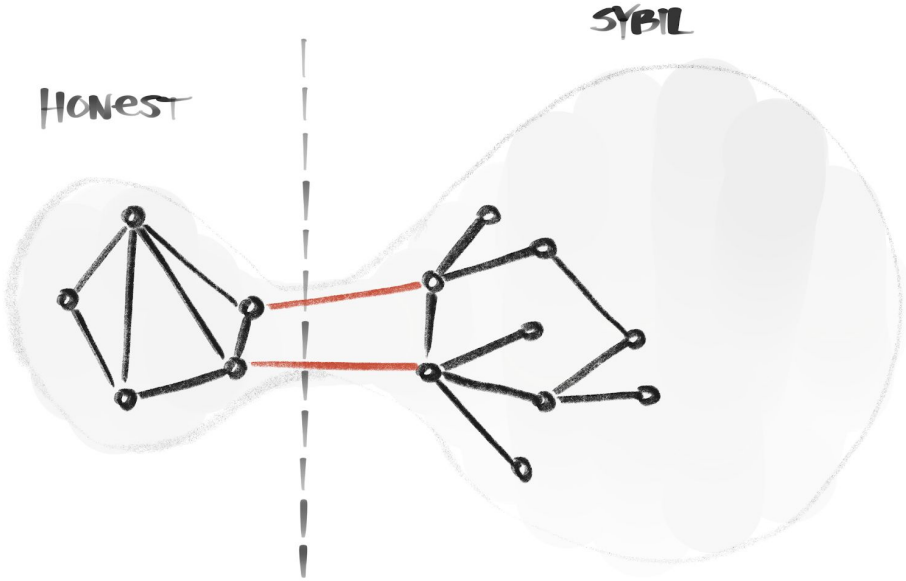
- Lineage relationships: “Alice joined the network via Bob and Chandra”

Introspective / objective:

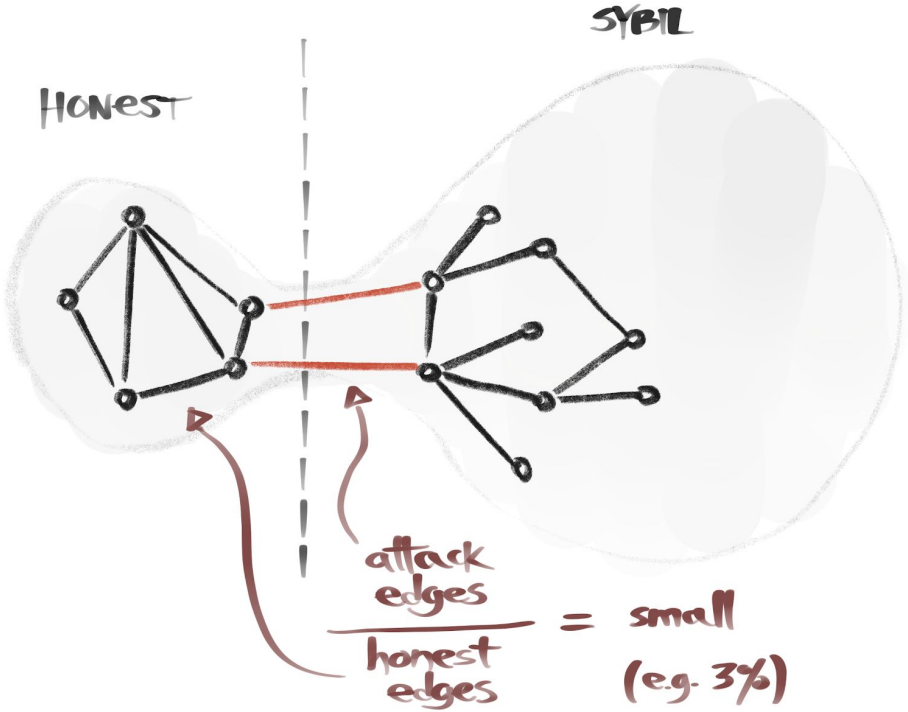
- App-related history: “Alice performed good work for Bob in past epoch.”

High-level plan

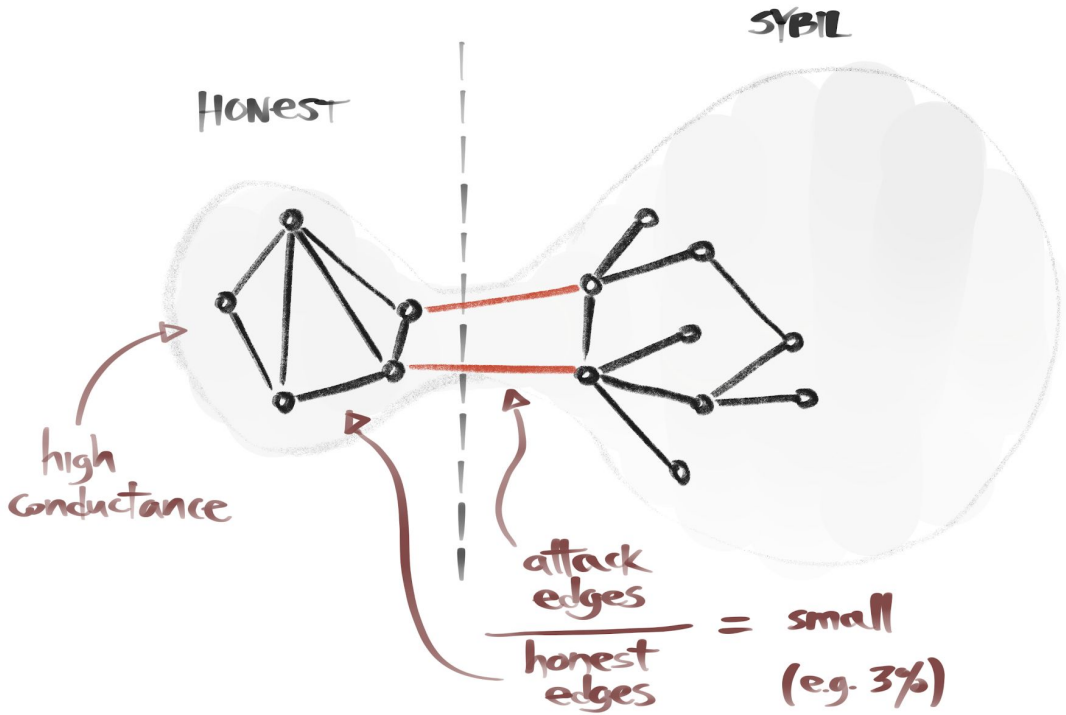
Model and assumptions



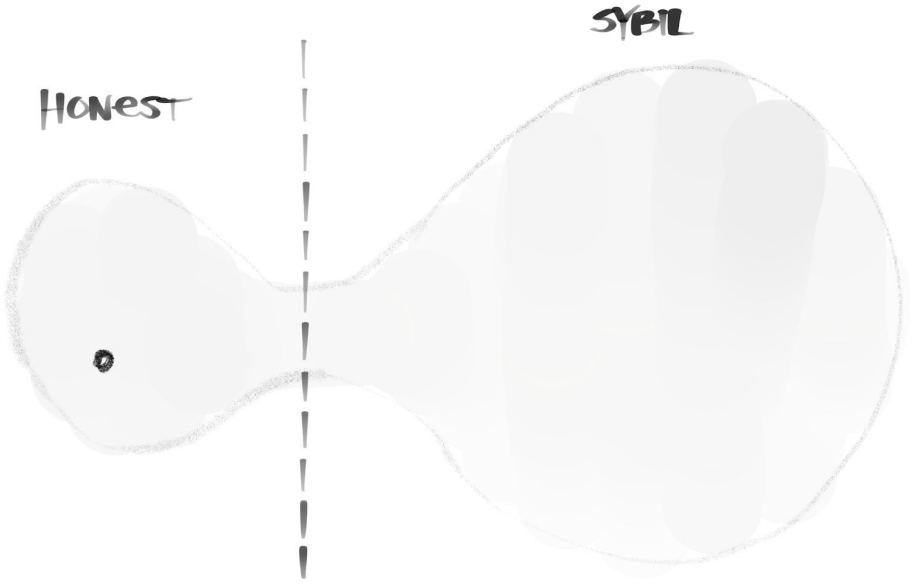
Model and assumptions



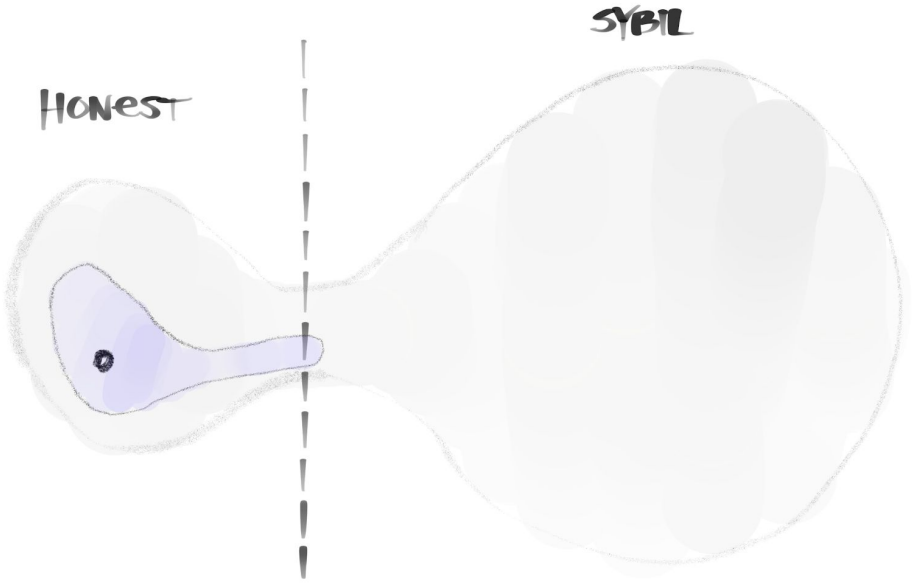
Model and assumptions



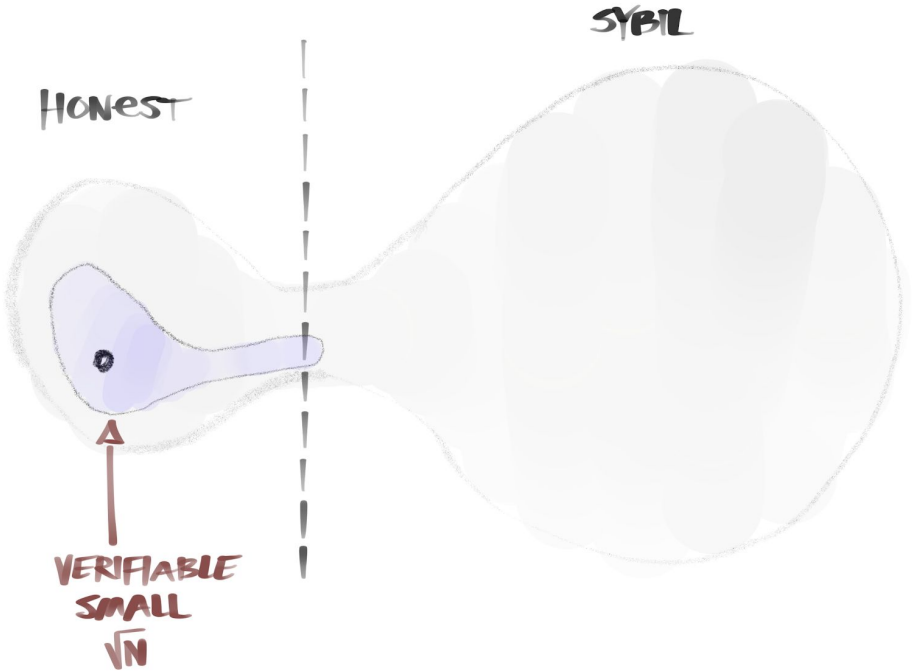
Proof idea: spectral view



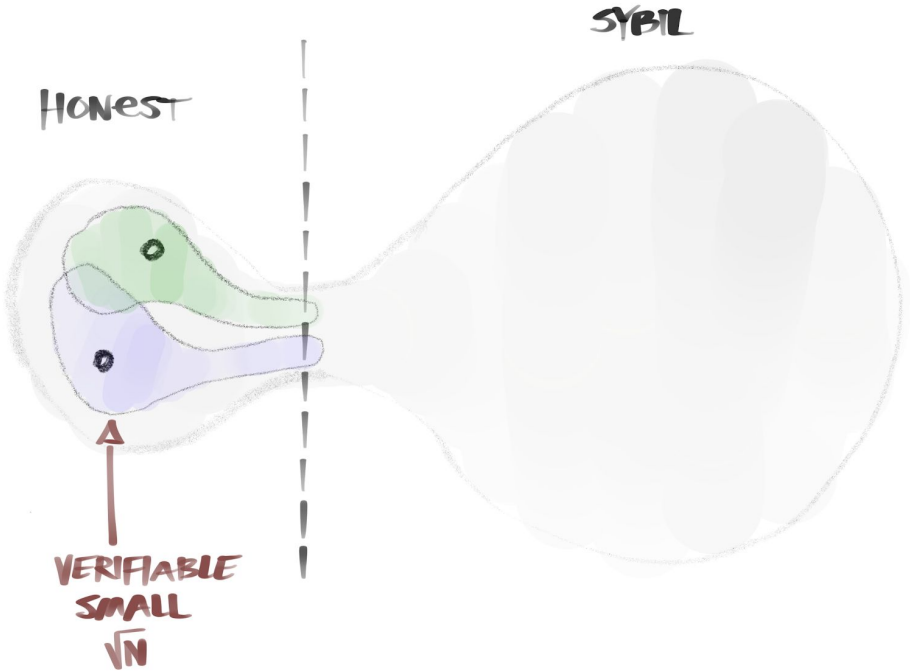
Proof idea: spectral view



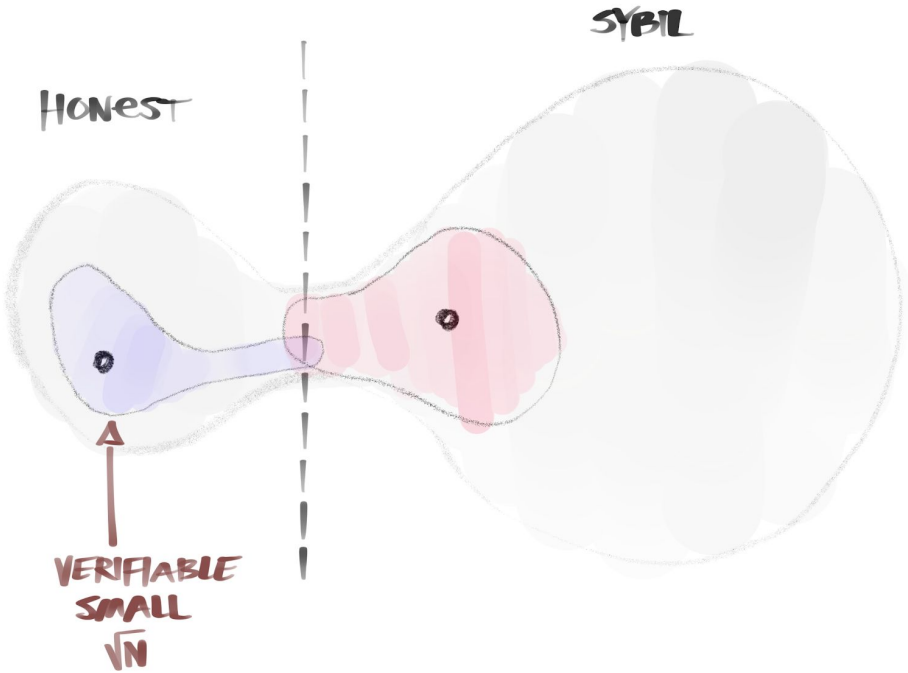
Proof idea: spectral view



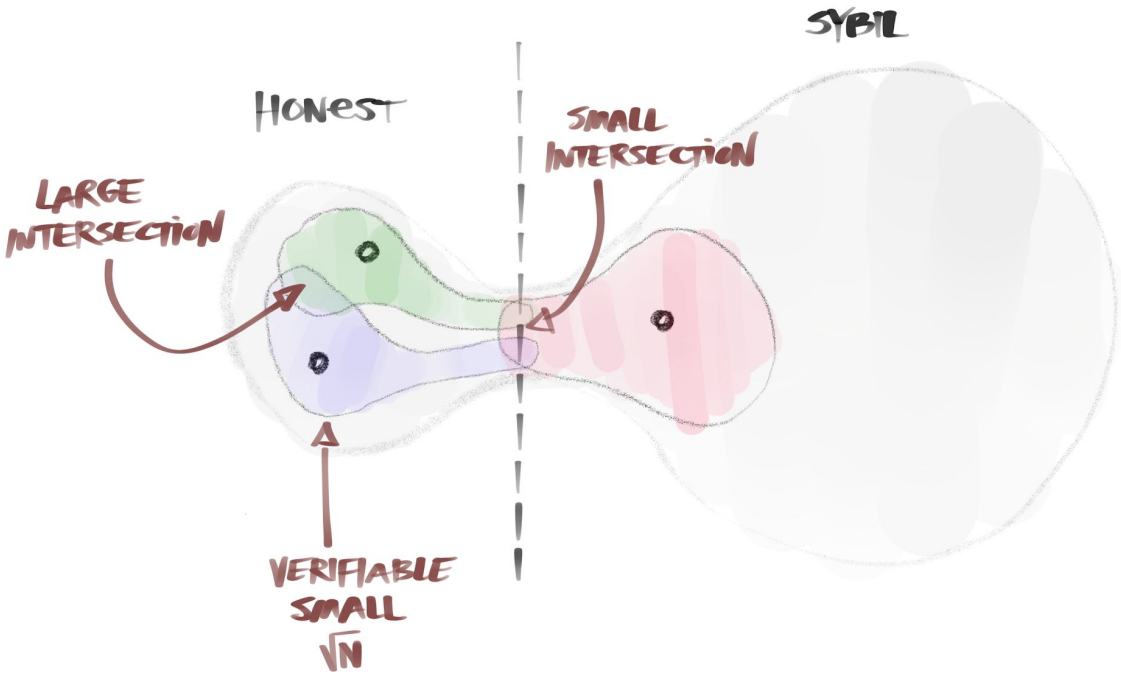
Proof idea: spectral view



Proof idea: spectral view



Proof idea: spectral view

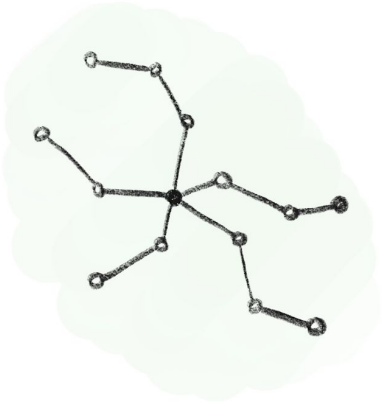


Protocol and algorithms

Verification algorithm

$\sqrt{N}$ WALKS

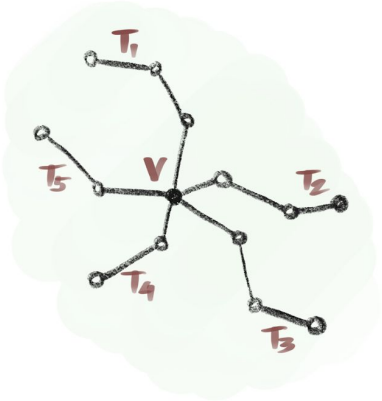
$\log N$ LENGTH



Verification algorithm

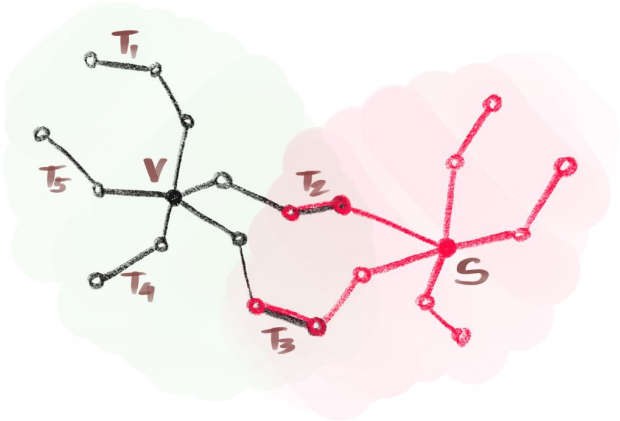
$\sqrt{N}$ WALKS

LogN LENGTH



Verification algorithm

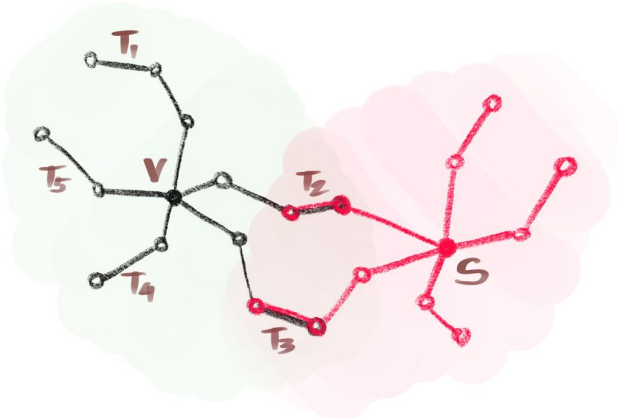
√N WALKS
LogN LENGTH



INTERSECTION
CONDITION

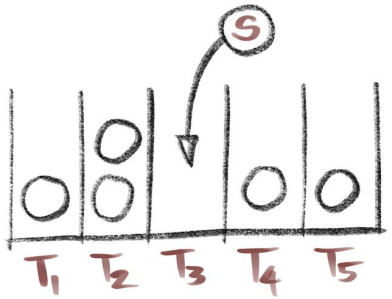
Verification algorithm

$\sqrt{N}$ WALKS
LOGN LENGTH



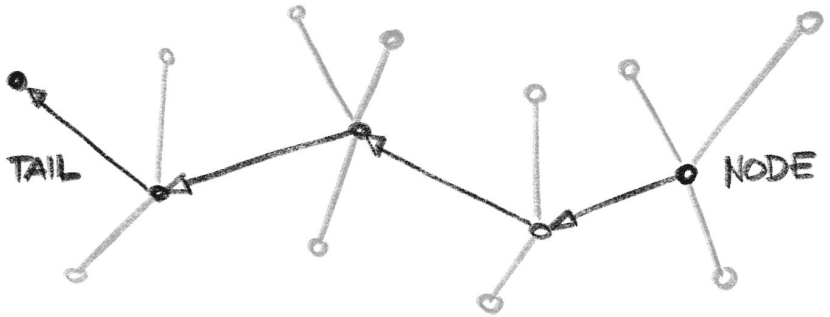
INTERSECTION CONDITION

$$\max \text{ counter} - \min \text{ counter} \leq 2$$

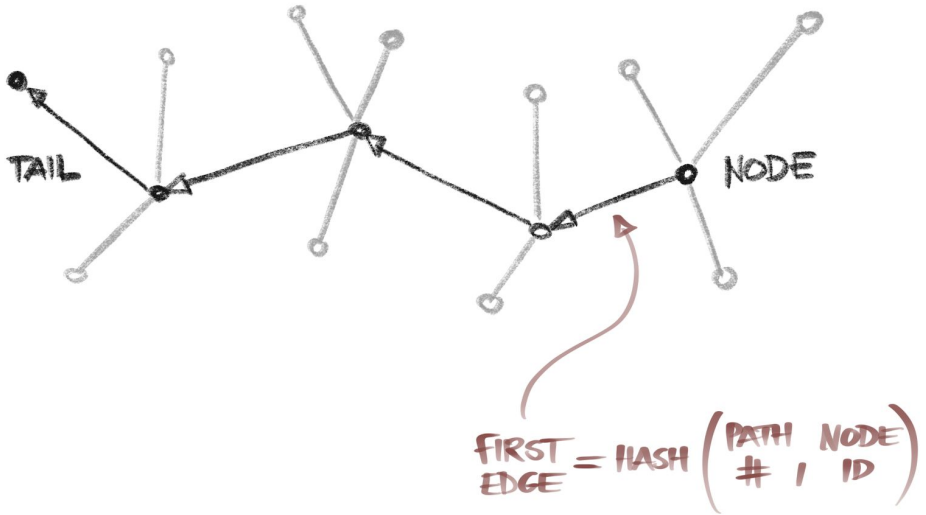


BALANCE CONDITION

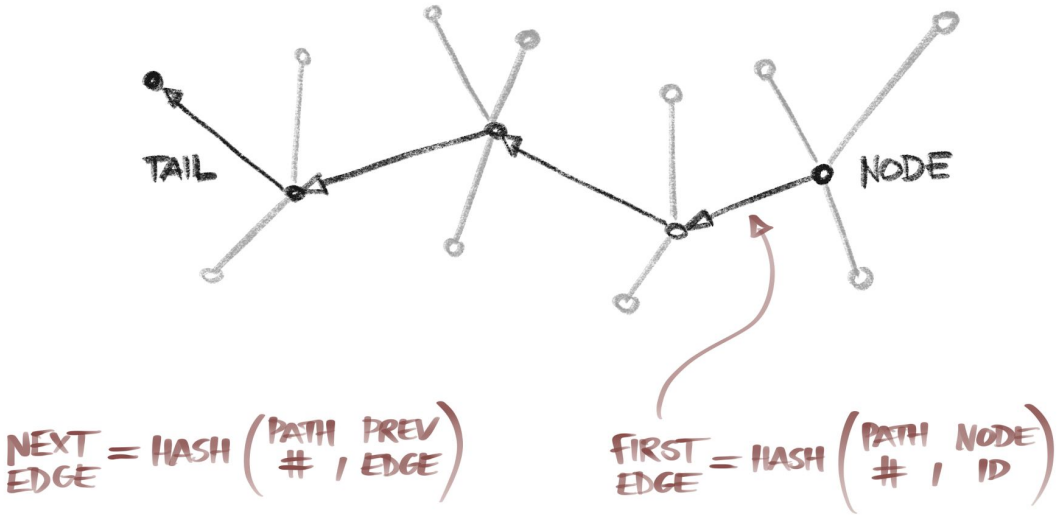
Verifiable paths



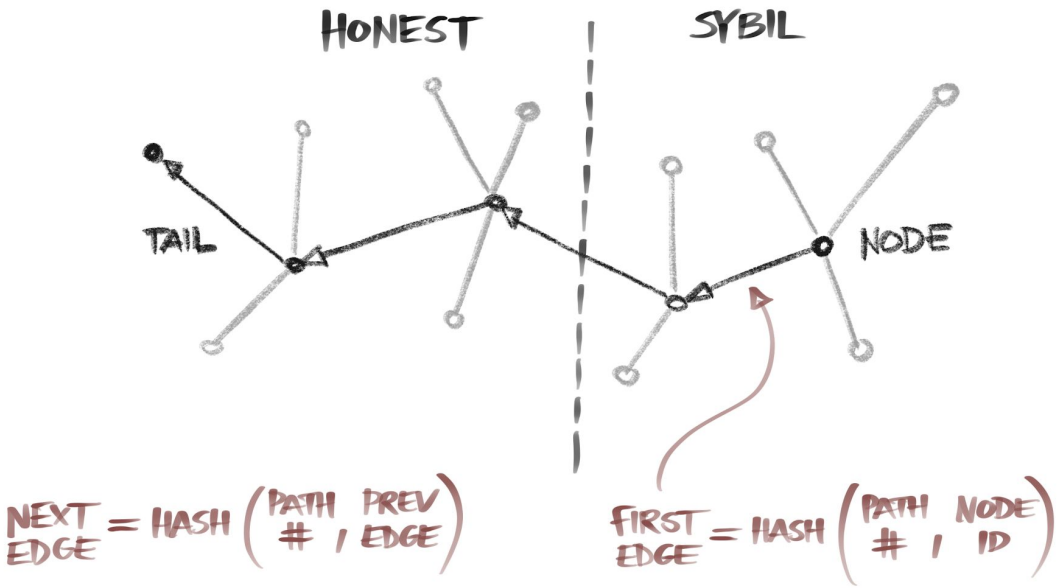
Verifiable paths



Verifiable paths



Verifiable paths



SYBILS CANNOT TAMPER
THE PATH
IN THE HONEST REGION

Proof of security

Theorem

MOST HONEST VERIFIERS:

— ACCEPT MOST HONEST SUSPECTS

— ACCEPT FEW SYBIL SUSPECTS

Theorem

$\geq (1-\epsilon)N$

MOST HONEST VERIFIERS:

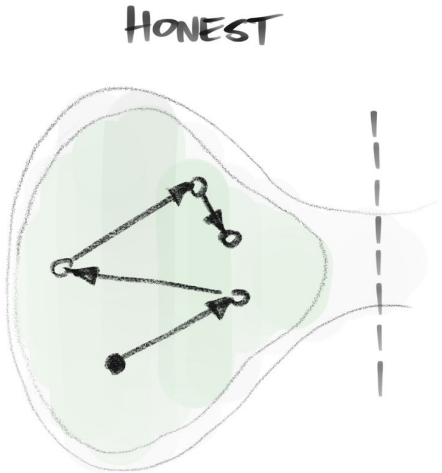
— ACCEPT $\geq (1-\epsilon)N$ MOST HONEST SUSPECTS

— ACCEPT $\leq \epsilon N$ FEW SYBIL SUSPECTS

N = number of honest nodes

$$\epsilon \approx \frac{\text{attack edges}}{\text{honest edges}}$$

First step



$\geq (1-\epsilon)N$

FOR MOST HONEST NODES,

THE TAIL OF A RANDOM WALK

IS

UNIFORM ACROSS

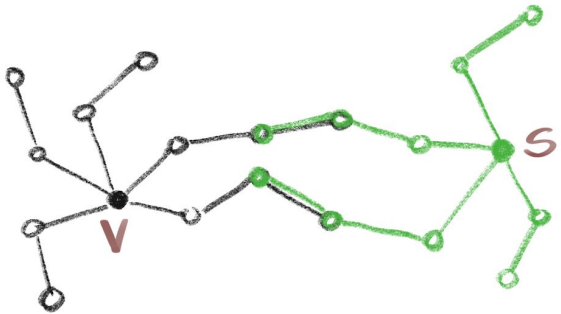
THE

HONEST EDGES

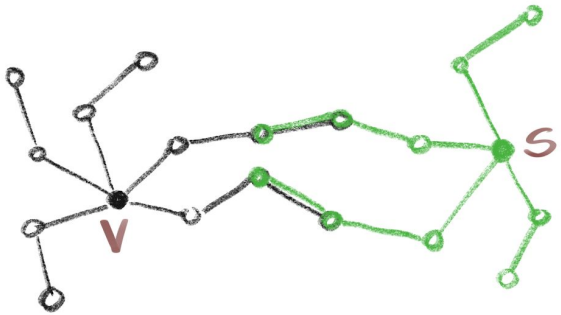
WITH HIGH PROBABILITY

$$\geq 1 - \frac{\text{attack edges} \cdot \log N}{N}$$

Accepting an honest suspect

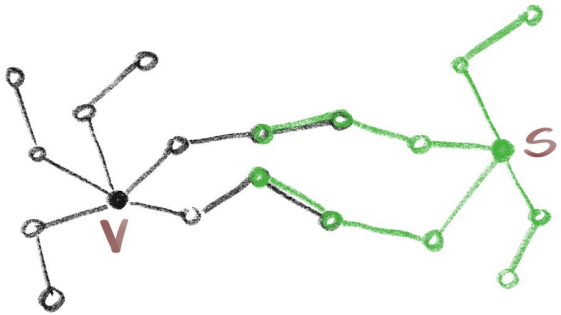


Accepting an honest suspect



DO $\sqrt{N}$ RANDOM EDGES
INTERSECT
$\sqrt{N}$ RANDOM EDGES?

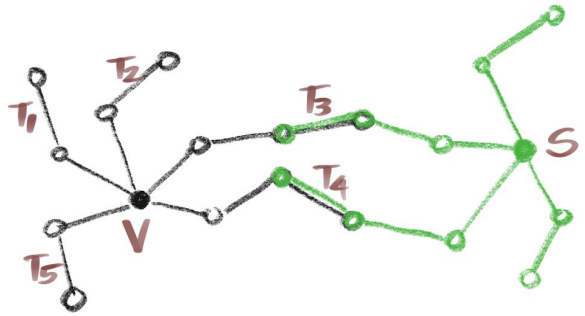
Accepting an honest suspect

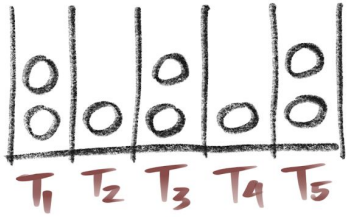


DO $\sqrt{N}$ RANDOM EDGES
INTERSECT
$\sqrt{N}$ RANDOM EDGES?

YES!
(BIRTHDAY PARADOX)

Accepting an honest suspect

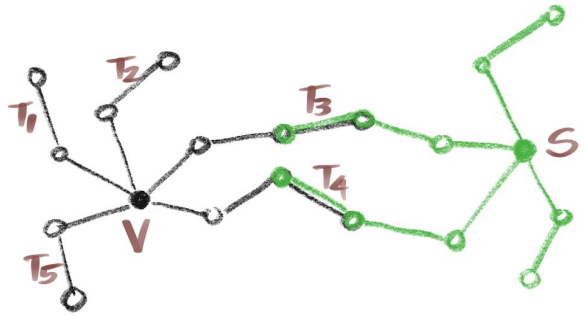




DO $\sqrt{N}$ RANDOM EDGES
INTERSECT
$\sqrt{N}$ RANDOM EDGES?

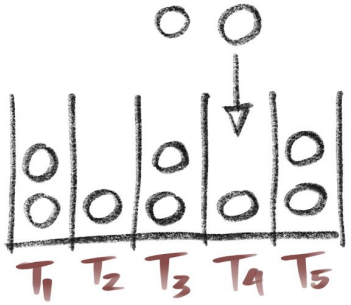
YES!
(BIRTHDAY PARADOX)

Accepting an honest suspect



Do $\sqrt{N}$ random edges intersect $\sqrt{N}$ random edges?

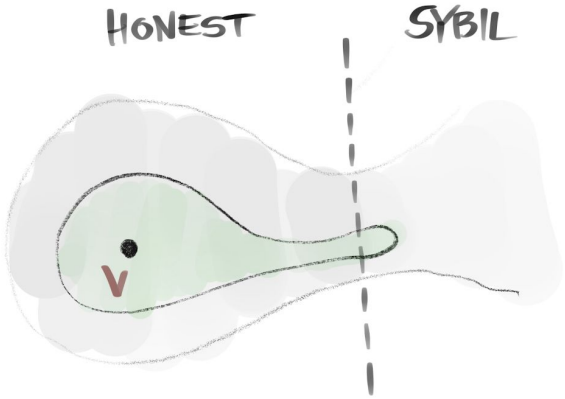
Yes!
(Birthday paradox)



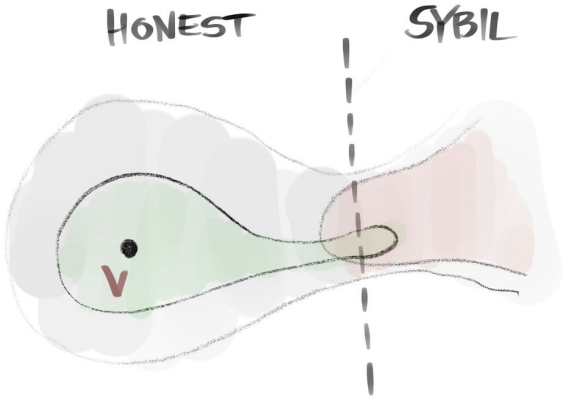
If the intersection has two or more tails there is one tail that can be inserted without violating balance

(Power of two choices)

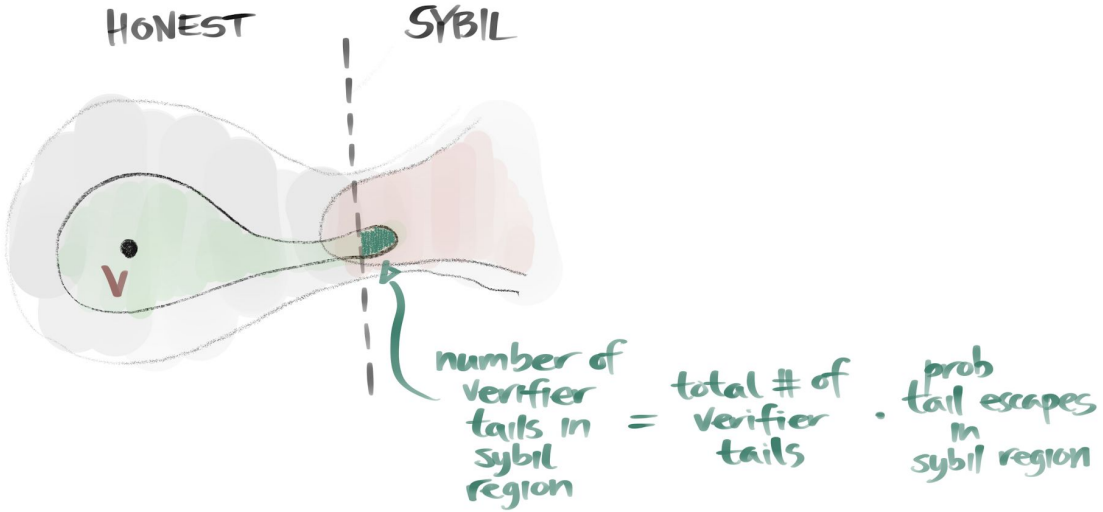
Rejecting a Sybil suspect



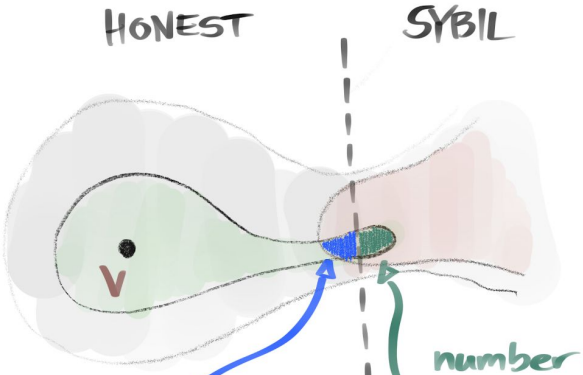
Rejecting a Sybil suspect



Rejecting a Sybil suspect



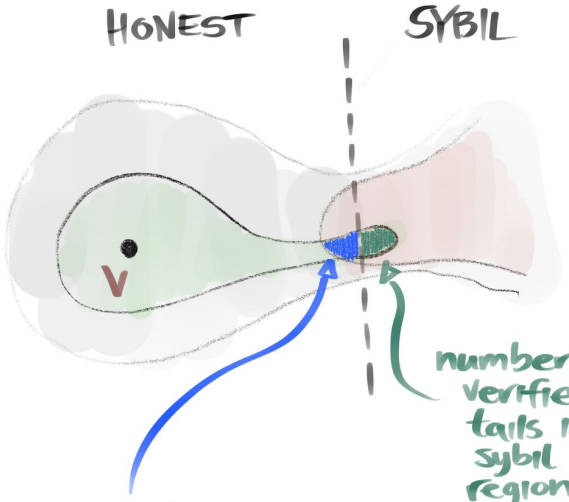
Rejecting a Sybil suspect



number of verifier tails in sybil region = total # of verifier tails · prob tail escapes in sybil region

number of verifier tails in honest region reachable by sybils = total number of honest tails reachable by sybils · prob verifier picks a tail · total number of verifier tails

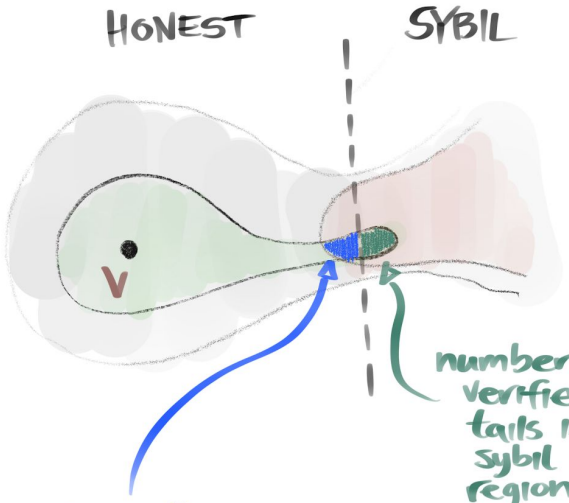
Rejecting a Sybil suspect

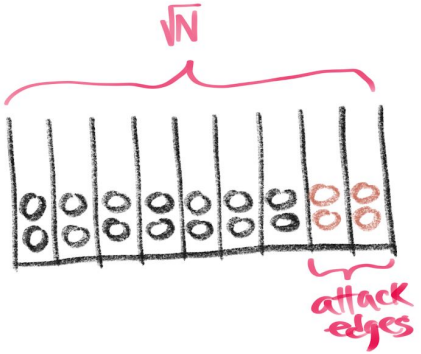


number of verifier tails in sybil region = $\sqrt{N}$ (total # of verifier tails) · $\frac{\text{attack edges} \cdot \log N}{N}$ (prob tail escapes in sybil region) < 1

number of verifier tails in honest region reachable by sybils = (total number of honest tails reachable by sybils: attack edges $\cdot \log N \cdot \sqrt{N}$) · (prob verifier picks a tail: $\frac{1}{N}$) · (total number of verifier tails: $\sqrt{N}$) $\approx$ attack edges

Rejecting a Sybil suspect





number of verifier tails in sybil region = total # of verifier tails $\cdot$ prob tail escapes in sybil region $< \frac{\text{attack edges} \cdot \log N}{N}$

number of verifier tails in honest region reachable by sybils = total number of honest tails reachable by sybils $\cdot$ prob verifier picks a tail $\cdot$ total number of verifier tails $\approx$ attack edges

attack edges $\cdot \log N \cdot \sqrt{N}$ $\cdot \frac{1}{N} \cdot \sqrt{N}$

The end

Thank you!

petar@protocol.ai