On the Feasibility of Spam-Protection Using GossipSub Peer-Scoring

Sanaz Taheri

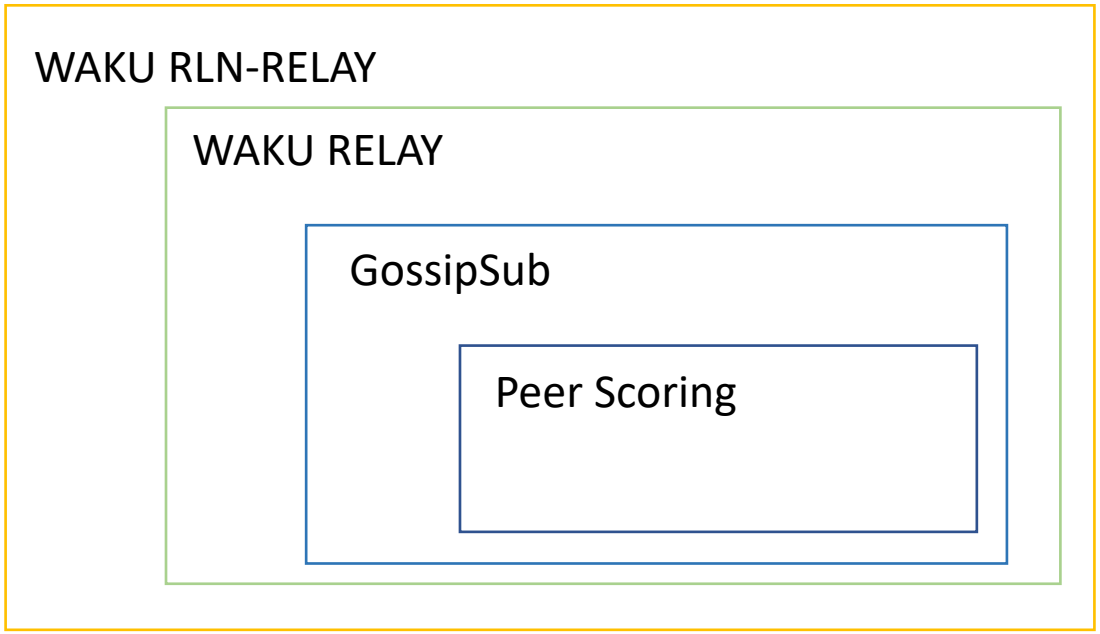
August 2022

Agenda

- Motivation
- GossipSub Overview
- Attacks on Open Permissionless Messaging Protocols
- GossipSub Main Components
  - Mesh Construction
  - Score Function
- GossipSub Mitigation Strategies
- Conclusion and Discussion

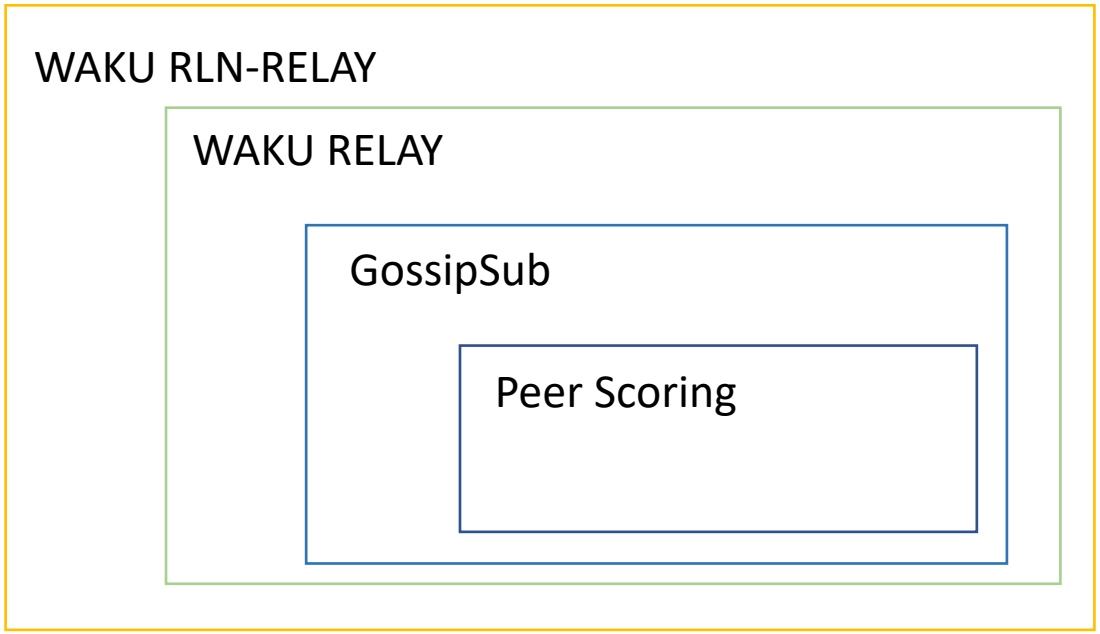
Motivation

- Understanding of GossipSub peer scoring is relevant to
  - Spam protection in WAKU-RELAY



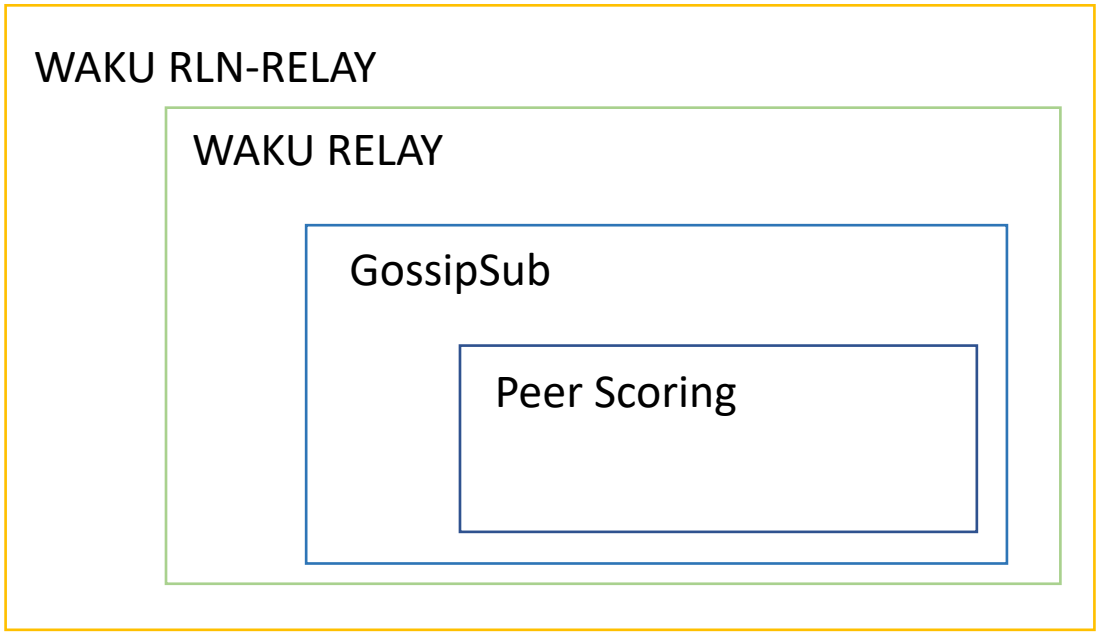
Motivation

- Understanding of GossipSub peer scoring is relevant to
  - Spam protection in WAKU-RELAY
  - WAKU Incentivization



Motivation

- Understanding of GossipSub peer scoring is relevant to
  - Spam protection in WAKU-RELAY
  - WAKU Incentivization
  - WAKU-RELAY Adversarial Model



GossipSub

- Gossip-based pubsub protocol
- Designed to deal with both **fast** and **resilient** message propagation in **permissionless** networks (mostly as a messaging layer of Blockchain environments)
- Resilient against a wide range of attacks, in contrast to past pubsub protocols
- Two main components: **Mesh construction** and **Score function** (+ mitigation mechanisms)

Attacks on Open Permissionless Messaging Protocols

- Attacks that target **Delayed message propagation** and **Forking the system**
- Can you tell a few?

Attacks on Open Permissionless Messaging Protocols

- Sybil Attack
  - Creating large numbers of identities
  - Sybils will attempt to get into the mesh, through a process called grafting (will explain later)
  - This is a first step for carrying out all of the following attacks
- Eclipse Attack:
  - Silencing victim by not relaying its messages or delaying its incoming messages
  - This attack can be carried out against a single victim or the whole network

Attacks on Open Permissionless Messaging Protocols

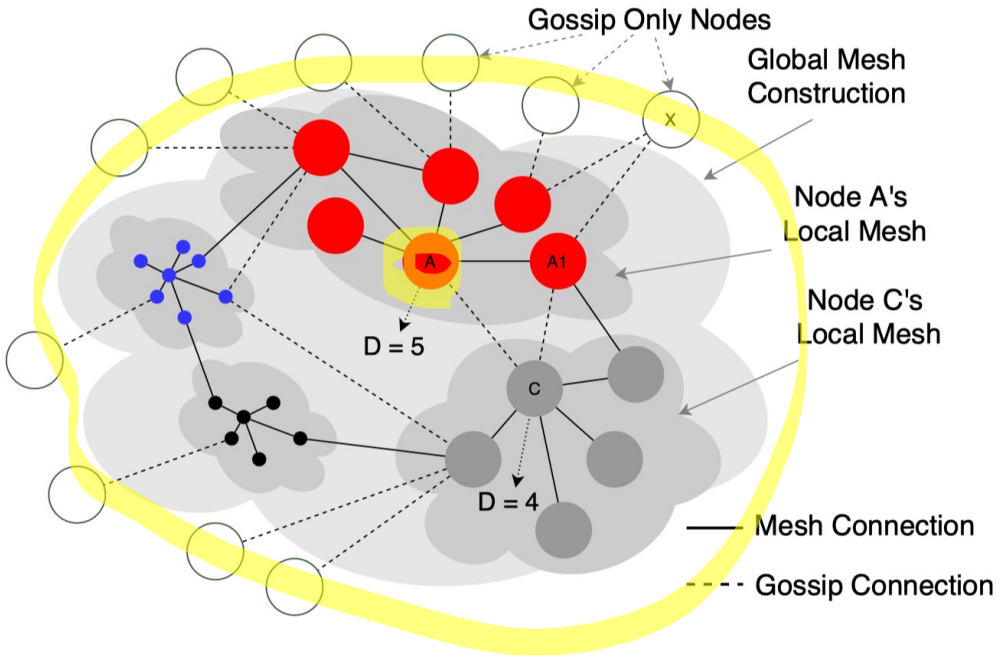
- Censorship Attack:
  - Sybils seek to establish themselves in the mesh and propagate all messages except those published by the target peer
  - Hard to detect by monitoring and scoring peers: Sybils build up score by propagating all other messages
- Cold Boot Attack:
  - The Sybils manage to largely take over the mesh since 1) honest and sybil nodes join concurrently when the network bootstraps 2) there is no score built up from an honest-only network to protect the mesh
  - Can happen in two cases:
    - i) when the network bootstraps with Sybils joining at time t_0
    - ii) when new nodes are joining the network while the network is under attack

Attacks on Open Permissionless Messaging Protocols

- Flash & Covert Flash Attack:
  - Sybils connect and attack the network at once
  - Sybils connect to the network but behave properly for some time in order to build up score, then they disrupt the network in a coordinated attack by not propagating messages
  - Difficult to identify since attackers behave properly and build a good profile before turning malicious

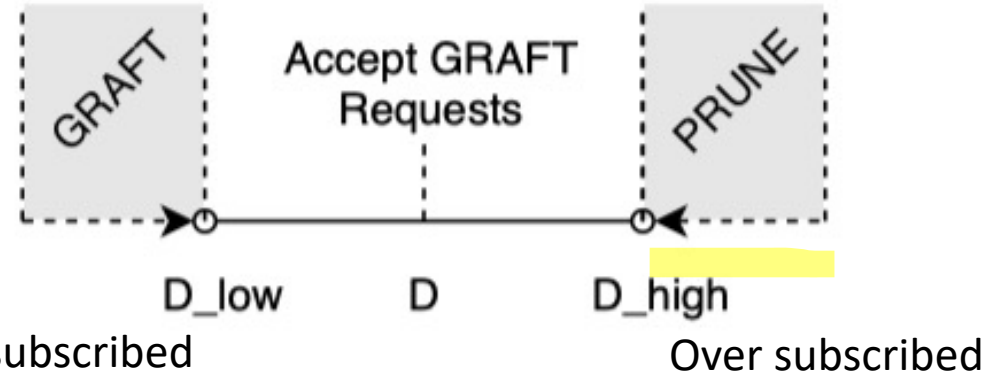
Mesh Construction

- Basic building block of the GossipSub protocol
- Each node maintains a list of peers with which it is directly connected with bidirectional, reciprocal links, forming its **local mesh**
- **Local Mesh**: The group of nodes that connects a peer to the global mesh
- Nodes can join and leave the mesh based on network-level conditions or application-level semantics
- **Eager push**: Mesh-connected nodes directly share messages with one another, realizing an eager push communication model
- Routing/publishing is done by broadcasting the message to the local mesh only
- **Gossip:** Nodes that are not part of the mesh communicate with mesh-connected nodes through gossip



Mesh Construction Parameters

- Mesh construction parameters include
  - D: the target degree
  - D_{low} and D_{high}: admissible mesh degree bounds
- **Over subscription**: the number of direct connections in the local mesh exceeds D_{high}
- **Under subscription**: the number of direct connections in the local mesh falls below D_{low}

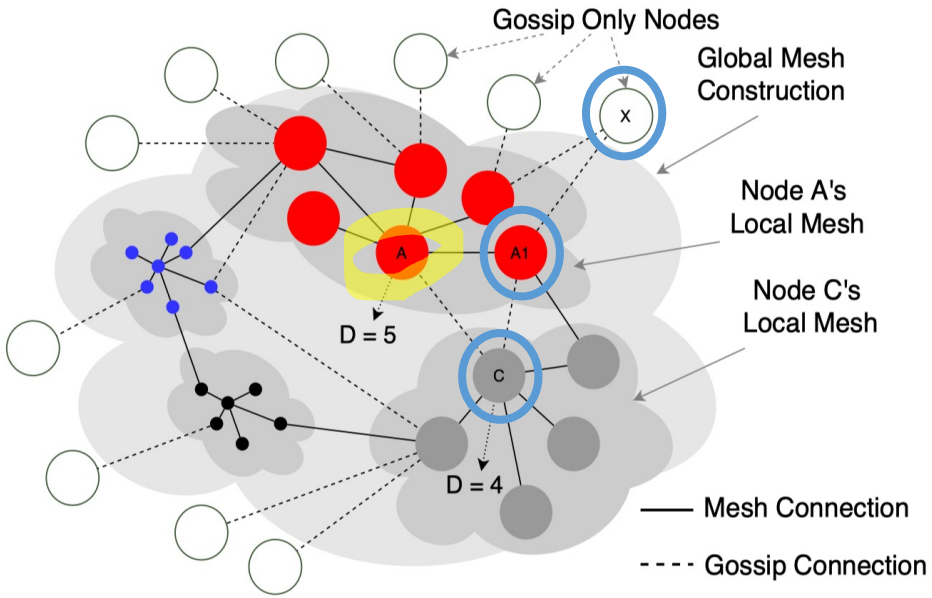


Protocol Control Messages

- **Graft a mesh link**: this notifies the peer that it has been added to the local mesh of the grafting node (notice it is push based and not requested by the connected peer).
- **Prune a mesh link**: this notifies the peer that it has been removed from the local mesh of the pruning peer.
- **PRUNE-Peer Exchange (PX):** the pruning peer sends a list of peer IDs to the pruned peer to help it connect to alternative peers and expand its mesh.
- **IHAVE: gossip**; this notifies the peer that the following messages were recently seen and are available on request.
- **IWANT: gossip**; request transmission of messages announced in an IHAVE message.
- A router periodically runs a heartbeat procedure, which is responsible for maintaining the mesh and emitting gossip. The value of the heartbeat is currently set to 1s.

Gossip Factor

- Gossip messages propagate arbitrary metadata
- Baseline is to include message IDs of the messages seen by the peer in the last 3 seconds
- Gossip is emitted to a random subset of peers that may or may not be part of the mesh.
- Gossip is emitted every heartbeat i.e., every 1 s



Score Function

- A performance monitoring mechanism to identify and remove badly-behaving nodes from the mesh
- Each peer in the mesh keeps a score for every other peer that it directly interacts with (including peers in or outside of the local mesh)
- Scores are not shared between peers
- Scoring function is a weighted sum of parameters with the weights being set according to environment conditions

Score Function

- Let's think about some of the parameters, what are good/bad behaviors? How would you parametrize them?

Score Function

- Some parameters are topic based
- t_i : topic
- $w(t_i)$: weight for topic t_i

$$Score(peer) = TC(\sum_{n=1}^{4} w_n(t_i) * P_n(t_i)) + w_5 * P_5 + w_6 * P_6 \quad (1)$$

Score Function

$$Score(peer) = TC(\sum_{n=1}^{4} w_n(t_i) * P_n(t_i)) + w_5 * P_5 + w_6 * P_6 \quad (1)$$

- P_1: Time in mesh for a topic
  - The time a peer has been in the mesh
  - Positive weight
  - Capped
  - To boost peers already in the mesh so that they are not prematurely pruned because of oversubscription
- P_2: Fist message deliveries for a topic
  - Number of messages first delivered by a peer in the topic
  - Positive weight
  - To reward peers who act fast on relaying messages
- P_{3a}: Mesh Message Delivery Rate for a topic
  - 0 if above the expected message delivery rate within the local mesh, square of deficit if below the expected rate
  - Negative weight
  - Capped
  - To penalize peers in the mesh who are not delivering the expected number of messages so that they can be removed from the mesh
  - In order to avoid triggering the penalty too early, the parameter has an activation window.

Score Function

$$Score(peer) = TC(\sum_{n=1}^{4} w_n(t_i) * P_n(t_i)) + w_5 * P_5 + w_6 * P_6 \quad (1)$$

- P_{3b}: Mesh Message Delivery Failures (for a topic)
  - The number of mesh message delivery failures
  - negative weight
  - Uncapped
  - Whenever a peer is pruned with a negative score, the parameter is augmented by the rate deficit at the time of prune
  - To keep history of prunes so that a peer which was pruned because of under-delivery cannot quickly re-graft onto the mesh
- P_4 : Invalid Messages (for a topic)
  - Number of invalid messages delivered in the topic
  - Negative weight
  - Uncapped
  - To penalize peers who transmit invalid messages, according **to application-specific** validation rules

Score Function

$$Score(peer) = TC(\sum_{n=1}^{4} w_n(t_i) * P_n(t_i)) + w_5 * P_5 + w_6 * P_6 \quad (1)$$

- P_5 : Application-Specific score
  - Assigned to the peer using application-specific logic
  - Wight is Positive
  - Has arbitrary real value
- P_6 : IP Address Collocation Factor
  - If the number of peers connecting from the same IP (of the connected peer) exceeds **an application-specific threshold**, then the value of P_6 is the square of the surplus, otherwise, 0.
  - Negative weight
  - To make it difficult to carry out sybil attacks by using a small number of IPs

Scoring Function

- Do you think it is fair that a large positive or negative score sticks for the lifetime of a peer?

Parameter Decay

- The counters associated with $\mathrm{P}_2, P_3, P_{3b}, P_4$ decay periodically by multiplying with a **configurable decay factor**
- The **decay interval** is configurable by the application

Mitigation Strategies

GossipSub Attack & Mitigation Strategy Summary					
Attack / Mitigation	Sybil	Eclipse	Censor	Cold Boot	Covert Flash
Backoff on PRUNE (Sec. 6.5)	✓				
Adaptive Gossip Dissemination (Sec. 6.4)	✓	✓			
Controlled Mesh Maintenance (Sec. 6.1)	✓	✓		✓	✓
Opportunistic Grafting (Sec. 6.2)	✓	✓	✓	✓	✓
Flood Publishing (Sec. 6.3	✓	✓	✓	✓	✓

Mitigation Strategies

Controlled Mesh Maintenance

- Problem:
  - The GRAFT mechanism of GossipSub creates an attack vector.
  - Malicious nodes can create multiple Sybil identities and send GRAFT messages to honest peers.
  - The peer becomes oversubscribed and **IF it starts randomly pruning** its current connection, they will (statistically) prune honest peers in favor of malicious ones.
  - The sybils take over node's local mesh.
- Mitigation 1: Selective score-based pruning when oversubscribed. The peer keeps the best scoring peers from the existing mesh, selects the rest to graft at random from its list of known/seen peers.
- Mitigation 2 (Outbound Mesh Connection): peers make sure to always have certain outbound connections. Attacking outbound connections initiated by the node is harder for the attacker.

Mitigation Strategies

Controlled Mesh Maintenance

- The mesh maintenance process is run on every heartbeat (i.e., every 1 s)
  - i) prune peers with negative score from the node's local mesh,
  - ii) choose only peers with positive scores to graft to, in case of under-subscription
  - iii) always maintain at least D_{out} outgoing connections.

Conclusion and Discussion

Observations About GossipSub Peer Scoring

- Peer scoring is a tool but not the solution.
- To use peer scoring, the bad or good behavior needs to be defined and formulated as a scoring parameter.
- GossipSub focuses on the attacks that cause **delayed Message propagation -> network fragmentation/fork.** Due to this, the existing topic–specific scores intend to capture **under-performing** peers e.g., first message delivery, failed message delivery, and message delivery rate.
- Spamming behavior i.e., high messaging rate is not captured by the existing scoring parameters neither with the mitigation strategies.

Scoring parameter for spamming behavior

- The application-specific score P_5 can be used to capture spamming behavior
- How? How to measure the spamming behavior i.e., high messaging rate of peer's connections and score them accordingly?
- Ideas?

Spam-Protection Using Peer Scoring

Solution Proposal:

- Define per-peer (PeerID) messaging rate (per topic)
- Each peer counts the number of messages routed by each of its mesh connections based on the PeerID of the message origin, when a message rate violation is detected for a PeerID, then then the score of the corresponding routing peer is decremented by one
- This enforces routing peers to monitor the network and control the message rate of each others

Spam-Protection Using Peer Scoring

Solution Proposal:

- Defi
- Def ach
  Pee
  me ing
  rou
- This trol the
  message rate of each others

WAKU-RELAY ≠ GossipSub

WAKU-RELAY = Privacy-Preserving GossipSub

Privacy-Preserving Spam-Protection Using Peer Scoring

Solution Proposal (not suitable for WAKU-RELAY):

- Define per-peer ~~(PeerID)~~ messaging rate (per topic)
- Define a score with a negative value that messaging rate of each PeerID routed by a peer in the mesh, if that rate exceeds the messaging rate, then decrement the score of the corresponding routing peer by one
- This enforces routing peers to monitor the network and control the message rate of each others

Privacy-Preserving Spam-Protection Using Peer Scoring

- WAKU-RLN-RELAY allows identifying spamming behavior in an private/anonymous messaging protocol including anonymous GossipSub
- WAKU-RLN-RELAY is incorporated into GossipSub routing protocol via application-specific message validators [1].
- Can scoring function be useful? Ideas?

[1] https://github.com/libp2p/specs/blob/master/pubsub/gossipsub/gossipsub-v1.1.md#extended-validators

Privacy-Preserving Spam-Protection Using Peer Scoring

- Can scoring function be useful? Ideas?
  - Prevention of lazy routing (free riding) where routing peers do not verify the RLN proof of messages to save on computation
  - The scoring parameter P_4 i.e., invalid message score can be used

Privacy-Preserving Spam-Protection Using Peer Scoring

- Can scoring function be useful? Ideas?
  - Prevention of lazy routing (free riding) where routing peers do not verify the RLN proof of messages to save on computation
  - The scoring parameter P_4 i.e., invalid message score is used

Resources

- Vyzovitis, Dimitris, et al. "GossipSub: Attack-resilient message propagation in the Filecoin and ETH2. 0 networks." arXiv preprint arXiv:2007.02754 (2020).
- GossipSub V1.1 Specification https://github.com/libp2p/specs/blob/master/pubsub/gossipsub/gossipsub-v1.1.md
- Taheri-Boshrooyeh, S., Thorén, O., Whitehat, B., Koh, W. J., Kilic, O., & Gurkan, K. (2022). WAKU-RLN-RELAY: Privacy-Preserving Peer-to-Peer Economic Spam Protection. *arXiv preprint arXiv:2207.00117*.
- 17/WAKU-RLN-RELAY Specification: https://rfc.vac.dev/spec/17/
- 11/WAKU-RELAY Specification: https://rfc.vac.dev/spec/11/


Thank You!

Mitigation Strategies

Opportunistic Grafting

- Problem:
  - Getting stuck with a mesh of poorly performed peers due to high churn rate of honest nodes, or a successful take over of a peer's mesh.
  - Multiple rounds of selection needed from a Sybil-poisoned pool before the mesh becomes healthy.
  - Due to large population of sybils, their penalties may decay before selecting good peers, thus Sybils become re-eligible for grafting.
- Mechanics:
  - Peers periodically check the median score of other peers in their mesh against a threshold.
  - If the median score is below the threshold, the peer opportunistically grafts extra peers with score above the median in the mesh.
- Mitigates successful mesh take-over and therefore, a poisoned pool mesh, seen in the Cold-Boot and Covert Flash Attacks.

Mitigation Strategies

Flood Publishing

- Problem: censorship or eclipse attacks
- Mechanics:
  - Every newly published message is sent to all known peers with a positive score that are subscribed to the topic. This applies regardless of whether the publisher is subscribed to the topic.
  - Side benefits: reduces message propagation delay at the cost of increased bandwidth.
- Mitigates: Counters all identified attacks where Sybils attempt to eclipse or censor a specific target or the whole network in either warm or cold conditions.

Mitigation Strategies

Adaptive Gossip Dissemination

- Problem: In Gossipsub v1.0 gossip is emitted to a fixed number of peers.
- Mechanics: Adaptive dissemination in GossipSub v1.1: In each round, every peer emits gossip to a **gossip factor** of its known peers that are not part of its local mesh.
- Mitigates: Enhances the resistance of the protocol against **eclipse attacks**, in case of significant number of Sybils.

Mitigation Strategies

Backoff on PRUNE

- Problem: GRAFT spam attacks i.e., fast regrafting of sybil nodes
- Mechanics: Pruning functionality is extended to add a backoff period before a pruned peer can attempt to re-graft.
- Mitigates: Prevents sybils from continues grafting and taking over the mesh of the node.