Abstract

This post proposes a Tor-based approach for relatively easy-to-deploy validator anonymity,

with a focus on protecting the next epoch's validator from DoS attacks.

The proposed approach uses the Tor network to anonymously push messages into the existing BN gossipsub network.

This post also compares our Tor-based approach to aDandelion++-based approach.

In summary:

- the well established Tor network can be used to make finding the next validator's network parameters significantly harder
- still, a custom onion routing solution for the validator/BN network is desirable as a future goal, but requires more R&D
- Tor and Dandelion both add latency; thus compete for the same resource
- the latency added by Tor or Dandelion is feasible
- compared to Dandelion, Tor offers more desirable anonymity properties for roughly the same added latency

Many thanks to @Menduist

for suggestions and feedback.

Privacy/Anonymity Goals

Unlinkability of Validator ID to IP Address

Validator IDs should not be linkable to the IP address (and peer-ID) of the beacon node they are connected to.

There are two main reasons

1. If an attacker is able to link a validator ID to the corresponding IP address, the attacker can identify proposer chosen for the next slot, learn its IP address, and DoS the corresponding node.
2. protecting the anonymity of validators

The main issue to solve is 1), even though it is not directly related to anonymity.

It is important to solve this issue, because it can actually be exploited and DoS Ethereum operation.

Future Anonymity Goals

Future goals that are currently out of scope comprise

- hiding participation in the beacon network

Dandelion Approach

Dandelion and its successor Dandelion++ are mitigation techniques against mass deanonymization. (The first sections of 44/WAKU2-DANDELION can be used as an overview over Dandelion's functioning.)

Dandelion has been investigated regarding its potential as a solution for validator anonymity in

Ethereum consensus layer validator anonymity using Dandelion++ and RLN conclusion

The conclusion of this analysis is that Dandelion is not a feasible solution.

Even though the latency added by Dandelion is lower than first assumed – see my comment on this post – the relatively poor anonymity properties that Dandelion offers still make it too expensive in terms of latency added for the gained anonymity.

Dandelion was designed for Bitcoin (with longer block-times and looser latency-bounds), where the latency cost is no issue and the anonymity gain (even though small) comes basically for free.

Further research analysing the relatively small anonymity gain: On the Anonymity of Peer-To-Peer Network Anonymity Schemes Used by Cryptocurrencies.

Dandelion is a message spreading method

, which increases the uncertainty of an attacker when trying to link messages to senders.

Unlike Tor, it does not guarantee unlinkability in a relatively strong attacker model.

Expected Latency cost

The expected added latency for Dandelion stem is ~500ms (assuming 100ms average latency between nodes and 5 hops on average).

For now, we ignore an optional random delay added on each fluff hop.

Issues

Since in stem phase, Dandelion messages are sent to a single peer, resilience is significantly worse than plain gossipsub (which has a recommended mesh out-degree of 8).

This requires a fail-safe, where sending nodes store a message and re-send it if they don't receive it via gossipsub after a random time in the range of the expected latency.

This adds to the expected latency.

The main issue with Dandelion is that it only provides mitigation and no unlinkability guarantee.

The application to the Beacon network adds to this problem:

messages in the beacon network are inherently linkable to validators IDs.

This allows attackers to link messages to specific sources, where only the network parameters (IP address, peer-ID, …) are unknown.

Such an attacker is much stronger than the attacker model the Dandelion papers use,

which breaks the anonymity guarantees of Dandelion.

An attacker can (try to) connect to peers (sending graft) from which he receives messages originated by a specific validator.

Dandelion will make this process longer and more prone to failures compared to plain gossipsub,

but eventually an attacker can detect the network parameters of the originator.

Tor Approach

The following gives an overview over a potential Tor-based approach to validator anonymity.

Prerequisites

The proposed Tor-based approach requires a way of allowing validator/BNs to push messages to one (or more) beacon node(s) over Tor circuit(s).

We propose tor-push, which allows message originators to push messages over Tor to gossipsub peers.

Tor-push uses the same protocol ID as gossipsub, which makes it fully backwards compatible.

Messages sent via tor-push are sent via a separate libp2p tor-switch, which forwards messages over Tor. (This requires SOCKS5 support for libp2p.)

Tor and non-tor switches must never be mixed; attackers must not be able to link these two switches.

Non-tor switches must not be used as a fail-safe for tor-push messages.

The tor-switch

- is not subscribed to any pubsub topic;

- only sends messages the validator/BNs originates, while the non-tor switch handles the typical gossipsub tasks;
- connects to a separate set of peers (via Tor), randomly chosen via a discovery method (discv5).

Functioning

Beacon nodes receiving a tor-push message relay this message via gossipsub.

Since tor-push messages are typical gossipsub messages,

every BN can act as such a diffuser node, even if it does not support tor-push itself.

(Because hiding participation is not a goal for now, tor-push is not offered as an onion service.

This saves latency, as the number of hops is only 3, and allows backwards compatibility.)

Validators can either directly send messages they originate via tor-push,

or let the beacon node they are directly connected to (and which is controlled by the same entity) send tor-push messages.

Validator/BNs send messages to D

(gossipsub mesh out-degree) diffuser beacon nodes.

This keeps resilience at a level similar to gossipsub, and is a significant advantage over Dandelion.

Per default, tor-push connections are kept open for one epoch.

The connection life-time can be adjusted as a trade-off between efficiency and anonymity (further analysis necessary).

Establishing circuits for a given epoch can be done ahead of time (in the preceeding epoch).

Since D

connections are established, at least some of them are expected to be ready for their respective epoch. Establishing connections ahead of time avoids adding latency to message delivery.

Issues

The following is a list of known issues (non-comprehensive):

- Malicious guards could identify validator traffic because it features distinct patterns, and correlate it to specific messages
- e.g. specific validators send attestations in specific slots
- padding / cover traffic could mitigate this; still needs further investigation
- naive solution: each validator that does not attest in a given slot sends a dummy attestation
- naive solution: each validator that does not attest in a given slot sends a dummy attestation
- task: identify all patterns specific to Validator/BN network traffic
- e.g. specific validators send attestations in specific slots
- padding / cover traffic could mitigate this; still needs further investigation
- naive solution: each validator that does not attest in a given slot sends a dummy attestation
- naive solution: each validator that does not attest in a given slot sends a dummy attestation
- task: identify all patterns specific to Validator/BN network traffic
- similar to website fingerprinting, an attacker between the victim node and the Tor network could identify and correlate validator traffic
- also mitigate with padding, cover traffic
- also mitigate with padding, cover traffic
- Using Tor for hiding validators could incentivise large scale DoS attacks on Tor

- also, cannot check message validity until messages reach a diffuser node, which might be abused for spam
- however, this requires an attacker with lots of resources, which could DoS the current network, too
- also, cannot check message validity until messages reach a diffuser node, which might be abused for spam
- however, this requires an attacker with lots of resources, which could DoS the current network, too
- the discovery mechanism could be abused to link requesting nodes to their Tor connections to discovered nodes
- an attacker that controls both the node that responds to a discovery query, and the node who's ENR the response contains, can link the requester to a Tor connection that is expected to be opened to the node represented by the returend ENR soon after
- an attacker that controls both the node that responds to a discovery query, and the node who's ENR the response contains, can link the requester to a Tor connection that is expected to be opened to the node represented by the returend ENR soon after
- the discovery mechanism (e.g. discv5) could be abused to distribute disproportionately many malicious nodes
- e.g. if p% of the nodes in the network are malicious, an attacker could manipulate the discovery to return malicious nodes with 2p% probability
- the discovery mechanism needs to be resilient against this
- e.g. if p% of the nodes in the network are malicious, an attacker could manipulate the discovery to return malicious nodes with 2p% probability
- the discovery mechanism needs to be resilient against this

Even though these are potential attack vectors, the proposed Tor approach makes finding the network parameters (e.g. IP address) of the next validator significantly more difficult.

Further Issues + Solutions

The Tor approach requires validator/BNs to setup a tor daemon.

The overhead for operators can be significantly reduced by bundling tor with the validator/BN software (cmp.Tor Browser).

If there are only a few validators/BNs using Tor, attackers can narrow down the senders of Tor messages to the set of BNs that do not originate messages.

This could be ignored, explaining that anonymity guarantees only hold when a certain percentage of BNs support the Tor approach.

Validators who want anonymity guarantees from day one on should have separate sets of network parameters for their non-tor and tor switches, respectively.

For the best protection, the tor-switch and gossipsub switch can be run on separate physical machines.

Latency

For now, we assume an added latency around 500ms, similar to Dandelion.

Experimental evaluation of the impact of Tor latency on web browsingcan be used as a reference.

(I could work on more analysis of tor latency, if desired.)

Note: Tor has since introduced congestion control, further reducing average latency.

Also, the analysis linked above measures RTT not latency.

The effect of broken circuits has to be investigated, but opening D

connections should mitigate the effect.

As connections are established ahead of time (see Functioning), connection establishment does not add additional latency to message delivery.

Dandelion vs Tor-based solution

Advantages of Tor

- offers significantly better anonymity properties
- relatively high resilience; same as gossipsub (same out-degree), while Dandelion has an effective stem out-degree of 1
- easier to deploy (even though Dandelion is relatively easy to deploy, too)
- fully backwards compatible; could be started by a single validator (Dandelion is also incrementally deployable, but needs critical mass to be useful)

Advantages of Dandelion

- can check message validity at each stem hop
- does not rely on an external anonymization network

In our opinion, the main advantage of Tor – offering significantly better anonymity properties – clearly outweighs the advantages of Dandelion.

Combined Solution

Tor and Dandelion could be combined:

Validator/BNs use tor-push to introduce new messages to the gossipsub network, and diffuser BNs feature Dandelion.

A message first gets routed though Tor and then along a Dandelion stem.

The Dandelion stem would make it more difficult for attacks to link messages to the nodes that received said message via Tor.

While this adds further mitigation against correlation attacks, it seems not enough to justify the added latency.

Current conclusion: adding Dandelion would roughly double the added latency,

and the anonymity added by Dandelion does not seem worth it.

Integrated Onion-Routing/Encryption Approach

We propose the Tor-based solution as an intermediate solution, while researching the integration of onion routing/encryption into gossipsub.

The main advantages of the Tor-based solution for now is:

it can yield a significant anonymity gain very soon.

Integrating a custom onion routing solution into gossipsub takes much more R&D time.

In the long run, the integrated solution has several advantages:

- avoids having to bundle the tor daemon and depending on an external anonymization network
- allows specific tweaking to better fit the Ethereum beacon network
- for instance, it can be aware of the fact that beacon messages follow strict rules
- for instance, it can be aware of the fact that beacon messages follow strict rules
- could include spam protection: ZK proof in each onion layer