In this post, I outline various considerations regarding network-level validator privacy and things to take into account before settling on any particular methods/strategy.

I'm looking for feedback on these considerations and more ideas.

Overview

Validator privacy concerns itself with masking/obfuscating (or potentially) removing the link between validator's IP addresses with their on-chain identities. The rationale is to make attacking validators much harder in order to increase the overall robustness of the network.

Desiderata

In order to maintain the scalability benefits of sharding and proof of stake, we need to consider how much privacy is necessary (or even possible) in order to determine what protocols would suit the needs of the system.

- Low latency

: Latency has a direct impact on the profitability of being a validator. It is reasonable to assume that a validator will most likely prioritize being profitable and making money over being private and potentially vulnerable to attacks.

- Low bandwidth

: Bandwidth affects the accessibility of being a validator. A major goal of the current ETH2.0 design is being able to run a validator client on low-resource devices.

- Accountable

: If we were to adopt a pure anonymous protocol for validators (hint: we can't anyway by the first two criteria), this can leave the network vulnerable to sybil and other kinds of DoS attacks on the network.

- Availability favoring

: The current protocol is designed to value availability over consistency. As such, applying an anonymous communication protocol that favors the opposite will obviously have bad repercussions at the consensus layer.

Current Solutions

Given the above desiderata, we can gauge what currently proposed or deployed ACN (anonymous communication network) can potentially be used to for our purposes.

- Tor

: It's the most well-known and well-studied ACN. Moreover, the tooling is there for integration. The main ramifications is it's effects on latency and bandwith, although it is a low latency favoring network. Moreover, it has had demonstrable attacks occur and has well known scalability issues.

- i2p

: It's less studied than Tor and has less tooling for it. It has been deployed for a long-time and hasn't been attacked as far we can tell. It's design is also better suited for the p2p nature of ETH2.0's design as it employs a p2p architecture instead of a client-server architecture.

- Mixnets

: Provides anonymity in the face of a global passive adversary (think NSA) but none have been deployed for applications other than email/chat. They tend to be low latency and high bandwidth and have a client-server architecture.

- Onion Routing

: I separated this as there are other onion routing designs other than Tor. In particular, Hornet has a lot of desirable properties and provides more theorectical security than Tor. However, there are no working implementations as of this writing.

Other ACNs designs proposed in the literature aren't suited for networks running thousands of nodes and are thus ignored in this section.

Open Questions

There are still many open questions regarding theses criteria and current solutions. Some of which I will outline here.

- How does using an ACN affect the fault tolerance of the overall network?

Here has been no work done to see how fault tolerant ACNs are. In fact, there has been no work to show how they react in the case of crash faults. This needs to be explored more.

- How does composing ACNs affect the overall robustness of the network?

Composing ACNs with orthogonal properties has been proposed in the literature to overcome the anonymity trilemma. However, this hasn't been done in practice. Exploring the affects of composing ACNs is worth exploring in the goal of providing the best validator privacy possible.

- Given our conditions, what kind of threat model is sufficient?

: By the anonymity trilemma, the strongest threat model comes at the expense of having a practical ACN. In other words, if we want validators to be safe from a global active/passive adversary, we would need to give up the requirement of low latency and low bandwidth. Ovbiously, this is unreasonble. Thus, what kind of threat model is sufficient for providing some form of validator privacy? My initial thoughts lean towards the onion routing threat model as they are well tested and are reasonable in practice despite some of the attacks on it.