Migrated from research forum. Original author: yush

I wanted to propose a new AVS plugin based around generating consensus between a web2 value and the same value on-chain, but we had some questions about the security of such a design. I was curious if anyone had any feedback on other oracle designs or considerations/concerns before we build such a system. This post will consider such a design from the perspective of DNS, but the same system can be used for general consensus on globally consistent (i.e. public) off-chain values.

One of the desires we have with zk-email powered apps (https://prove.email) is that we would like to provide hard economic guarantees on the security of the oracle that fetches the DNS value of the DKIM public key for any website, and stores that value in a smart contract. For context, this value usually rotates every 6 months or so, at which there will be a short-lived coordination failure.

The general design of such an system would likely be a plugin which queries DNS at each validator, then generates a signature and posts it on-chain. In our case, the posted value will be a mapping of (domain [string], selector [string]) to a 2048 bit RSA key, along with a signed message per validator. No one is slashed while at least 90% of validators agree on the DNS value. Every 6 months, when the key rotates and caches take anywhere between 0-48 hours to catch up, there will likely be disagreement. During such a period, the key exposed in the smart contract can be set to null (and thus zk-email apps using that key halted) until 90% of eigenlayer nodes agree on the value again. We expect we can greatly reduce this time by optimistically flushing all caches when the value changes, but in the worst case there may still be a significant delay.

I have some questions about how to deal with slashing and such a design in Eigenlayer.

- Do we slash the disagreeing 10% of nodes if there is i.e. only 90% agreement? We want to minimize existential risk to Ethereum as well as incentivize validators to participate in the protocol, so we worry that validators may disagree due to factors out of their control. For instance, if a country implements a firewall on such DNS queries, or some other failure happens, we want to ensure validators' stakes can stay safe. At the same time, we don't want to invite the ability for an attacker to lurk on the network with 10% of the stake, although maybe this is OK.
- If someone attempts to attack the network by deploying several validators to pass the 90% threshold during a key rotation, and attempts to post a malicious value, we want to provide guarantees on the cost of such an attack. Unfortunately, we don't see an easy way to differentiate between the situation in which the website posting the DNS value posts a short term value that they i.e. change twice within 48 hours, and in which nodes attack the network. I don't yet understand how Eigenlayer nodes can recover from such an attack and guarantee that a malicious actor gets slashed (i.e. is there some way to slash-them post-hoc?). If there is no guaranteed way to slash them post-hoc, then the economic cost would only be the cost of holding Ethereum for 2 days, which seems nearly negligible.

We think there is benefit to using Eigenlayer that outweighs using a Schellingcoin-style system (like Uma) in which escalation games provide economic security. My main concern with escalation game designs is that the economic security of each query would depend on the amount of collateral we can put up per query, which doesn't seem like a scalable way to secure data for any non-defi project.

We would like to hear more from the team and community about these specific questions as they are the main barriers for us implementing such a system. Alternatively, we are happy to wait until Eigenlayer implements their own oracle system, but we would like clarification on how such a system would deal with these issues before opting to wait for its release.