Migrated from research forum. Original author: turnoffthiscomputer

Hi everyone,

I'm currently working on Proof of Passport. We are developing open source circuits and a mobile app that lets users scan the NFC chip in their government-issued passport and generate proofs of humanity, nationality and age. With zero-knowledge proofs we can allow users to disclose only the attributes they want public. This enables uses cases in proof of unique identity, sybil resistance and selective disclosure of private data.

The way passport signature verification is done is through a certificate chain passing through multiple authorities. Without getting into too much details, the file containing personal data inside passport chips (SOD file) is signed by a Document Signing Certificate (DSC) which is itself signed by a Country Signing Certification Authority (CSCA) which is itself included in the Masterlist signed by the ICAO Master List Signer Certificate .

All those verifications can be made trustlessly in the zero-knowledge circuit. But there needs to be a final source of truth for the ICAO public key, which rotates every ~3 years. As we will be moving away from multisigs towards a more future-proof setup, we will want to provide cryptoeconomic guarantees on the correctness of this public key on chain. This problem is a very similar to the one encountered by zk-email.

Current options include ICP (as used here by zk-email) and Chainlink Any API, which both provide oracle services for arbitrary HTTPS request when there is no provable source of truth. Chainlink does not have slashing implemented except for ETH/USD price feeds and it looks that ICP does not have slashing implemented at all. This seems to be because the only slashing they could do is minority slashing, which works well if a few nodes provide inaccurate data but is catastrophic in the case of a successful attack, as honest nodes get slashed.

Our question here is the following: could EigenLayer be used to provide quantified cryptoeconomic guarantees on an HTTPS querying service? A naive implementation with BLS aggregation and minority slashing doesn't seem sufficient.

A better approach would be to add EigenLayer restaking on top of a TLS Notary network. This could allow slashing not when the information provided is not true (which cannot be checked) but when nodes leak their secret (in TLSN, shares of the symmetric TLS encryption key used to send and receive requests). This could prevent nodes from coordinating. If any team is working on that or on other similar designs, please reach out to us at @colinremi

or @FlorentTavernier

on telegram!