Background reading

- Viaduct paper. In the language used by Viaduct, Anoma is a “runtime”, not a language. There are other differences - e.g. Anoma will need to compute acceptable protocols dynamically at runtime, not once in a compiler, and programs for Anoma are written for a local perspective view and composed, not so much top-down like Viaduct - but those can be covered separately in further analysis; we just need something simple for now.

Context

Anoma sends many different kinds of messages over the network - intents, consensus votes, P2P metadata gossip, etc. - and eventually we will want information flow control for all of them. Initially, however, I suggest we start simple, with the most important data - intents (represented by partial transactions). Intents are typically sent by a user to the intent gossip network, sometimes directly to a solver, sometimes not. They then may be gossiped amongst multiple solvers, with partial solving happening, before being submitted to the mempool, ordered by consensus, and executed. The user who authors an intent may want precise control over where the information in their intent goes - perhaps they have a set of trusted solvers, but don’t want any data leaked to the consensus provider. I propose that we craft the right “type” for this kind of precise control, but limit the instantiations to the most useful, simple ones at first (as full analysis of compositional information flow control will take awhile).

Proposal

Specifically, I propose that we add a field to partial transactions called information_flow_predicate

, of type PTX \to ExternalIdentity \to {0 | 1}

. Conceptually, this predicate is a way for the author of the partial transaction to instruct nodes (especially solvers) who may be processing it who they are allowed to forward the results of processing it to.

Runtime

At runtime - e.g. in the engines related to solving, and wherever else partial transactions are processed, when constructing a partial transaction which includes (by composition) a previously received partial transaction, and deciding whether or not to send this new partial transaction to a particular node (identified by their external identity), Anoma MUST send this new partial transaction only if this predicate returns true, when called on the new partial transaction and the external identity in question.

Instantiation

Initially, instead of being instantiated by an arbitrary function (which would be difficult to analyze), this predicate MUST be instantiated by one of the following options:

data InformationFlowPredicate = AllowAny | AllowOnly (Set ExternalIdentity) | RequireShielded (Set Hash) | And (Set InformationFlowPredicate) | Or (Set InformationFlowPredicate)

In English:

- AllowAny

allows any derived partial transaction to be sent to anyone.

- AllowOnly

allows any derived partial transaction to be sent to any of the specified external identities.

- RequireShielded (Set Hash)

requires that the included hashes have been wiped from the partial transaction extra data (and associated resource proofs created, in order to make the derived partial transaction valid).

- And (Set InformationFlowPredicate)

requires that all included predicates in the set are satisfied. For example, one might use And (RequireShielded (...), AllowOnly (...))

to only allow members of the listed identity set to view the derived partial transaction, and only after the listed resources have been shielded.

- Or (Set InformationFlowPredicate)

requires that one of the included predicates in the set is satisfied. For example, one might use Or (RequireShielded, AllowOnly (...))

to allow anyone to view the derived partial transaction after the specified resources are shielded, or only members of the listed identity set to view it beforehand.

I think that these predicates should be relatively easy and computationally efficient for a solver to analyze.

Partial transaction composition

When partial transactions are composed, their information flow predicates MUST be and-ed together (e.g. with And (Set InformationFlowPredicate)

), to ensure that the properties desired by all prior partial transactions are satisfied.

Thoughts & feedback welcome.