- Resource machine:

defines the state to be stored and the security requirements

- Storage:

cares about how to store the state

The stored state is represented as a key-value store. The goal of this discussion is to specify the structure of this key-value store: what would be the keys and what would be the values for each state component recognized by the resource machine: CMtree, NFset, data blobs (think encrypted resources)?

A

B

key

value

CMtree

leaf

timestamp + cm

1

node

prefix of the timestamp

node value (the hash of the child nodes)

NFset

nf

1

Mutable indices

kind

hash

Data blobs

hash

blob

Timestamps

Associate a timestamp with each tree node s.t. lower depth tree node corresponds to a less specified timestamp: a parent node timestamp is a prefix of the child node timestamp. The leafs of the tree would have fully specified timestamps: t1. * . * - t1.t2. * - t1.t2.t3

Questions

- What to use as a key for data blobs?
- What would be the implications of using resource kind as a key for data blobs?
- If we use a resource kind for data blob keys, can we organise the resource kinds in any kind of hierarchy?
- What would be the security implications of using resource kinds as data blob keys?
- Note: the value stored under the key in the case we use the resource kind as a key would be determined by the resource kind
- What would be the implications of using resource kind as a key for data blobs?
- If we use a resource kind for data blob keys, can we organise the resource kinds in any kind of hierarchy?

- What would be the security implications of using resource kinds as data blob keys?
- Note: the value stored under the key in the case we use the resource kind as a key would be determined by the resource kind
- Do we want to determine the position of the next commitment added to the tree sequentially? It would allow to conveniently forget about some parts of the tree after a certain points, and whoever has unconsumed resources in that part of the tree would only need to store the sub-tree path (and fetch the rest). What are the security implications?
- Do we want to associate an expiration date with each resource/nullifier and stop storing the data once the resource has expired?