NAMELESS OBJECTS

Marc Mosko (marc.mosko@parc.com)

ICNRG Interim (Paris, FR) January 14-15, 2016

parc
A Xerox Company

NAMED ADDRESS (W/O NAMELESS OBJECTS)

- The full address of a CCNx Interest is:
  - {Name, [KeyIdRestr], [ObjectHashRestr]}
- A ContentObject is:
  - {Name, [KeyId], ObjectHash}
- In CCNx1.0 without Nameless Objects, a ContentObject matches an Interest if and only if this predicate is true:

```
Interest                                ContentObject
Ni = Name (always exists)               No = Name (always exists)
Ki = KeyIdRestr (may be empty)          Ko = KeyId (may be empty)
Hi = COH Restr (may be empty)           Ho = Hash (always exists)
```

$$(Ni=No)\ \&\ (\nexists\ Ki\ |\ (Ki=Ko))\ \&\ (\nexists\ Hi\ |\ (Hi=Ho))$$

HASH-BASED NAME

- Name and KeyId are not strong names
  - A publisher (in CCNx or NDN) can published many payloads with the same (Name, KeyId) pair.
  - CCNx ContentObjects (and NDN Data) are not immutable under only (Name, KeyId).
- The Hash restriction (or NDN hash name component) is the only way to name immutable.
  - If one can name something by its Hash, then the Name and KeyId are largely irrelevant, as far as naming goes.
  - If one is worried about hash collisions or hash attacks, then naming something with (KeyId, Hash) would require the collision to have a known string towards the end of the message (and TLV parse correctly).

WHAT IS A NAMELESS OBJECT

- A ContentObject without a Name.
  - It is identified by {[KeyId], ObjectHash}.
- A (Nameless) Content Object matches an Interest if and only if this predicate is true:

```
Interest
Ni = Name (always exists)
Ki = KeyIdRestr (may be empty)
Hi = COH Restr (may be empty)
```

```
ContentObject
No = Name (may be empty)
Ko = KeyId (may be empty)
Ho = Hash (always exists)
```

$$(\nexists\ No \mid (Ni=No)) \ \&\ (\nexists\ Ki \mid (Ki=Ko)) \ \&\ (\nexists\ Hi \mid (Hi=Ho)) \ \&\ (\exists\ No \mid \exists\ Hi)$$

Terms specific to Nameless Objects

parc
A Xerox Company

TRUST CHAIN

- A Nameless object may be signed
  - Or, it could only include a KeyId as part of the packet for matching.
  - In any case, a signature does not imply trust. Some external mechanism must assert that the public key is to be trusted.
- A Nameless Object does not imply trust.
  - It only implies that one receives the immutable object named by {[KeyId], ObjectHash}.
- A trust chain for immutable objects is a function of how one learns the {[KeyId], ObjectHash}.
  - Will not be discussed here, but properly constructed manifests plus external system on public keys could achieve it.

OTHER BENEFITS OF THIS CONSTRUCTION

- One can mix Nameless and Named objects.
- A Nameless object can come from anywhere
  - The Name in an Interest is a locator used to find the {[KeyId], ObjectHash} pair.
- Because it can come from anywhere, couldn't you poison caches?
  - No, because a Nameless object has no name! It can only be requested by Hash, so there's no possibility of poisoning a cache. Either it's the immutable object you want or it isn't.
  - Cache poisoning would require that someone requesting the object without a Hash, such as by {Name, [KeyId]}, but that will not match a Nameless Object.

DOWNSIDE

- It can only be delivered by {[KeyId], ObjectHash}
  - Every intermediate system needs to know ObjectHash.
  - Thus, every system has to calculate ObjectHash, which is likely a SHA256 or a SHA512-256. That will add latency at each hop.
  - But, you do get truly immutable objects!
- Possible solutions
  - Within a trust domain (e.g. autonomous system), ingress router computes ObjectHash and puts in a hop-by-hop header. An ingress router should always remove the header.
  - Use a different switching technique, like PIT-less solutions on intermediate or core routers, only do expensive evaluation on the edge.

CONCLUSION

- Supporting nameless objects requires:
  - Making the Name optional in a Content Object
  - Changing the matching predicate

$$(Ni=No)\ \&\ (\nexists\ Ki\ |\ (Ki=Ko))\ \&\ (\nexists\ Hi\ |\ (Hi=Ho))$$

becomes

$$(\nexists\ No\ |\ (Ni=No))\ \&\ (\nexists\ Ki\ |\ (Ki=Ko))\ \&\ (\nexists\ Hi\ |\ (Hi=Ho))\ \&\ (\exists No\ |\ \exists Hi)$$

- Can come from anywhere without renaming (and re-signing).
- Can mix nameless and named objects in one system.