**About Digest Support in the KeyLocator**

AlexA: *My opinion was and is that this is completely unnecessary to have anything except name in key locator.*

Actually, I believe that you at one point agreed with this change. :) See my 10/29/13 email to Lixia, Van, kc, with cc to Jeff T & Alex A had this in it: “We discussed including the publisher public key digest as a KeyLocator option for some potential simplifications for ongoing communication in which the full name of the key is unnecessary. I think that Lixia and Alex agreed with this, though we also all agreed that name engineering will be needed in certain cases to keep names short, too.”

*Without certificate name in key locator, it will be unclear for anybody else in the network to fetch the key.*

This is not necessarily true; it just means that the KeyLocator cannot be used *on its own* to fetch the key - but I am not sure this has to be a design goal. Many names are interpreted in the context of their applications; why would keys be any different?[[1]](#footnote-1) For example:

* Segmented data that in the first segment provides the KeyLocator with the full name, but in later segments uses the digest. This is a convention where it could be straightforward on both the publisher and consumer side. I can’t see a good reason not to allow this, and it seems to offer some benefits.
* Application-specific naming conventions where the key name can be derived from the data name if necessary to do a fetch of the key.
* Applications that have conversations, where the signing key is fixed at the beginning of the conversation.

Further, not all keys (certs) need to be published and named in a way that is globally routable, so it’s not true that having a name means the key is fetchable by someone who got ahold of a packet. In fact, if the data is intended to be verified only by consumers that already have the key, isn’t providing the fingerprint a non-ambiguous way to make that clear in the data packet, rather than providing a name that doesn’t resolve or (worse) reveals internal information about the content?

A not too well thought out example: Consider a sensitive health data packet with an encrypted name as required by federal law: /org/humana/pt/<encrypted-patient-name-and-other-things> Why should the KeyLocator for this data be understandable to anything outside of Humana? In this case, perhaps allowing digest is a reasonable alternative to /<externally-meaningless-prefix>/<obscure–or-encrypted-key-name> This is a little bit of a stretch, but the point is I’m not sure that every node on the network will understand the KeyLocator even if it is a name, so lack of fetchability doesn’t seem like a good argument.

It occurred to me that it seems ambiguous in the spec whether the Name in the KeyLocator must only one object. (Rather than a prefix that if included in an Interest could return multiple objects.) This needs to be clarified.

1. For this reason, I could be convinced of a *naming convention* within the KeyLocator that indicated a local-to-the-consumer key digest, such as the equivalent of /\_digest/<key\_digest>. This relates to the marker naming conversation. [↑](#footnote-ref-1)