RFC3344 Issue:

Selectors for the FA-HA Security Association

- The origin of this issue is the question which was raised some time ago of how to handle de-registrations in the case where a mobile node is registered with one FA, but desires to de-register a previous session with another FA.

- In this case, if the de-registration is sent through a new FA which has a security association with the HA and therefore adds a FA-HA Authentication-enabling extension, **and** the wrong information is used by the HA to select the SA (Security Association) used to verify the FA-HA Auth.Ext, the verification will fail.

- We discussed a solution to this at the last IETF, but have not reached closure.

- It turns out that this may be a deeper issue about what should be the selectors of the FA-HA Security Association.

- Whatever selectors we choose, it is important to understand that they serve two purposes:

1. The sender of a message must find the right SA with which to compute the authentication extension, or determine that an authentication extension is not necessary.

2. The receiver of a message must find the right SA with which to check the authentication extension, or to determine that a message without an authentication extension should be dropped.

(Both ends of the selector computation must be done in exactly the same way to guarantee interoperability.)

- Possible Registration Request Selector Fields:
  a. The Source IP address
  b. The Destination IP address
  c. The Care-of Address
  d. The Home Agent Address
  e. The Lifetime field (or a bit which says whether or not it is zero)
  f. The D bit (set to 1 if co-located CoA is being used)

- Current status quo in the 3344bis text is (c,d) with a proposal to add (e).

- Before we consider adding (e) we need to get our story straight on why (c,d) is such a good choice. We run into difficulty with (c,d) when using co-located CoA registering via the FA. A possible solution to this difficulty is to configure duplicate SAs for all CoAs that could be used from a given FA, but this seems like way too much state and doesn't handle the case where more than one FA are deployed on a given link.

- Another possible approach is to go back to using (a,d) instead of (c,d). Of course, NATs can change the value of (a) and so this is why 3519 mandated (c,d) and a=c. However, my understanding is that most current implementations (those that don't do 3519) actually use (a,d) even though this has been ambiguous ever since RFC 2002. (a,d) seems to be the most straightforward approach and matches e.g., IPSec style processing.

Thoughts?