IEEE P802.21  
Media Independent Handover Services

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proposed revision to Annex N.6 (DCN# 21-13-0064-00) | | | | |
| Date: 2013-04-08 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Charlie Perkins | Futurewei |  |  | Charlie.Perkins@huawei.com |

Abstract

This document contains a proposed revision to Annex N.6 to resolve comments received during Letter Ballot 802.21c Draft/D02. It shows revised descriptions for the steps illustrated in figure N.1, and raises some questions about the proper division of functionality between the MIHF and the MIH application.

## MIH\_Prereg\_Xfer messages for Optimized SA Establishment

### OPoS distributing key derivation key *K* to TPoS and MN



Figure N‑1 -- OPoS distributing key derivation key *K* to TPoS and MN

The signaling diagram illustrated in Figure N.6 1 shows the following steps

1. MIH\_Prereg\_Xfer.request: the MN user application asks to initiate preregistration to a suitable target PoA (TPoA)
2. MIH\_Prereg\_Xfer Request: MN’s MIHF selects Nonce-T and transmits request to Originating PoS (OPoS)
3. MIH\_Prereg\_Xfer.indication: OPoS MIHF presents MN’s Request to OPoS MIH application.
4. MIH\_N2N\_Prereg\_Xfer.request: OPoS MIH application identifies TPoS (if not already identified) and requests MIHF to relay information to enable TPoS to identify suitable TPoA and compute MISK
5. MIH\_N2N\_Prereg\_Xfer Request: OPoS MIHF creates Nonce-N and *K*, and relays Nonce-T in message to TPoS, possibly encapsulated with IPSec
6. MIH\_N2N\_Prereg\_Xfer.indication: presented to TPoS MIH application for extraction of *K*.
7. TPoS MIH application provides MIAK and any other appropriate keys to AAA for future authentication purposes
8. TPoS MIH application computes MNmsrk from *K* and sends appropriate LL frames to TPoA for key distribution, (i.e., MSPMK between MN and TPoA), and any other preregistration tasks.
9. MIH\_N2N\_Prereg\_Xfer.response: TPoS MIH application sends response (containing MN\_NAI) to OPoS.
10. MIH\_N2N\_Prereg\_Xfer Response: TPoS relays message to OPoS containing MN\_NAI.
11. MIH\_N2N\_Prereg\_Xfer.confirm: OPoS presents message to OPoS MIH application containing MN\_NAI.
12. MIH\_Prereg\_Xfer.response: OPoS MIH application initiates message to MN user application via OPoS containing MN\_NAI.
13. MIH\_Prereg\_Xfer Response: OPoS relays message to MIHF running on MN containing MN\_NAI, *K*.
14. MIH\_Prereg\_Xfer. confirm: MIHF running on MN relays message to MN user application containing MN\_NAI, *K*.
15. MN user application extracts *K*, computes MNmsrk, continues any necessary preregistration activities
16. MN continues with additional preregistration signaling

Questions:

1. At TPoS, does MIHF or MIH application compute MSPMK? [step 8] <my guess: MIH appl.>
2. If MIH application computes MSPMK, how does MIH application get *K* <my guess: primitive gets *K*>
3. At OPoS, does MIHF compute *K,*  fetch *K*otpos ,and encrypt using PNFotpos ? [steps 4 and 5]
4. At MN, does MIHF compute *K,*  and decrypt using PNFotpos ? [step 15]
5. If (as in question 2 and step 7) TPoS application sees *K*, it means that any such MIH application can see *K*. Does this mean that we should imagine (or even specify) that MIH application should be able to extract *K* from MIHF (perhaps by way of another function from MIH\_application 🡪 MIHF)?