**IEEE P802.21  
Media Independent Handover Services**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Proposed Remedy for the 802.21d LB7 comment #44** | | | | |
| **Date:** 2013-09-16 | | | | |
| **Author(s):** | | | | |
| **Name** | **Affiliation** | **Address** | **Phone** | **Email** |
| Yoshikazu Hanatani | Toshiba |  |  | yoshikazu.hanatani@ toshiba.co.jp |

Abstract

This document contains proposed remedy for “the 802.21d ballot 7 comment #44 about multicast ciphersuites.

**Remedy for the 802.21c LB6b comment #44**

**9.4.6 Multicast Ciphersuites**

The ciphersuites used for securing multicast MIH message is defined in Table 26.

**Comments**

**We agreed that IEEE 802.21d do no support ECDSA-224. (# 220)**

**MAC\_Algorithm\_for\_Verify\_Group\_Key is optional.**

**For simplicity, “Encryption Algorithm for Group Manipulation” and “Encryption Algorithm for Group Command” can be merged as follows.**

**—Table 26 Multicast Ciphersuites**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Encryption Algorithm** | **Digital Signature Algorithm** | **MAC Algorithm for Verify Group Key** |
| 10000000 | NULL | NULL | NULL |
| 10010101 | AES\_CCM-128 | ECDSA-256 | AES\_CMAC-128 |
| 10010100 | AES\_CCM-128 | ECDSA-256 | NULL |
| 10000100 | NULL | ECDSA-256 | NULL |

**All MNs and PoSs shall mandatory support the code “10000101”.**

**For updating the multicast ciphersuites, Section 9.4.3 should also revise as follows.**

**9.4.3 Multicast message encryption based on group key**



1. **—Key derivation example**

When an MN successfully recovers a GKB, it obtains a master group key (MGK). The following two keys are derived from MGK:

* Group key confirmation key (MIGKCK) used as a key confirmation key to confirm that the correct MGK is obtained through a Message Authentication Code (MAC);
* Group encryption key (MIGEK) used to protect the MIH group command.”

The deriving key is specified by the different multicast ciphersuites described in 9.4.6. For the key derivation, the following notations and parameters are used.

* *K*: key derivation key. It is truncated from a master group key (MGK). The length of *K* is determined by the pseudorandom function (PRF) used for key derivation. If HMAC-SHA-1 or HMAC-SHA-256 is used as a PRF, then the full MGK is used as key derivation key, *K*. If CMAC-AES is used as a PRF, then the first 128 bits of MGK are used as derivation key, *K*.
* *L*: The binary length of derived keying material MIGSK. *L* is determined by selected multicast ciphersuites described in 9.4.6.
* *h*: The output binary length of PRF used in the key derivation. That is, *h* is the length of the block of the keying material derived by one PRF execution. Specifically, for HMAC-SHA-1, *h* = 160 bits; for HMAC-256, *h* = 256 bits; for CMAC-AES, *h* = 128 bits.
* *n*: The number of iterations of PRF in order to generate *L*-bits keying material.
* *c*: The multicast ciphersuite code is a one octet string specified for each ciphersuite. The code is defined in 9.4.6.
* *v*: The length of the binary representation of the counter and the length of keying material L. The default value for *v* is 32.
* “MIGSK”: 0x4D4947534B, ASCII code in hex for string “MIGSK.”
* [a]2: Binary representation of integer *a* with a given length.

For given PRF, the key derivation for MIGSK can be described in the following procedures:

**Fixed input values**: h and v.

**Input**: *K*, *L*, and multicast ciphersuite code.

**Process**:

1. If *n* > 2v-1, then indicate an error and stop.
2. Result(0) := empty string.
3. For i = 1 to *n*, do
   1. *K*(i) := PRF(*K*, “MIGSK” || [i]2 || *c* || [*L*]2).
   2. Result(i) = Result(i-1) || *K*(i).
4. Return Result(n) and MIGSK is the leftmost *L* bits of Result(n).

**Output**: MIGSK.

The MIGSK is parsed in such a way that

MIGSK = MIGKCKK || MIGEK.

With the above procedure, a key hierarchy is derived as shown in Figure 47.