|  |  |
| --- | --- |
| Project | **IEEE 802.21 MIHS****<**[**http://www.ieee802.org/21/**](http://www.ieee802.org/21/)**>** |
| Title | **Proposed remedy for Comments 64 and 65** |
| DCN | **21-14-0096-00-MuGM** |
| Date Submitted | **May 14, 2014** |
| Source(s) | Subir Das (ACS) |  |
| Re: | IEEE 802.21 Session #62 in Waikoloa |
| Abstract | This document describes a proposed remedy for LB7c Comments #64, and #65 about Subclause 9.5.1.3 |
| Purpose | For LB7c Comment Resolution |
| Notice | This document has been prepared to assist the IEEE 802.21 Working Group. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. |
| Release | The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE’s name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE’s sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that IEEE 802.21 may make this contribution public. |
| Patent Policy | The contributor is familiar with IEEE patent policy, as stated in [Section 6 of the IEEE-SA Standards Board bylaws](http://standards.ieee.org/guides/opman/sect6.html#6.3) <[http://standards.ieee.org/guides/bylaws/sect6-7.html#6](http://127.0.0.1:4664/cache?event_id=757737&schema_id=1&s=5X0vID10lu_E6yrIkWkNd4Wz2H8&q=hancock)> and in *Understanding Patent Issues During IEEE Standards Development* <http://standards.ieee.org/board/pat/faq.pdf> |

# Comments



# Suggested Remedy:

**Delete Section 9.5.3.1 and modify the Section 9.5.2 as follows:**

**9.5.2 GKB generation by the complete subtree method**

A GKB contains a complete subtree (CompleteSubtree) part and a group key data (GroupKeyData) part. A GroupKeyData part appears when a GKB is used to deliver a group key. When GroupkeyData part does not exist, CompleteSubtree part is used to identify the groups.

CompleteSubtree identifies all the complete subtrees covered by the group such that each member represented by a leaf node in the group belongs to one and only one complete subtree. Each complete subtree is identified by its root node. Each root node has a Node Index. A Node Index is a pair of binary strings (Node Depth, Node Index Value). The first parameter, Node Depth, indicates the binary length of the index of the root node represented by one octet. The second parameter, Node Index Value, is a binary string representing the index of the root node.

The Node Depth is represented by an octet. The corresponding integer L is the binary length of the index of the root node representing the complete subtree. The Node Index Value is represented by ⎡L/8⎤ octets, where ⎡x⎤ is the ceiling function, which takes the minimum integer, which is larger than or equal to x. For example, ⎡1.2⎤ = 2. With the ceiling function, to demonstrate, if the value of a Node Depth L is between 1 and 8, the size of the binary string used to represent the Node Index Value is 1 octet (= 8 bits). If L is between 9 and 16, the size of the binary string used to represent the Node Index Value is 2 octets (= 16bits). A Node Index Value is left aligned in the network byte order. A Node Index Value shall have a zero padding added to the right. An example of Node Index is (0x05, 0b10011000). This Node Index represents the node ‘10011’, which is the root node of the complete subtree in a key tree. The depth of the key tree is a

system parameter and configured to the applications. If the key tree has depth 8, then the root node ‘10011’ identifies a depth-3 complete subtree. Another example of Node Index is (0x0e, 0b1100101000011100),which represents the node ‘11001010000111’. If it is in a depth-16 key tree, then the node identifies a depth-2 complete subtree. IEEE P802.21d/D4.0, April 2014

A complete subtree part is a list of Node Indices. The Node Indices are ordered based on the 1 following rule. Let (L1, I1) and (L2, I2) be two Node Indices. (L1, I1) appears in the front of (L2, I2), if and only if Int(d, I1) < Int(d, I2), where Int(d, Ih) is a function to convert binary index Ih to an d-bit long integer. It uses Ih as the binary values in the most significant bits of a d-bit integer. For example, if d=8, (0x05, 0b10011000) is converted to 27+24+23 = 152. When another Node Index is (0x08, 0b00011001) (with index 00011001 and convert to 24+23+1=25), then (0x05, 0b10011000) appears first. A group key data part of a GKB is a sequence of ciphertexts of the encrypted group key, where a group key is encrypted by Node Keys corresponding to all the root nodes of the complete subtrees covered by the group. There is a one-one correspondence between the complete subtree part and the group key data part in a GKB. The number of Node Indices, i.e. the number of complete subtrees, in the complete subtree part is equal to the number of ciphertexts of the encrypted group key in the group key data part. The n-th ciphertext of the group key is generated by the Node Key assigned to the root node designated by the n-th Node Index. AES\_Key\_Wrapping-128 or AES\_ECB-128 is used for the encryption algorithm.

In the example illustrated in Figure 33, the group with nodes 000, 001, 010, 011, 101, 111 is covered by three complete subtrees at root node 0, 101, and 111. Therefore, the complete subtree part in the GKB includes Node indices for node 0, 101, and 111. The key data part, the ciphertexts of MGK encrypted by k(0), k(101), and k(111) as indicated in Figure 34.



 **Figure 34—GKB for the group with nodes 000, 001, 010, 011, 101 and 111**

Notice that the example is for illustration purpose. The key tree has depth 3, not a multiple of 8. The Node Index Value is not illustrated with the binary string in Figure 34. In fact, Node Index (0) can be represented as (0x01, 0b000000000), Node Index (101) as (0x03, 0b10100000) and Node Index (111) as (0x03, 0b11100000).

There are two ways group members can be identified in a GKB when only CompleteSubtree exists in a GKB. The following methods are used to identify the group members. :

Method 1: The set of leaf nodes specified by the complete subtree part of the GKB represents the members who belong to the group

Method 2: The set of leaf nodes specified in the complete subtree part of the GKB represents the members who do not belong to the group. In other words, the complete subtree part represents the complement set of the leaf nodes.

For example, in a depth-3 group management tree, the set of all the leaf nodes is S = {000, 001, 010, 011,100, 101, 110, 111} and the group consists of members with leaf nodes in a set is; A = {000, 001, 010, 011,100}. When Method 1 is used, the complete subtree part shall represent set A, while when Method 2 is used, the complete subtree part shall represent S-A = {101, 110, 111}.

In order for a recipient to distinguish the two methods, a group manipulation command accompanies a flag named ComplementSubtreeFlag. If the flag is 0, Method 1 is used. If the flag is 1, Method 2 is used. The ComplementSubtreeFlag thus helps the recipient to correctly interpret the complete subtree part of a GKB.

A command center has a component called GKB Generator. A GKB Generator receives all the device keys assigned to all the recipients associated to a group and a MGK. The MGK is a master group key for that group. recipients and the CC can generate a Media Independent Group Session Key (MIGSK) from MGK (see 9.6). The mechanism to provide all device keys to the GKB generator is out of the scope of this specification. This mechanism can just encompass the explicit provision of the device keys to the GKB Generator or the random seed used to derive them. On receiving those data, a GKB Generator outputs a GKB, or several GKBs.