IEEE P802.21
Media Independent Handover Services

|  |
| --- |
| Introduction to key tree, complete subtree, and device key |
| Date: 2013-11-xx |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Lily Chen | NIST | 100 Bureau Dr. Gaithersburg, MD 20878 | +1 (301) 975-6974 | Lily.Chen at nist.gov |

Abstract

This contribution provides introduction on key tree, complete subtree, and device key to resolve comment #107.

**9.4.1 Key distribution for multicast MIH message protection**

Multicast MIH messages are protected through group session keys. A group key can be a master group key (MGK) or a group session key derived from an MGK. An MGK is distributed through a group manipulation command. The command carries a special data field, called group key block (GKB), in which the encrypted MGK is included.

It is assumed that each MN is provisioned with a set of device keys. Depending on the members in the group, an MGK is encrypted by a specific set of device keys so that each member in the group can decrypt one of the encrypted versions to obtain the MGK. An MN belonging to the group can use one of its device keys to recover MGK, from which it can derive the group session keys used to protect the multicast MIH messages.

**9.4.1.1 Device key assignment through a key tree**

The device key assignment mechanism specified in this standard is based on a binary key structure, called a key tree. A key tree is a binary tree in depth d, where d is a system constant. The root of the key tree is called a level 0 node. At level k, there are 2k nodes, 0 ≤ k ≤ d. Each level k node can be represented as a k-bit string. The string is called the index of the node. For example, when d > 1, the level 2 nodes are represented by the indices 00, 01, 11, 10. Each node is assigned to a key, called a node key and identified by the node index. For example, the node keys assigned to level 2 nodes are denoted as k<00>, k<01>, k<10>, k<11>.

For a key tree in depth d, all the level d nodes are called leaf nodes. Each MN is assigned to a leaf node. For an MN assigned to a leaf node with index x0x1…xd-1, d device keys k<x0>, k<x0x1>, …, k<x0x1…xd-1> are provisioned. Figure x illustrates a key tree in depth 3. Table x lists the device key assignment for each MN represented by the leaf node index.

0

1

00

01

10

11

000

001

010

011

100

101

110

111

**Figure x. A key tree in depth 3**

Table x: Device key assigments for MNs through a depth-3 key tree



When determining the system constant d for a key tree, 2d must be no less than the number of all the MNs to be grouped in the system. [shall we also introduce the leaf number here?]

**9.4.1.2 Complete subtree**

A complete (k-depth) subtree is a subtree which has a group of 2k leaf nodes such that their indices have a common prefix of d-k bits. For the key tree in Figure x, nodes represented with indices 000 and 001 form a 1-depth complete subtree, while nodes represented with indices 000, 001, 010, and 011 form a 2-depth complete subtree.

When the device keys are assigned through a key tree as introduce in 9.4.1.1, all the MNs corresponding to the leaf nodes in a complete subtree share a common node key (as one of the device keys), determine by the d-k bit prefix. In the example above, the MNs corresponding to the leaf nodes represented with indices 001 and 001 share node key k<00>. Therefore, a subset of MNs in a group correspond to the leaf nodes in a complete subtree can decrypt the MGK encrypted by the common shared key.

In order to distribute a MGK to a group of MNs, the first step is to sort corresponding leaf nodes to non-overlap complete subtrees so that each leaf node belongs to one complete subtree. Note that a single leaf node can be considered as a depth-0 complete subtree. In this case, the MGK will be encrypted by the minimum number of keys for the group.

In the example illustrated in Figure x, the group represented by the leaf nodes 000, 001, 010, 011, 101, 110 can be sorted as one 2-depth complete subtree 000, 001, 010, 011 and two 0-depth complete subtree 101 and 110. For the MNs in such a group, the MGK shall be encrypted by k<0>, k<101> and k<110>.

The algorithm of sorting a group of leaf nodes to non-overlap complete subtrees is introduced in …[Yoshi’s algorithm, see doc #xxx].