|  |  |  |
| --- | --- | --- |
| Project | **IEEE 802.21d**  **<<http://www.ieee802.org/21/>>** | |
| Title | Suggested Remedy for IEEE 802.21d Lb7b comments #79, #81, #82, #100 | |
| DCN | 21-14-0057-01-MuGM | |
| Date Submitted | **April, 4th, 2014** | |
| Source(s) | Antonio de la Oliva (UC3M), Yoshikazu Hanatani (Toshiba) |  |
| Re: | IEEE 802.21 Session #61 in Beijing | |
| Abstract |  | |
| Purpose |  | |
| 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" \l "6.3" \t "_parent) <[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" \t "_parent)> and in *Understanding Patent Issues During IEEE Standards Development* [http://standards.ieee.org/board/pat/faq.pdf](http://standards.ieee.org/board/pat/faq.pdf" \t "_parent) | |

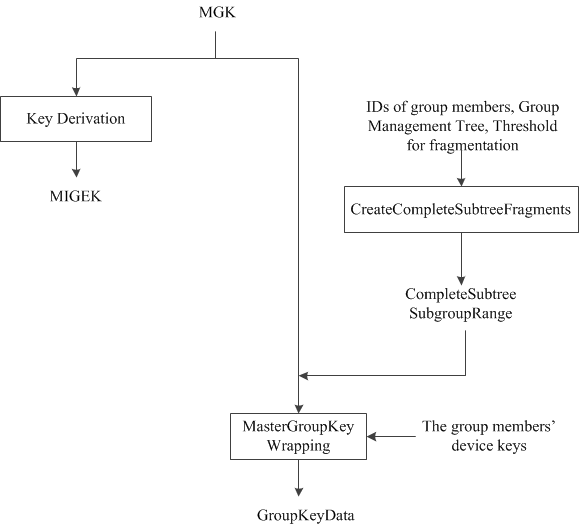
* + - * 1. MIH user of a GMCS

Required components in an MIH User of a GMCS in a PoS relevant to group manipulation and group commands are listed as follows:

* A GKB Generator. This component is constructed from CreateCompleteSubtreeFragments (see 9.4.2.3), KeyDerivation (see 9.5) and MasterGroupKeyWrapping (9.4.2.1).
* A *Group Management Tree Information Base* (of type GRP\_MGT\_TREE\_INFO\_BASE as defined in Table F.25). This information base contains following information.
  + The list of nodes that can be managed by groups.
  + (Optional) The MIHF\_IDs and device keys of all nodes managed by the GMCS. A group management tree can be derived from this information.
* A *Managed Group Information Base* (of type MANAGED\_GROUP\_INFO\_BASE as defined in Table F.25). This information base stores the information about a group which is managed by the GMCS. It contains the MIHF Group ID, the different group members, the MGK (an optional) used and the transport addresses for multicast (an optional).

A Flow diagram of the generation process of the GKB parameters is given in Figure 24. The MIH User generates MIH\_Net\_Group\_Manipulate.request described in 7.4.3.2 as follows:

1. Define a group to manipulate:
   1. Decide a TargetGroupIdentifier which indicates a group to manipulate.
      * If it is a new group, choose the TargetGroupIdentifier by consulting with the *Managed* *Group Information Base*.
   2. Decide group members belonging to the group from the *Command Center Information Base*.
      * The group members are the recipients of group manipulate commands and group addressed commands to the group.
   3. If contents of group manipulate commands and group addressed commands to the group shall be encrypted, the MIHF chooses an MGK for the group.
   4. Update with the new MGK and the new membership information the *Managed Group Information Base.*
2. Define CompleteSubtree and SubgroupRange:
   1. Send MIHF IDs of the group member, the group management tree, and threshold for fragmentation to the CreateCompleteSubtreeFragments procedure, and receive CompleteSubtree and SubGroupRange.
   2. If the CompleteSubtree is not fragmented, SubgroupRange is removed.
3. (Optional) Define GroupKeyData:
   1. When MGK is not used, this process is skipped.
   2. Send the MGK and the CompleteSubtree to the MasterGroupKeyWrapping procedure, and receive GroupKeyData. The procedure accesses the *Command Center Base* to refer the group members’ device key.



1. — Flow diagram of the generation process of the GKB parameters
2. (Optional) Construct the UserSpecificData field.
3. Choose a DestinationIdentifier. A DestinationIdentifier is an MIHF Group ID, which represents an existing group. The group indicated by the DestinationIdentifier shall include all recipients who are manipulated by this command.
4. Generate an MIH\_Net\_Group\_Manipulate.request from the DestinationIdentifier, the TargetGroupIdentifier, the SubgroupRange (an option), the UserSpecificData (an option), the CompleteSubtree and the GroupKeyData (an option). Set the GroupKeyUpdateFlag if the MGK of the group designated by the TargetGroupIdentifier should be updated. Send it to the local MIHF.
5. Optionally, in case the CC obtains a Multicast Address to be used by the group (through any mean outside of this specification), it can choose to ask the MIHF to use it by including it in the MIH\_Net\_Group\_Manipulate.request.

Figure 25 shows a flow diagram summarizing the steps performed by the MIH User on a PoS, described in this Clause.

Fig38

1. — Summary of steps performed by PoS MIH User
   * + - 1. MIHF of a GMCS

Required components relevant to group manipulation and group commands are listed as follows:

* A signing key (of type SIGNING\_KEY as defined in Table F.25). The key is for creation of a signature at the Command center.
* A *Recipient Information Base* (of type RECIPIENT\_MIHF\_BASE as defined in Table F.25)containing the set of device keys to retrieve a group key from a GKB which is received from the local MIH User, the certificate used to verify digital signatures, and the information required to send commands to the group, i.e., the MIHF Group ID, the transport address used, the MGK, the sequence number and the SAID associated to the group.

It is assumed that the MIHF is able to obtain in some way a multicast address associated with a MIHF Group ID. The multicast address may be contained in the MIH\_Net\_Group\_Manipulate.request received from the MIH User. In this case, if the TargetGroupIdentifier in the received request is not registered in the *Recipient Information Base*, obtain the multicast address associated with the TargetGroupIdentifier and update the *Recipient Information Base* with the DestinationIdentifier and the associated multicast address. The MIHF of the Command center receives an MIH\_Net\_Group\_Manipulate.request, which is generated by the MIH User, the MIHF generates and sends an MIH\_Net\_Group\_Manipulate indication/request message to a multicast group. Note that this behavior depends on the ResponseFlag parameter. When “ResponseFlag=1”, the MIHF will generate MIH\_Net\_Group\_Manipulate request message. When “ResponseFlag=0”, the MIHF will generate MIH\_Net\_Group\_Manipulate indication message.

In the following we detail the steps performed to generate the message:

1. Generate a Source MIHF ID TLV using its own MIHF ID.
2. Generate a Destination MIHF ID TLV from the DestinationIdentifier in the received MIH\_Group\_Manipulate.request.
3. If GroupKeyUpdateFlag = 0 and GroupKeyData is contained in the received MIH\_Group\_Manipulate.request, it generates Sequence Number TLV from a current SequenceNumber with respect to the TargetIdentifier in the MIH\_Group\_Manipulate.request. Else Sequence Number TLV is not generated.
4. The MIHF generates a Multicast Address TLV. If the MIH\_Net\_Group\_Manipulate.request contains a MulticastAddress parameter, the parameter is contained in the Multicast Address TLV. Otherwise, a multicast address determined by the MIHF is contained in the Multicast Address TLV.
5. If the MIH\_Net\_Group\_Manipulate.request contains a SubgroupRange, it generates a SubgroupRange TLV from the SubgroupRange.
6. If the MIH\_Net\_Group\_Manipulate.request contains a UserSpecificData, it generates an Aux Data TLV from the UserSpecificData.
7. Generate a Complete Subtree TLV from the CompleteSubtree in the received MIH\_Net\_Group\_Manipulate.request.
8. If the MIH\_Net\_Group\_Manipulate.request contains a GroupKeyData, it generates a Group Key Data TLV from the GroupKeyData.
9. If GroupKeyUpdateFlag = 0, SAID TLV is generated using a security association ID with respect to the TargetIdentifier stored in the *Recipient Base*. Else decide new security association ID and generate SAID TLV from the security association ID.
10. If a security association ID with respect to the DestinationIdentifier is stored in own *Recipient Information Base*, it encrypts Service Specific TLVs of this group manipulation command as shown in 9.5.4.
11. Generate a Signature TLV as shown in 9.5.4 using the signing key of the MIHF.
12. If ResponseFlag=0, generate an MIH\_Net\_Group\_Manipulate indication using the preceding TLVs, else generate an MIH\_Net\_Group\_Manipulate request using the preceding TLVs.

Figure 26, shows a flow diagram summarizing the steps performed by the MIHF at a PoS, described in this Clause.

Fig39

We shall revise here.  
Hana

1. —Summary of steps performed by PoS MIHF
   * + 1. Procedures for group manipulation command recipients (GMCR)

Required components relevant to group manipulation and group commands are listed as follows:

* A *Recipient Information Base* (of type RECIPIENT\_MIHF\_BASE as defined in Table F.25)containing the set of device keys to retrieve a group key from a GKB which is received from the local MIH User, the certificate used to verify digital signatures, and the information required to send commands to the group, i.e., the MIHF Group ID, the transport address used, the MGK, the sequence number and the SAID associated to the group.

When a client MN receives a group manipulation command, i.e., an MIH\_Net\_Group\_Manipulate indication/request message, issued by a Command center, the MIHF of the MN processes the command.

1. The MIHF obtains a Source Identifier from the Source MIHF ID TLV.
2. The MIHF verifies the Signature TLV using a verification key in the certificate corresponding to the obtained SourceIdentifier stored in the *Recipient Information Base*. If the verification fails, the MIHF shall cancel the following steps and stop processing the command.
3. The MIHF checks the DestinationIdentifier in the Destination MIHF ID TLV. If the DestinationIdentifier does not match one of the following MIHF IDs, the MIHF shall cancel the following steps and stop processing the command: (i) An MIHF Group ID corresponding to a broadcast address, (ii) an MIHF Group ID which is registered with a multicast address in the the *Recipient Information Base*, or (iii) the MN's own MIHF ID.
4. The MIHF decrypts the payload if it is encrypted, i.e., if it is a Security TLV. The decryption key is derived from the MGK associated with the DestinationIdentifier in the the *Recipient Information Base*.
   1. In case an MN cannot decrypt the Security TLV, the message will be silently discarded.
5. If a SubgroupRange TLV exists in the indication, the MIHF obtains a SubgroupRange and checks whether its own Leaf Number is contained in the SubgroupRange or not. If it is not, the MIHF shall cancel the following steps and stop processing.
6. The MIHF obtains the TargetIdentifier in the Group Identifier TLV.

Fig40

1. —MGK generation process
2. The MIHF processes the Complete Subtree TLV as described in 9.4.2.2. If the MIHF succeeds to find a matching pair of Node Indices, go to the next step. Otherwise, go to Step r).
3. The MIHF obtains a GroupKeyUpdateFlag from the GroupKeyUpdateFlag.
4. If a MulticastAddress TLV exists in the indication, the MIHF obtains a MulticastAddress. Otherwise, the MIHF obtains a multicast address with respect to the TargetIdentifier from a server (Note that this operation is out of the scope of this specification).
5. If a GroupKeyData TLV exists in the indication, the MIHF obtains a GroupKeyData and derives a group key by processing the GroupKeyData using a Node Key corresponding with the Node Index as described in 9.4.2.2.
6. If a SAID TLV exists in the indication, the MIHF obtains a SAID.
7. If a Sequence Number TLV exists in the indication, the MIHF obtains a SequenceNumber. If the GroupKeyUpdateFlag is “1,” the MIHF resets the SequenceNumber to an initial value.
8. The MIHF checks whether the TargetIdentifier obtained in Step f) has already been registered or not in the *Recipient Information Base*. If it has been, go to Step n) [Stay]. Otherwise, go to Step p) [Join].
9. [Stay] The MIHF updates the multicast address, the group key and the SAID, and the SequenceNumber, with respect to the TargetIdentifier, in the *Recipient Information Base*.
10. The MIHF throws an MIH\_Net\_Group\_Manipulate.indication described in 7.4.32.2 to the MIH User. The GroupStatus field of the indication shall be “Unchanged successful” (5). The procedure of command processing terminates.
11. [Join] The MIHF starts listening to the multicast address associated with the TargetIdentifier. The MIHF saves in the the *Recipient Information Base*  the TargetIdentifier, the associated multicast address, the group key (Option), the SequenceNumber (Option), and the SAID (Option).
12. The MIHF issues an MIH\_Net\_Group\_Manipulate.indication described in 7.4.32.2. to the MIH User. The GroupStatus field must be “Join operation successful” (0). The procedure of command processing terminates.
13. The MIHF checks whether the TargetIdentifier obtained in Step f) has already been registered or not in the *Recipient Information Base*. If it has been, go to Step s) [Leave]. Otherwise, the MIHF terminates the procedure of command processing.
14. [Leave] The MIHF finds the multicast address recorded on the same row as the TargetIdentifier obtained in Step f) and the MIHF stops listening to it. The MIHF removes the row that has the TargetIdentifier.
15. The MIHF throws an MIH\_Net\_Group\_Manipulate.indication described in 7.4.32.2 to the MIH User. The GroupStatus field must be “Leave operation successful” (3). The procedure of command processing terminates.

Figure 28 summarizes the steps followed by the MIHF on the MN upon reception of an MIH\_Net\_Group\_Manipulation.indication.

Fig41

We shall revise here.  
Hana

1. —Summary of steps performed by the MN MIHF

Subclause 7.4.31 introduces a mechanism enabling the MN to trigger the Join/Leave operations controlled by the Command center. In order to do so, the MIH User located at the MN notifies the Command center of its desire to Join or Leave a group through the use of the MIH\_MN\_Group\_Manipulate primitive. The Command center, upon receiving the associated request message, performs the same process as defined in this Clause, for the use of the MIH\_Net\_Group\_Manipulate, although in this case, the group to be manipulated is provided by the MN. The resulting GKB parameters are returned to the MN in the MIH\_MN\_Group\_Manipulate response message.

|  |  |  |
| --- | --- | --- |
| MANAGED\_GROUP\_INFO | SEQUENCE(  MIHF\_ID,  CHOICE(MGK,NULL),  GROUP\_MEMBERS,  ) | The information base stores MIHF Group ID and its group members. |
| TRANSPORT\_ADDR\_BASE | LIST(TRANSPORT\_ADDR) | The information base stores transport addresses which can be used to issue group manipulation commands and group addressed commands. |
| MANAGED\_GROUP\_INFO\_BASE | LIST(MANAGED\_GROUP\_INFO) | Information Base containing the group information to manage the groups. |
| GRP\_MGMT\_TREE\_NODE | SEQUENCE(  NODE\_INDEX,  CHOICE(NODE\_KEY,NULL)) | This is the base data type of GRP\_MGT\_TREE\_PATH |
| GRP\_MGT\_TREE\_PATH | SEQUENCE(  LIST(GRP\_MGMT\_TREE\_NODE), MIHF\_ID  ) | The data type of a part of group management tree for a recipient. GRP\_MGT\_TREE\_PATH is equivalent to device keys when each ROUP\_MGMT\_TREE\_NODE contains a node key. |
| GROUP\_MEMBERSHIP\_BASE | SEQUENCE(  MIHF\_ID,  TRANSPORT\_ADDR,  CHOICE(MGK,NULL),  CHOICE(SEQUENCE\_NUMBER,NULL),  CHOICE(SAID,NULL)  ) | The information base stores group membership information to manage a belonged group. |
| GRP\_MGT\_TREE\_INFO\_BASE | LIST(GRP\_MGT\_TREE\_PATH) | The information base for a key management tree which stores all parameters to manage group membership using GKB. |
| RECIPIENT\_MIHF\_BASE | SEQUENCE(  CHOICE(DEVICE\_KEYS, NULL),  CERTIFICATE,  LIST(GROUP\_MEMBERSHIP\_BASE)  ) | The information base stores an individual device keys to decrypt GKB, Certificate to verify digital signatures, and Group\_Membership\_Base to manage belonged groups. |