|  |  |  |
| --- | --- | --- |
| Project | **IEEE 802.21d**  **<**[**http://www.ieee802.org/21/**](http://www.ieee802.org/21/)**>** | |
| Title | Suggested Remedy for IEEE 802.21d Lb7b comments #58 | |
| DCN | 21-14-0075-02-MuGM | |
| Date Submitted | **April, 19th, 2014** | |
| Source(s) | Yoshikazu Hanatani, Toru Kambayashi (Toshiba), Subir Das (ACS) |  |
| 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#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> | |

**Add texts to explain the purpose and functionality of SubtreeFlag**

**~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

In case when a complete subtree is only present in a GKB, the GKB is used for specifying the group members of a particular group instead of a group key distribution. Following methods are used to identify the group members appropriately.

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 SubtreeFlag. If the flag is 0, Method 1 is used. If the flag is 1, Method 2 is used. The SubtreeFlag thus helps the recipient to correctly interpret the complete subtree part of a GKB.

**Amend the primitives for the group manipulation commands as follows**

**~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

1. * + 1. MIH\_MN\_Group\_Manipulate.response
          1. Function

This primitive is generated by an MIH User in a PoS to acknowledge result of an MIH\_MN\_Group\_Manipulate request from an MN.

* + - * 1. Semantics of service primitive

MIH\_MN\_Group\_Manipulate.response (

DestinationIdentifier,

TargetIdentifier,

MulticastAddress,

SubgroupRange,

UserSpecificData,

CompleteSubtree,

SubtreeFlag,

GroupKeyData,

GroupStatus

)

Parameters:

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
| DestinationIdentifier | MIHF\_ID | Specifies the MIHF ID of the destination of the primitive. |
| TargetIdentifier | MIHF\_ID | The target MIHF group identifier for the group operation. |
| MulticastAddress | TRANSPORT\_ADDR | (Optional) Multicast address corresponding with the target group identifier. |
| SubgroupRange | SUBGROUP\_RANGE | (Optional) Subgroup to process the command.a |
| UserSpecificDatab | OCTET\_STRING | (Optional) Auxiliary data. |
| SubtreeFlag | SUBTREE\_FLAG | (Optional) Flag to interpret the complete subtree data |
| CompleteSubtree | COMPLETE\_SUBTREE | (Optional) Complete Subtree data. |
| GroupKeyData | GROUP\_KEY\_DATA | (Optional )Encrypted group key. |
| GroupStatus | GROUP\_STATUS | Status of the group operation. |

a SubgroupRange parameter shall be present for a fragmented GKB.

b The UserSpecificData parameter can be used to convey additional information such as version information of the GKB used or additional credentials.

* + - * 1. When generated

An MIH User at the PoS generates this primitive after receipt and processing of MIH\_MN\_Group\_Manipulate request. This primitive returns the status of the action asked in the request. Optionally, it may respond with the security mechanisms required by the group.

* + - * 1. Effect on receipt

MIH\_MN\_Group\_Manipulate response message is sent back to the requester.

* + 1. 1. MIH\_Net\_Group\_Manipulate.request
          1. Function

This primitive is generated by the MIH User of a PoS to manipulate group membership of one or more MN(s) or other PoS(es).

* + - * 1. Semantics of service primitive

MIH\_Net\_Group\_Manipulate.request (

DestinationIdentifier,

ResponseFlag,

GroupKeyUpdateFlag,

TargetIdentifier,

MulticastAddress,

SubgroupRange,

UserSpecificData,

SubtreeFlag,

CompleteSubtree,

GroupKeyData

)

Parameters:

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
| DestinationIdentifier | MIHF\_ID | Specifies group MIHF-ID of the remote MIHF peers. DestinationIdentifier may be different from TargetIdentifier. |
| ResponseFlaga | RESPONSE\_FLAG | (Optional) Flag that represents whether or not a response is needed. |
| GroupKeyUpdateFlag | GROUP\_KEY\_UPDATE\_FLAG | Flag that represents whether or not a group key in GroupKeyData is updated. |
| TargetIdentifier | MIHF\_ID | The target MIHF group identifier for the group operation. |
| MulticastAddress | TRANSPORT\_ADDR | (Optional) Multicast address corresponding with the target group identifier. |
| SubgroupRange | SUBGROUP\_RANGE | (Optional) Subgroup to process the command |
| UserSpecificData | OCTET\_STRING | (Optional) Auxiliary data. |
| SubtreeFlag | SUBTREE\_FLAG | (Optional) Flag to interpret the complete subtree data. |
| CompleteSubtree | COMPLETE\_SUBTREE | Complete Subtree data. |
| GroupKeyData | GROUP\_KEY\_DATA | (Optional) Encrypted group key. |

a In case the ResponseFlag parameter is not present, the MIHF should always generate a request message, and otherwise the MIHF generates either a request or an indication message, based on the ResponseFlag parameter.

* + - * 1. When generated

The MIH user generates this primitive to create, delete or modify group membership.

* + - * 1. Effect on receipt

Upon receipt of this primitive, MIHF on the PoS sends the corresponding MIH\_Net\_Group\_Manipulate indication message or MIH\_Net\_Group\_Manipulate request message to the MN(s) or other PoS(es). The ResponseFlag TLV indicates which message shall be sent.

**Amend the messages for the group manipulation commands as follows**

**~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

1. * + 1. MIH\_MN\_Group\_Manipulate response

The corresponding MIH primitive of this message is defined in 7.4.31.3.

This message is used by the MIHF to supply the group status of MIH node(s) identified by the Source Identifier.

|  |
| --- |
| MIH Header Fields (SID=1, Opcode=2, AID=11 ) |
| **Source Identifier** = sending MIHF ID  (Source MIHF ID TLV) |
| **Destination Identifier** = receiving MIHF ID  (Destination MIHF ID TLV) |
| TargetIdentifier  (Group Identifier TLV) |
| SequenceNumber (conditional)ª  (Sequence Number TLV) |
| MulticastAddress (Optional)  (Multicast Address TLV) |
| SubgroupRange (Optional)  (Subgroup\_Range TLV) |
| UserSpecificData (Optional)  (Aux Data TLV) |
| SubtreeFlag (Optional)  (Subtreeflag TLV) |
| CompleteSubtree (Optional)  (Complete Subtree TLV) |
| GroupKeyData (Optional)  (Group Key Data TLV) |
| GroupStatus  (Group Status TLV) |
| SecurityAssociationID (Optional)  (SAID TLV) |

ª This parameter is only used in the case CCM encryption method is used and the group key is not updated.

* + - 1. MIH\_Net\_Group\_Manipulate request

The corresponding MIH primitive of this message is defined in 7.4.32.1.

This message is used by the MIHF to manipulate group membership of MIH node(s) identified by the Destination Identifier.

|  |
| --- |
| MIH Header Fields (SID=1, Opcode=1, AID=12 ) |
| **Source Identifier** = sending MIHF ID  (Source MIHF ID TLV) |
| **Destination Identifier** = receiving MIHF ID  (Destination MIHF ID TLV) |
| GroupKeyUpdateFlag  (Group Key Update Flag TLV) |
| TargetIdentifier  (Group Identifier TLV) |
| SequenceNumber (Optional)a  (Sequence Number TLV) |
| MulticastAddress (Optional)  (Multicast Address TLV) |
| SubgroupRange (Optional)  (Subgroup Range TLV) |
| UserSpecificData (Optional)  (Aux Data TLV) |
| SubtreeFlag (Optional)  (Subtreeflag TLV) |
| CompleteSubtree  (Complete Subtree TLV) |
| GroupKeyData (Optional)  (Group Key Data TLV) |
| SecurityAssociationID (Optional)  (SAID TLV) |

a This parameter is only used in the case CCM encryption method is used and the group key is not updated.

* + - 1. MIH\_Net\_Group\_Manipulate indication

The corresponding MIH primitive of this message is defined in 7.4.32.2.

This message is used by the MIHF to manipulate group membership of MIH node(s) identified by the Destination Identifier.

|  |
| --- |
| MIH Header Fields (SID=1, Opcode=3, AID=12 ) |
| **Source Identifier** = sending MIHF ID  (Source MIHF ID TLV) |
| **Destination Identifier** = receiving MIHF ID  (Destination MIHF ID TLV) |
| TargetIdentifier  (Group Identifier TLV) |
| GroupKeyUpdateFlag  (Group Key Update Flag TLV) |
| SequenceNumber (Optional)  (Sequence Number TLV) |
| MulticastAddress (Optional)  (Multicast Address TLV) |
| SubgroupRange (Optional)  (Subgroup Range TLV) |
| UserSpecificData (Optional)  (Aux Data TLV) |
| SubtreeFlag (Optional)  (Subtreeflag TLV) |
| CompleteSubtree  (Complete Subtree TLV) |
| GroupKeyData (Optional)  (Group Key Data TLV) |
| SecurityAssociationID (Optional)  (SAID TLV) |

**Define new data type:**

**~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

Table F.24—Data type for security

|  |  |  |
| --- | --- | --- |
| Data type name | Derived from | Definition |
| SUBTREE\_FLAG | BOOLEAN | This indicates whether the leaf nodes of the complete subtree belong to the group or not.  0 (FALSE): Leaf nodes belong to the group  1 (TRUE): Leaf nodes that do not belong to the group. |

**Define new TLV:**

**~~~~~~~~~~~~~~~~~~**

Table L.2 —Type values for TLV encoding

|  |  |  |
| --- | --- | --- |
| TLV type name | TLV type value | Data Type |
| SubtreeFlag | 97 | SUBTREE\_FLAG |

**Add steps to process SubtreeFlag**

**~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

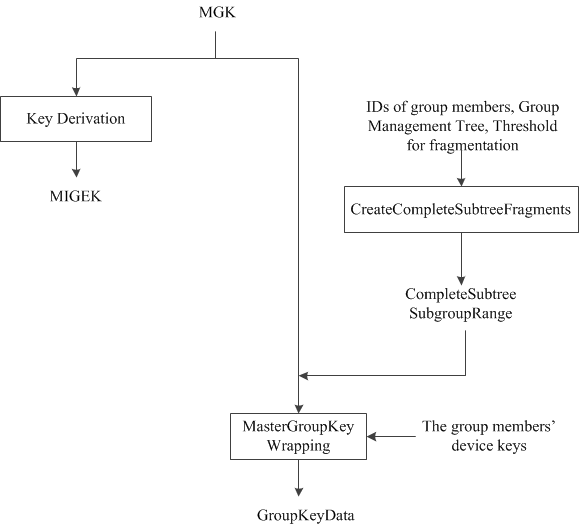
* + - * 1. MIH user of a GMCS

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 comprised of CreateCompleteSubtreeFragments (see 9.4.2.3), 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 all the pairs of an MIHF ID and a corresponding leaf number, and all the pairs of a Node Index and a corresponding Node Key.
* A *Managed Group Information Base* (of type MANAGED\_GROUP\_INFO\_BASE as defined in Table F.25). This information base stores the information about groups which are managed by the GMCS. It stores tuples of an MIHF Group ID, the MIHF IDs of the group members, the MGK (an optional) assigned to the group and the transport addresses for multicast (an optional) assigned to the group.

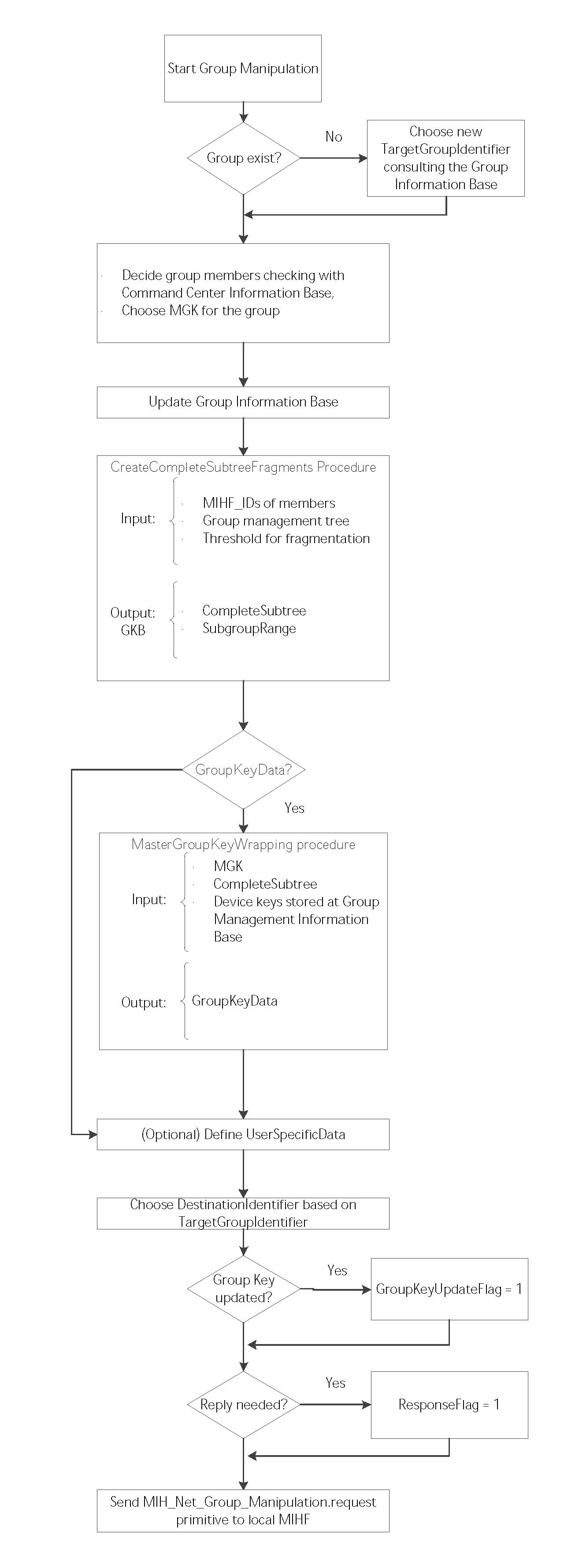
A Flow diagram of the generation process of the GKB parameters is given in Figure 37. The MIH User generates MIH\_Net\_Group\_Manipulate.request described in 7.4.32.1 as follows:

1. Choose an MIHF Group ID and group members to manipulate.
2. If necessary, update the membership information, the MGK and the transport address in the *Managed Group Information Base.*
3. Define TargetGroupIdentifier:
   1. Set the MIHF Group ID chosen in step a) to TargetGroupIdentifier.
4. Define CompleteSubtree and SubgroupRange:
   1. Set SubtreeFlag.
   2. If SubtreeFlag = 0, the MIH User sends MIHF IDs of the group members, all Node Indices, and a threshold for fragmentation to the CreateCompleteSubtreeFragments procedure, and receive CompleteSubtree and SubGroupRange.
   3. If SubtreeFlag = 1, the MIH User sends MIHF IDs of the non-group members, all Node Indices, and a threshold for fragmentation to CreateCompleteSubtreeFragments procedure, and receive CompleteSubtree and SugGroupRange.
   4. If the CompleteSubtree is not fragmented, SubgroupRange is removed.
5. (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 *Group Management Tree Information Base* to refer all the pairs of a Node Index and a corresponding Node Key.

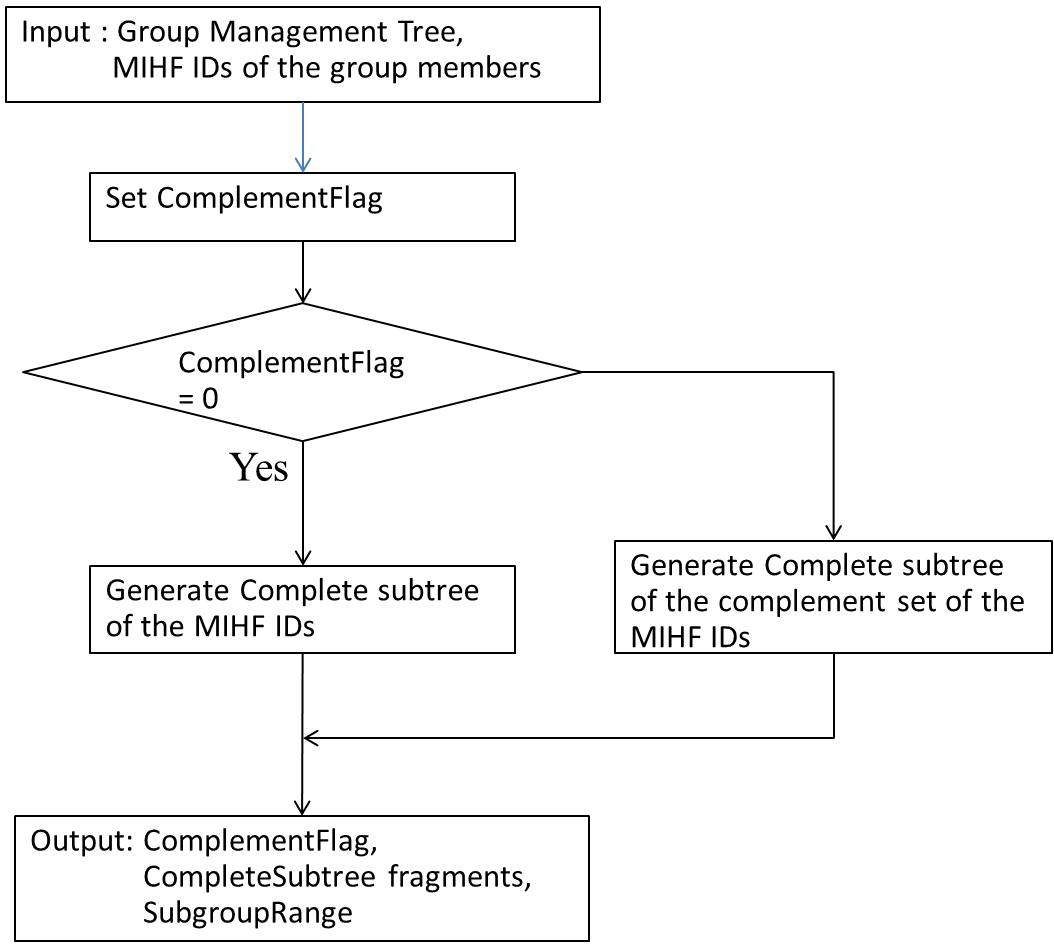


1. — Flow diagram of the generation process of the GKB parameters
2. (Optional) Construct the UserSpecificData field.
3. Choose a DestinationIdentifier. A DestinationIdentifier is an MIHF Group ID, which represents an existing group. The group indicated by the DestinationIdentifier shall include all 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 MIH User of GMCS 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 38 shows a flow diagram summarizing the steps performed by the MIH User on a PoS, described in this Clause. Figure 39 shows a flow diagram summarizing the steps to define CompleteSubtree and SubgroupRange which are corresponding with CreateCompleteSubtreeFragments procedure in Figure 38..



1. — Summary of steps performed by PoS MIH User



1. — Flow diagram of CreateCompleteSubtreeFragments Procedure
2. * + - 1. MIHF of a GMCS

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

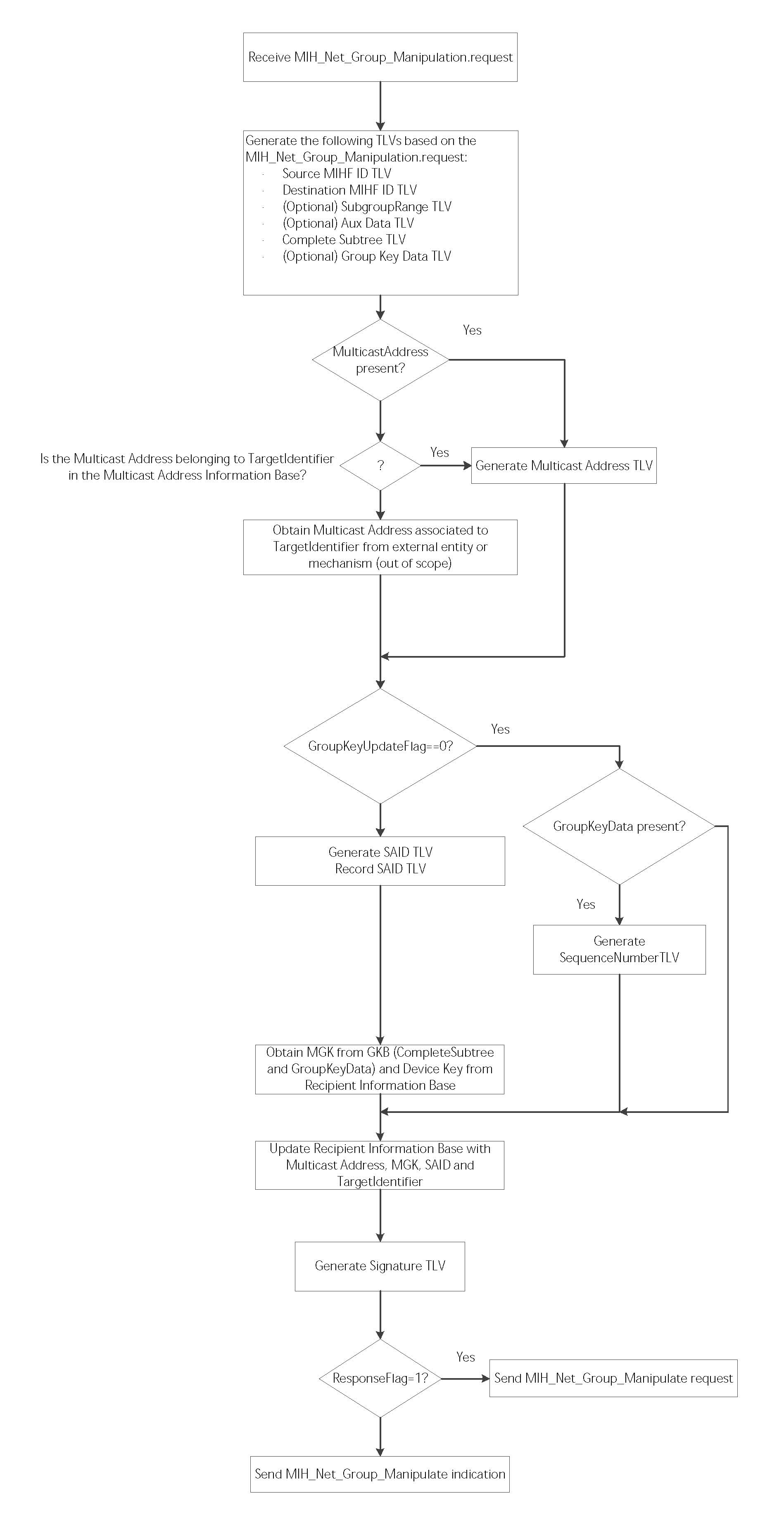
* A signing key (of type SIGNING\_KEY as defined in Table F.25). The key is for creation of a signature at the GMCS.
* A *Recipient Information Base* (of type RECIPIENT\_MIHF\_BASE as defined in Table F.25)stores the pairs of a Node Index and a corresponding Node Key (i.e., device keys) to retrieve a group key from a GKB, 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. Else if the MIH\_Net\_Group\_Manipulate.request does not contain a MulticastAddress parameter, the MIHF decides a multicast address parameter.
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 SubtreeFlag TLV from the SubtreeFlag in the received MIH\_Net\_Group\_Manipulate.request.
8. Generate a Complete Subtree TLV from the CompleteSubtree in the received MIH\_Net\_Group\_Manipulate.request.
9. If the MIH\_Net\_Group\_Manipulate.request contains a GroupKeyData, it generates a Group Key Data TLV from the GroupKeyData.
10. If GroupKeyUpdateFlag = 0, SAID TLV is generated using a security association ID with respect to the TargetIdentifier stored in the *Recipient Information Base*. Else decide new security association ID and generate SAID TLV from the security association ID.
11. If a security association ID with respect to the DestinationIdentifier is stored in its own *Recipient Information Base*, it encrypts Service Specific TLVs of this group manipulation command as shown in 9.5.4.
12. Generate a Signature TLV as shown in 9.5.4 using the signing key of the MIHF.
13. 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 39, shows a flow diagram summarizing the steps performed by the MIHF at a PoS, described in this Clause.



Please remove this step

Please remove this step

Complement Flag TLV

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 pairs of a Node Index and a corresponding Node Key (i.e., device keys) to retrieve an MGK from a GKB, 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 GMCS, the MIHF of the GMCR 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 *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 *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 a TargetIdentifier in the Target Identifier TLV, a SubtreeFlag in the SubtreeFlag TLV, and a CompleteSubtree in the Complete Subtree TLV.

Fig40

1. —MGK generation process
2. The MIHF processes the Complete Subtree 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 i).
3. If SubtreeFlag = 0, go to Step j). Otherwise, go to Step t).
4. If SubtreeFlag = 0, go to Step t). Otherwise, go to Step j).
5. The MIHF obtains a GroupKeyUpdateFlag from the GroupKeyUpdateFlag TLV.
6. 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).
7. 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.
8. If a SAID TLV exists in the indication, the MIHF obtains a SAID.
9. 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.
10. 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 p) [Stay]. Otherwise, go to Step r) [Join].
11. [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*.
12. 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.
13. [Join] The MIHF starts listening to the multicast address associated with the TargetIdentifier. The MIHF saves in the *Recipient Information Base* the TargetIdentifier, the associated multicast address, the group key (Option), the SequenceNumber (Option), and the SAID (Option).
14. 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.
15. The MIHF checks whether the TargetIdentifier has already been registered or not in the *Recipient Information Base*. If it has been, go to Step u) [Leave]. Otherwise, the MIHF terminates the procedure of command processing.
16. [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.
17. 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 42 summarizes the steps followed by the MIHF on the MN upon reception of an MIH\_Net\_Group\_Manipulation.indication.



No

Yes

No

Yes

Yes

SubtreeFlag = 0

SubtreeFlag = 0

Matching pair of GKB indices found?

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.