|  |  |
| --- | --- |
| Project | **IEEE 802.21 MIHS****<**[**http://www.ieee802.org/21/**](http://www.ieee802.org/21/)**>** |
| Title | **Proposed remedy for SB Comment i-115** |
| DCN | **21-14-0167-01-MuGM** |
| Date Submitted | **November 4, 2014** |
| Source(s) | Yoshikazu Hanatani and Yoshihiro Ohba (Toshiba) |  |
| Re: | IEEE 802.21d Sponsor Ballot comment resolution |
| Abstract | This document describes a proposed remedy for SB comment i-115 about protection method for MIH fragments. |
| Purpose | For Sponsor Ballot 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

# Comment i-115

This is comment on overall GKB fragmentation: In base specification, we specified that "No retransmission by the MIH protocol (defined in 8.2) is performed for any single fragment of a multifragment message (802.21-2008, pp 154)". Now assume that MIH uses unreliable transport and sets the ACK bit and also both GKB and MIH both fragments the packet. In this scenario, if the source MIHF does not receive the ACK response, it will start to retransmit. Now if the GKB fragment length is more than the MIH fragment length, and GKB expects the source MIHF to retransmit, the retransmission will never happen according to the base specification. What I was thinking is that we should add some text to avoid this situation so that the implementers will know how to choose the fragment length. For example, if the fragmentation happens both at MIH and GKB layers, GKB fragment length should be <= MIH fragment length.

# Related Problem on Comment i-115

Descriptions of MIH PDU protected through Group key generated MIHSA and digital signature are missed, when MIH PDU fragmentation is occurred.

# Proposed resolution

* Revise Figure 30 and 31.



MIH Service Specific TLVs or a fragment

MIH Service Specific TLVs or a fragment



MIH Service Specific TLVs or a fragment

* Simplify 9.5.3.1.2 and Figure 40
	+ Change 9.5.3.1.2 to explain generating method of MIH\_Net\_Group\_Manipulate message’s the MIH Service Specific TLVs.
		- Add a step for generating the Group Identifier TLV.
		- Remove the steps which do not generate the Service Specific TLVs.
		- Remove the redundant step.
		- Change the condition for generating the Sequence Number TLV. (To resolve Comment i-37)
	+ Note: This remedy is made from P802.21d/D6.0.

 **9.5.3.1.2 MIHF of a PoS with Group Manager**

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 PoS with group manager.
* A *Group Membership Information Base* (of type GROUP\_MEMBERSHIP\_BASE as defined in Table F.25)stores the information required to send commands to the group, i.e., the MIHF Group ID and the transport address used. If the service specific TLVs carried in group addressed commands and group manipulation commands addressed to the group are encrypted, the *Group Membership Information Base* also stores the MGK, the sequence number and the SAID associated with the group.

Allocation of a transport address to an MIHF Group ID is implementation specific and outside the scope of this standard. The transport address may be contained in the MIH\_Net\_Group\_Manipulate.request received from the MIH User. When the MIHF receives an MIH\_Net\_Group\_Manipulate.request, generated by the MIH User, the MIHF generates and sends an MIH\_Net\_Group\_Manipulate indication/request message to an MIHF Group or an MIHF. 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 MIH Service Specific TLVs of the message:

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

1. Generate a Group Identifier TLV from the TargetIdentifier in the received MIH\_Net\_Group\_Manipulate.request.
2. If the MIH\_Net\_Group\_Manipulate.request contains a SubgroupRange, it generates a SubgroupRange TLV from the SubgroupRange.
3. If the MIH\_Net\_Group\_Manipulate.request contains a UserSpecificData, it generates an Aux Data TLV from the UserSpecificData.
4. Generate a Complete Subtree TLV from the CompleteSubtree in the received MIH\_Net\_Group\_Manipulate.request.
5. If the MIH\_Net\_Group\_Manipulate.request contains a GroupKeyData, it generates a Group Key Data TLV from the GroupKeyData.
6. If MIH\_Net\_Group\_Manipulate.request contains a ComplementSubtreeFlag, it generates Complement Subtree Flag TLV from the ComplementSubtreeFlag.
7. The MIHF generate a Sequence Number TLV.
8. The MIHF generates optionally a Transport Address TLV. If the MIH\_Net\_Group\_Manipulate.request contains a TransportAddress parameter, the parameter is contained in the Transport Address TLV. Else if the MIH\_Net\_Group\_Manipulate.request does not contain a TransportAddress parameter, the MIHF decides a transport address parameter (which is implementation specific and outside of the scope of this specification).
9. If GroupKeyData is accompanied, generate an SAID Notification TLV. If GroupKeyUpdateFlag=0, the TLV contains the security association identifier associated with the GroupKeyData. Otherwise, the TLV contains a newly allocated security association ID for the GroupKeyData. The security association identifier obtained through the SAID Notification TLV is stored. This identifier is used in SAID TLVs, which are carried in subsequent MIH messages encrypted by the group key corresponding to the GroupKeyData.
10. Update the *Group Membership Information Base* with TargetGroupIdentifier, the transport address parameter, the sequence number and the SAID. If a MasterGroupKey is contained in the MIH\_Net\_Group\_Manipulate.request, also update the *Group Membership Information Base* with the MasterGroupKey as the MGK.

Figure 40, shows a flow diagram summarizing the steps performed by the MIHF at a PoS, described in this Clause.



Add “Group Identifier TLV,”

Add “Sequence Number TLV”

Remove

Please move to here.

Remove
(These steps are shown in 9.6.2.)

Output the Service Specific TLVs

Remove

* Simplify 9.5.3.2 and Figure 42
	+ Change 9.5.3.2 to explain generating method of MIH\_Net\_Group\_Manipulate message’s the MIH Service Specific TLVs.
		- Remove the steps which do not generate the Service Specific TLVs.
		- Remove the redundant step.
		- Merge the step on the Complete Subtree TLV and the step on the Group Key Data TLV.
		- Change the condition for SAID since a SAID shall be present in case the Group Key Data TLV presents.
		- Move the step on the Transport Address TLV for simple description.
		- Correct Step c) and Step d) (To resolve Comment i-68.)
		- Remove Group Key Update Flag TLV from the group manipulation messages since it is redundant.
			* Related revisions are shown in 8.6.1.25 and 8.6.1.26. See below.
		- Revise Figure 42.
	+ Note: This remedy is made from P802.21d/D6.0.

 **9.5.3.2 Receiving procedures for group manipulation commands**

The recipient of a group manipulation message is either an MN or a PoS that receives and understands group manipulation commands.

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

* A *Group* *Recipient Information Base* (of type GROUP\_RECIPIENT\_INFO\_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 and the certificate used to verify digital signatures.
* A *Group Membership Information Base* (of type GROUP\_MEMBERSHIP\_BASE as defined in Table F.25) stores the information required to receive commands to the group, i.e., the MIHF Group ID and the transport address used. If the service specific TLVs carried in group addressed commands and group manipulation commands addressed to the group are encrypted, the Group Membership Information Base also stores the MGK, the sequence number and the SAID associated with the group.

When an MIHF receives an MIH\_Net\_Group\_Manipulate indication/request or MIIH\_MN\_Group\_Manipulate response message, it processes the message, after reassembling fragments if any, as follows:

1. If a SubgroupRange TLV exists in the message, 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.
2. The MIHF processes the Complete Subtree in the Complete Subtree TLV and a GroupKeyData in the Group Key Data TLV as described in 9.5.2.2. If the MIHF succeeds to find a matching pair of Node Indices, go to Step c). Otherwise, go to Step d).
3. If ComplementSubtreeFrag in the Complement Subtree Flag TLV is “0”, go to Step e), else go to Step m).
4. If ComplementSubtreeFrag in the Complement Subtree Flag TLV is “0”, go to Step m), else go to Step g).
5. If a group key MGK is derived in Step b), the MIHF obtains a SAID in the SAID notification TLV.
6. If a Transport Address TLV exists in the message, the MIHF obtains a TransportAddress. Otherwise, the MIHF obtains a transport address with respect to the TargetIdentifier from a server (Note that this operation is out of the scope of this specification).
7. If a Sequence Number TLV exists in the message, the MIHF obtains a SequenceNumber.
8. The MIHF checks whether a TargetIdentifier in the Group Identifier TLV has already been registered or not in the *Group* *Recipient Information Base*. If it has been, go to Step i) [Stay]. Otherwise, go to Step k) [Join].
9. [Stay] The MIHF updates the transport address, the group key and the SAID, and the SequenceNumber, with respect to the TargetIdentifier, in the *Group Membership 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 transport address associated with the TargetIdentifier. The MIHF saves in the *Group Membership Information Base* the TargetIdentifier, the associated transport 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 a TargetIdentifier in the Group Identifier TLV has already been registered or not in the *Group Membership Information Base*. If it has been, go to Step n) [Leave]. Otherwise, the MIHF terminates the procedure of command processing.
14. [Leave] The MIHF finds the transport address recorded on the same row as the TargetIdentifier 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 42 summarizes the steps followed by the MIHF on the MN/PoS upon reception of an MIH\_Net\_Group\_Manipulation indication or request message, after reassembling fragments if any.

****

1. —Summary of steps performed by the recipient MIHF

 Subclause 7.4.31 introduces a mechanism enabling the recipient to trigger the Join/Leave operations controlled by the PoS with group manager. In order to do so, the MIH User located at the recipient notifies the PoS with group manager of its desire to Join or Leave a group through the use of the MIH\_MN\_Group\_Manipulate primitive. The MIHF of the PoS with group manager, 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 recipient. The resulting GKB parameters are returned to the recipient in the MIH\_MN\_Group\_Manipulate response message.

* Revise 9.6.2
	+ Change “MIH Service Specific TLVs” to “MIH Service Specific TLVs or a fragment.”
	+ **Changes in this section are incrementally made against the proposed text in 21-14-0152-03-MuGM**

**9.6.2 Multicast message protection**

 a) The MIHF generates a Source MIHF ID TLV based on its own MIHF ID.

 b) The MIHF generates a Destination MIHF ID TLV based on the DestinationIdentifier in the received request.

 c) The MIHF generates MIH Service Specific TLVs or a fragment based on the received request primitive.

 d) Consulting with the Multicast Address Information Base, the MIHF finds the multicast address associated with the DestinationIdentifer in the received request.

e) The MIHF generates an MIH request or indication message.

 1) The MIH Service Specific TLVs or fragment may be encrypted with an MIGSK associated to the

 DestinationIdentifier to make a Security TLV if necessary in the scheme described in 8.4.2.

 2) A signature TLV may be generated as shown in 9.6.3.1 using the signing key of the MIHF.

f) The MIHF sends the message to the multicast address found in Step d).

 The MIH Service Specific TLVs or fragment may be encrypted to make a Security TLV if necessary in the scheme described in 8.4.2.

 When an MIHF of a recipient receives the message, the following steps are taken:

 a) The Destination Identifier is retrieved from the Destination MIHF ID TLV. The MIHF checks if the Destination Identifier is registered in the Group Information Base or not. If it is not, the message is not for the recipient. Thus, it cancels the following steps and stops processing.

 b) The Source Identifier is retrieved from the Source MIHF ID TLV.

c) If the Signature TLV is attached, the MIHF verifies SIGNATURE\_DATA in the Signature TLV using the verification key corresponding with the CERT\_SERIAL\_NUMBER in the Signature TLV as shown in 9.6.3.2.

 d) If a Security TLV is contained, the MIHF decrypts the Security TLV with the MIGSK associated with the preceding Destination Identifier and the SAID in the Security TLV that is available in the Group Information Base. If the decryption fails, it cancels the following steps and abort. If the decryption succeeds, MIH Service Specific TLVs or a fragment is obtained.

 e) If MIH service specific TLVs is obtained or a fragment is obtained and reassembling fragments succeeds, the MIHF issues an indication primitive to its local MIH User.

* Revise 9.6.3.1
	+ Change “MIH Service Specific TLVs” to “MIH Service Specific TLVs or a fragment.”
	+ **Changes in this section are incrementally made against the proposed text in 21-14-0152-03-MuGM**

**9.6.3.1 Multicast Message Signatures**

Multicast Messages are signed with the message source using a private key of the message source. Integrity and proof of origin of a multicast message is verified by verifying the message signature with the public key of a message source. The message content is signed using elliptical curve cryptography.

In case the MIH PDU is protected through GKB-generated MIH SA with a signature as specified in 8.4.2.3, the MIHF of PoS generates a Signature TLV consisting of a CERT\_SERIAL\_NUMBER and a SIGNATURE\_DATA. The SIGNATURE\_DATA is created by signing an MIH\_Group\_Manipulate command or a group addressed command using a signing key corresponding with a verification key specified by CERT\_SERIAL\_NUMBER. Figure 44 illustrates the data protection procedure with confidentiality.

Service Specific TLVs or a fragment



1. —Signing (with confidentiality)

The MIHF encrypts the MIH Service Specific TLVs or fragment with MIGSK, and generates the Security TLV. The MIHF selects a certification serial number and generates CERT\_SERIAL\_NUMBER from the certification serial number. Then, the MIHF computes the SIGNATURE\_DATA of the Signature TLV from the MIH Header, the Source MIHF Identifier TLV, the Destination MIHF Identifier TLV, the SAID TLV, the Security TLV, and the Type, Length and Value fields of the Signature TLV excluding the SIGNATURE\_DATA.

Note: If the MIH PDU is protected through a GKB-generated MIH SA, the signature TLV shall not include the SEQUENCE\_NUMBER.

In case the MIH PDU is not protected through a GKB-generated MIH SA and protected with a signature only as specified in 8.4.2.4, the MIHF of PoS generates a Signature TLV consisting of a SIGNATURE\_DATA, a CERT\_SERIAL\_NUMBER and a SEQUENCE\_NUMBER. Figure 45 illustrates the data protection procedure without confidentiality.



Service Specific TLVs or a fragment

1. —Signing (without confidentiality)

The MIHF computes the SIGNATURE\_DATA of the Signature TLV from the MIH Header, the Source MIHF Identifier TLV, the Destination MIHF Identifier TLV, the MIH Service Specific TLVs or fragment, and the Type, Length and Value fields of the Signature TLV excluding the SIGNATURE\_DATA.

Note: If the MIH PDU is not protected through a GKB-generated MIH SA, the signature TLV shall include the SEQUENCE\_NUMBER.

On receipt of signed multicast message there is an optional response indicating the validity of signature.

Message source requests credentials for key updates. Message source provides updates of credentials to destination devices (with overlap period).

**9.6.3.2 Signature Verification**

The signature is verified using the message source signature verification key. The message source will identify which key is to be used for the multicast message so that verification will utilize the correct key for signature verification.

In case the MIH PDU received contains a Signature TLV and is protected through a GKB-generated MIH SA, then the MIHF of recipient retrieves a CERT\_SERIAL\_NUMBER and a SIGNATURE\_DATA from the Signature TLV. Then, the MIHF verifies the SIGNATURE\_DATA using a verification key specified by the CERT\_SERIAL\_NUMBER. If the Signature TLV includes a SEQUENCE\_NUMBER, the MIH PDU shall be dropped since it is a wrong form. Figure 46 illustrates the data protection procedure with confidentiality.



Service Specific TLVs or a fragment

1. —Signature verification (with confidentiality)

In the case the MIH PDU contains a Signature TLV and is not protected through a GKB-generated MIH SA, then the MIHF of recipient retrieves CERT\_SERIAL\_NUMBER, SIGNATURE\_DATA and SEQUENCE\_NUMBER from the Signature TLV. Then, the MIHF verifies the SIGNATURE\_DATA using a verification key specified by the CERT\_SERIAL\_NUMBER, and the SEQUENCE\_NUMBER. The received SEQUENCE\_NUMBER is considered valid if and only if the SEQUENCE\_NUMBER is greater than the last valid incoming sequence number maintained for the sender. Figure 47 illustrates the data protection procedure without confidentiality.



Service Specific TLVs or a fragment

1. —Signature verification (without confidentiality)
* Revise 8.6.1.25 MIH\_Net\_Group\_Manipulate request
	+ **Remove GroupKeyUpdateFlag from the message (not primitive) for simplicity.**

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) |
| TransportAddress (Optional)(Transport Address TLV) |
| SubgroupRange (Optional)(Subgroup Range TLV) |
| UserSpecificData (Optional)(Aux Data TLV) |
| CompleteSubtree(Complete Subtree TLV) |
| ComplementSubtreeFlag (Optional)b(Complement Subtree Flag TLV) |
| GroupKeyData (Optional)(Group Key Data TLV) |
| SecurityAssociationID (Optional) (SAID Notification TLV) |

a This parameter is carried for CCM counter synchronization purpose when GroupKeyData is carried.

b If ComplementSubtreeFlag is not present, it means ComplementSubtreeFl

* Revise 8.6.1.26 MIH\_Net\_Group\_Manipulate indication
	+ **Remove GroupKeyUpdateFlag from the message (not primitive) for simplicity.**

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)a(Sequence Number TLV) |
| TransportAddress (Optional)(Transport Address TLV) |
| SubgroupRange (Optional)(Subgroup Range TLV) |
| UserSpecificData (Optional)(Aux Data TLV) |
| CompleteSubtree(Complete Subtree TLV) |
| ComplementSubtreeFlag (Optional)b(Complement Subtree Flag TLV) |
| GroupKeyData (Optional)(Group Key Data TLV) |
| SecurityAssociationID (Optional) (SAID Notification TLV) |

a This parameter is carried for CCM counter synchronization purpose when GroupKeyData is carried.

b If ComplementSubtreeFlag is not present, it means ComplementSubtreeFlag = 0.