IEEE P802.11  
Wireless LANs

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | MLO extension for CFP | | | | | | Date: 2025-09-12 | | | | | | Author(s): | | | | | | Name | Affiliation | Address | Phone | email | | Po-Kai Huang | Intel |  |  | po-kai.huang@intel.com | | Ido Ouzieli |  |  | ido.ouzieli@intel.com | | Danny Alexander |  |  | danny.alexander@intel.com | | Danny Ben-Ari |  |  | danny.ben-ari@intel.com | | Laurent Cariou |  |  | laurent.cariou@intel.com | |  |  |  |  |  | |

Abstract

This submission proposes resolutions for the following comments from comment collection on P802.11-REVmf D1.0:

199

**Revision History:**

R0: Initial version.

R1: Add missing changes in 11.3.

R2: Update document number. Revision based on the comments in the teleconference. “control frame” to “Control frame” in appropriate place. “common info field” to “Common Info field”. Use “The format of the Data field of the MLO CIGTK KDE” to align with another editorial comment. Add “with which the R1KH has negotiated control frame protection.” to modification in 13.2.2.

# CID 199

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Page.Line** | **Comment** | **Proposed Change** | **Resolution** |
| 199 | 12.5.5 | 3308.38 | CFP and CIP need to be extended to MLO. | Contribution 11-25-1461 will be submitted to resolve this CID. | REVISED –  Agree in principle with the commenter.  Instruction to TGmf Editor:  Implement the proposed text updates for CID 199 in this 11-25/1461r2. |

## Discussion:

## Proposed Text for CID 199:

**TGmf Editor: *Instruction: Modify 5.1.2 as follows:***

* Security services

Security services in IEEE Std 802.11 are provided by the authentication service and the CCMP, GCMP, BIP, and CIP. The scope of the security services provided is limited to station-to-station data, robust Management frame transmissions, and transmissions of Control frames that are defined to be protected. When CCMP or GCMP is used, the data confidentiality service is provided for Data frames and individually addressed robust Management frames. For the purposes of this standard, CCMP, GCMP, BIP, and CIP are viewed as logical services located within the MAC sublayer as shown in the reference model, Figure 4-27 (Portion of the ISO/IEC basic reference model covered in this standard) (in 4.9 (Reference model)). Actual implementations of CCMP, GCMP, BIP, and CIP are transparent to the LLC sublayer and other layers above the MAC sublayer.

The security services provided by CCMP and GCMP in IEEE Std 802.11 are as follows:

* Data Confidentiality;
* Authentication; and
* Access control in conjunction with layer management.

BIP provides message integrity and access control for group addressed robust Management frames.

CIP provides message integrity and access control for Control frames that are defined to be protected.

During the authentication exchange, both parties exchange authentication information as described in Clause 12 (Security) and Clause 13 (Fast BSS transition).

The MAC sublayer security services provided by CCMP, GCMP, BIP, and CIP rely on information from non MAC sublayer management or system entities. Management entities communicate information to CCMP, GCMP, BIP, and CIP through a set of MAC sublayer management entity (MLME) interfaces and MIB attributes; in particular, the decision tree for CCMP, GCMP, BIP, and CIP is driven by MIB attributes.

The use of TKIP is obsolete. The TKIP algorithm is unsuitable for the purposes of this standard.

**TGmf Editor: *Instruction: Modify 9.3.1.8.6 as follows:***

* Multi-STA BlockAck variant

(…existing texts…)

If the AID11 subfield of the AID TID Info subfield is not 2045, 2009, or 2047,(#M7) then the Per AID TID Info subfield has the format shown in Figure 9-66 (Per AID TID Info subfield format if the AID11 subfield is not 2045, 2009, or 2047(#M7)(#11be)).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | AID TID Info | Block Ack Starting Sequence Control | Block Ack Bitmap |
| Octets: | 2 | 0 or 2 | 0, 4, 8, 16, 32, 64, or 128 |
| * Per AID TID Info subfield format if the AID11 subfield is not 2045, 2009, or 2047(#M7)(#11be) | | | |

If the AID11 subfield of the AID TID Info subfield is equal to 2009, then the Per AID TID Info subfield has the format shown in Figure 9-67 (Per AID TID Info subfield format if the AID11 subfield is equal to 2009(#M7)) for PN and MIC. The Per AID TID Info field with the AID11 subfield equal to 2009 follows all other Per AID TID Info fields in the Multi-STA BlockAck frame that have AID11 not equal to 2047 and are addressed to STAs that have negotiated control frame protection. The Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield is reserved. The Fragment Number subfield of the Block Ack Starting Sequence Control subfield is set as defined in Table 9-44 (Fragment Number subfield encoding for the Multi-STA BlockAck variant) for the length of the Block Ack Bitmap subfield to indicate the PN And MIC field as described in Figure 9-67 (Per AID TID Info subfield format if the AID11 subfield is equal to 2009(#M7)) and Figure 9-68 (PN And MIC subfield format).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | AID TID Info | Block Ack Starting Sequence Control | Block Ack Bitmap |
| Octets: | 2 | 2 | 32 |

(#M7)

* Per AID TID Info subfield format if the AID11 subfield is equal to 2009(#M7)

The PN And MIC field has the format shown in Figure 9-68 (PN And MIC subfield format(#M7)).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | PN | MIC | Reserved |
| Octets: | 6 | 16 | 10 |

(#M7)

* PN And MIC subfield format(#M7)

The PN field contains the PN corresponding to the integrity key (see 12.5.5 (Control integrity protocol (CIP)(#M7))) indicated by the Key ID field. The PN subfield format is the same as defined in Figure 9-1078 (PN field format).(#M7)

The MIC field contains a message integrity check calculated over the BlockAck frame as defined in 12.5.5 (Control integrity protocol (CIP)(#M7)).(#M7)

If the AID11 subfield of the AID TID Info subfield is equal to 2047, then the Per AID TID Info subfield has the format shown in Figure 9-69 (Per AID TID Info subfield format if the AID11 subfield is equal to 2047(#M7)) for padding. The Per AID TID Info field(s) with the AID11 subfield equal to 2047 follow(s) all other Per AID TID Info field(s) in the Multi-STA BlockAck frame with AID11 not equal to 2047. The Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield (if present as defined in Table 9-43 (Context of the Per AID TID Info subfield and presence of optional subfields if the AID11 subfield is not 2045) is reserved and the Fragment Number subfield of the Block Ack Starting Sequence Control subfield (if present as defined in Table 9-43 (Context of the Per AID TID Info subfield and presence of optional subfields if the AID11 subfield is not 2045)) is set as defined in Table 9-44 (Fragment Number subfield encoding for the Multi-STA BlockAck variant) for the length of the Block Ack Bitmap subfield to indicate the Padding field.(#M7)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | AID TID Info | Block Ack Starting Sequence Control | Block Ack Bitmap |
| Octets: | 2 | 0 or 2 | 0, 4, 8, 16, or 32 |

* Per AID TID Info subfield format if the AID11 subfield is equal to 2047(#M7)

(…existing texts…)

**TGmf Editor: *Instruction: Modify 9.3.1.22.2 as follows:***

**9.3.1.22.2 Common Info field**

(…existing texts…)

NOTE 1—For backward compatibility with HE variant Common Info field, an EHT AP sets B22, B26, B53, and B63 to 0 and sets B56–B62 to 1 in the EHT variant Common Info field unless specified otherwise.

(…existing texts…)

If control frame protection is negotiated with at least one of the recipient(s), B61 and B62 of any variant of Common Info field are set as follows:

* The Protected Control subfield is B61 of the Common Info field of the Trigger frame. The Protected Control subfield is equal to 1 if the Trigger frame is protected as defined in 12.5.5 (Control integrity protocol (CIP)) and is equal to 0 otherwise. If the Protected Control subfield is equal to 1, B62 of the Common Info field is the Key ID field; otherwise, B62 is reserved.

The UL HE-SIG-A2 Reserved subfield of the (#11be)HE variant Common Info field carries the value to be included in the Reserved field in the HE-SIG-A2 subfield of the solicited HE TB PPDUs. A non-EHT HE AP sets the UL HE-SIG-A2 Reserved subfield of the HE variant Common Info field to all 1s unless specified otherwise.

(…existing texts…)

**TGmf Editor: *Instruction: Modify 9.3.1.22.4 as follows:***

* HE variant User Info field(#11be)

The HE variant User Info field is defined in Figure 9-110 (HE variant User Info field format(#11be)) for all Trigger frame variants except the NFRP Trigger frame, the MU-RTS TXS Trigger frame, and the Ranging Trigger frame, which are defined in 9.3.1.22.13 (NFRP Trigger frame format), 9.3.1.22.9 (MU-RTS Trigger frame format), and 9.3.1.22.14 (Ranging Trigger variant), respectively.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | B0   B11 | B12    B19 | B20 | B21  B24 | B25 | B26            B31 | B32     B38 | B39 |  |
|  | AID12 | RU  Allocation | UL FEC Coding Type | UL HE-MCS | UL DCM | SS Allocation/RA-RU Information | UL Target Receive Power | Reserved | Trigger Dependent User Info |
| Bits: | 12 | 8 | 1 | 4 | 1 | 6 | 7 | 1 | variable |
| * HE variant User Info field format(#11be) | | | | | | | | | |

The AID12 subfield in the User Info field is encoded as defined in Table 9-61 (AID12 subfield encoding(#11be)):

|  |  |
| --- | --- |
| * AID12 subfield encoding(#11be) | |
| AID12 subfield | Description |
| 0 | User Info field allocates one or more contiguous RA-RUs for associated STAs |
| 1–2007 | User Info field is addressed to an associated STA whose AID is equal to the value in the AID12 subfield |
| 2008 | Reserved |
| 2009 | User Info field indicates PN. |
| 2010 | User Info field indicatres MIC. |
| 2011-2044 | Reserved |
| 2045 | User Info field allocates one or more contiguous RA-RUs for unassociated STAs |
| 2046 | Unallocated RU |
| 2047–4094 | Reserved |
| 4095 | Disallowed in a User Info field as it indicates the start of the Padding field |
| NOTE—The Padding field, if present in a Trigger frame, is a field with all padding bits set to 1. The Padding field, if present, has a length of at least two octets and is located between the User Info List field and the FCS field (see 9.3.1.22.1 (General)). | |

NOTE—The value 2007 in the AID12 subfield can be used for an HE variant User Info field if the Trigger frame is generated by a non-EHT HE AP, whereas the value 2007 in the AID12 subfield cannot be used for an HE variant User Info field if the Trigger frame is generated by an EHT AP (see 9.3.1.22.3 (Special User Info field(#11be)) for details).

(…existing texts…)

**TGmf Editor: *Instruction: Modify 9.4.2.46 as follows:***

* FTE

(…existing texts…)

The Subelement ID field is defined in Table 9-239 (Subelement IDs):

|  |  |
| --- | --- |
| * Subelement IDs | |
| Value | Subelement Name |
| 0 | Reserved |
| 1 | R1KH-ID |
| 2 | GTK |
| 3 | R0KH-ID |
| 4 | IGTK |
| 5 | OCI |
| 6 | BIGTK |
| 7 | WIGTK |
| 8(#11be) | MLO GTK |
| 9(#11be) | MLO IGTK |
| 10(#11be) | MLO BIGTK |
| 11(#11be) | Reserved |
| 12(#M7) | CIGTK |
| <ANA> | MLO CIGTK |
| 13(#M7)–255 | Reserved |

(…existing texts…)

(#11be)The definitions of the Key ID, BIPN, Key Length, and Wrapped Key fields are the same as in the BIGTK subelement.

(#11be)The definition of the Link ID Info field is the same as in the MLO GTK subelement described above.

The MLO CIGTK subelement contains the CIGTK for a link. The MLO CIGTK subelement format is shown in Figure 9-xxx (MLO CIGTK subelement format).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Subelement ID | Length | Key Info | CIPN | Link ID Info | Wrapped Key |
| Octets: | 1 | 1 | 1 | 6 | 1 | 40 |

Figure 9-xxx MLO CIGTK subelement format

The Link ID Info field of the MLO CIGTK subelement is as defined in 9.4.1.77 (Link ID Info field). The Link ID subfield of the Link ID Info field contains the link identifier for the link (see 35.3.3.2 (Link ID)).

The definitions of the Key Info, CIPN and Wrapped Key fields are the same as in the CIGTK subelement.

**TGmf Editor: *Instruction: Modify 9.6.13.20 as follows:***

* WNM Sleep Mode Response frame format

(…existing texts…)

|  |  |
| --- | --- |
| * Optional subelement IDs for WNM Sleep Mode parameters | |
| Value | Contents of subelement |
| 0 | GTK |
| 1 | IGTK |
| 2 | BIGTK |
| 3(#11be) | MLO GTK |
| 4(#11be) | MLO IGTK |
| 5(#11be) | MLO BIGTK |
| 7(#M7) | CIGTK |
| <ANA> | MLO CIGTK |
| 8–255(#11be) | Reserved |

(…existing texts…)

The CIGTK subelement is as shown in Figure 9-1411 (WNM Sleep Mode CIGTK subelement format(#M7)).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Subelement ID | Length | Key Info | CIPN | Key |
| Octets: | 1 | 1 | 1 | 6 | 32 |
| * WNM Sleep Mode CIGTK subelement format(#M7) | | | | | |

(#M7)

The Subelement ID field is defined in 9.6.13.20 (WNM Sleep Mode Response frame format).(#M7)

The Length field is defined in 9.4.3 (Subelements).(#M7)

The Key Info field is defined in Figure 9-1412 (WNM Sleep Mode CIGTK subelement’s Key Info field format(#M7)).

|  |  |  |
| --- | --- | --- |
|  | B0 | B1 B7 |
|  | Key ID | Reserved |
| Bits: | 1 | 7 |
| * WNM Sleep Mode CIGTK subelement’s Key Info field format(#M7) | | |

(#M7)

The Key ID field contains the CIGTK key ID. (#M7)

The CIPN field contains the current RSC for the CIGTK being installed. The RSC for a CIGTK is the CIGTK packet number (CIPN).(#M7)

The Key field is the CIGTK being distributed.(#M7)

The WNM Sleep Mode Element field contains a WNM Sleep Mode element, as described in 9.4.2.80 (WNM Sleep Mode element).

The TFS Response Elements field contains zero or more TFS Response elements to specify the traffic filters, as defined in 9.4.2.79 (TFS Response element).

The OCI Element field is optionally present, and contains an OCI element as defined in 9.4.2.235 (OCI element).

The MLO CIGTK subelement contains the CIGTK for the AP operating on the link identified by the Link ID Info field carried in the subelement. The format of the MLO CIGTK subelement is as shown in Figure 9-xxxx (WNM Sleep Mode MLO CIGTK subelement format).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Subelement ID | Length | Link ID Info | Key Info | CIPN | Key |
| Octets: | 1 | 1 | 1 | 1 | 6 | 32 |

Figure 9-xxxx - WNM Sleep Mode MLO CIGTK subelement format

The Subelement ID field is defined in 9.6.13.20 (WNM Sleep Mode Response frame format).

The Length field is defined in 9.4.3 (Subelements).(#M7)

The format of the Link ID Info field is as defined in 9.4.1.77 (Link ID Info field(#11be)).

The Key Info and CIPN fields are as defined for the CIGTK subelement in Figure 9-1411 (WNM Sleep Mode CIGTK subelement format).

The Key field is the CIGTK being distributed for the AP operating on the link identified by the Link ID subfield.

**TGmf Editor: *Instruction: Modify 9.6.38.14 as follows:***

* Link Reconfiguration Response frame format(#11be)

(…existing texts…)

The Key Data subfield contains one or more MLO KDEs for group keys corresponding to added links. Each MLO KDE is encapsulated using the KDE format shown in Figure 12-38 (KDE format). For each added link, an MLO GTK KDE is included as defined in Figure 12-57 (MLO GTK KDE format(#11be)), an MLO IGTK KDE is included as defined in Figure 12-58 (MLO IGTK KDE format(#11be)), an MLO BIGTK KDE is included as defined in Figure 12-59 (MLO BIGTK KDE(#11be)), and an MLO CIGTK KDE is included as defined in Figure 12-xx (MLO CIGTK KDE) if control frame protection is negotiated.

(…existing texts…)

**TGmf Editor: *Instruction: Modify 11.2.3.15.3 as follows:***

* WNM sleep mode AP operation

(…existing texts…)

(#11be)For MLO, with RSN and a valid PTK is configured for the non-AP MLD:

* If management frame protection is negotiated for the MLDs, the current GTK, IGTK when management frame protection is negotiated, BIGTK when beacon protection is negotiated, and CIGTK when control frame protection is negotatied for each setup link shall be included in the WNM Sleep Mode Response frame using the WNM Sleep Mode MLO GTK/IGTK/BIGTK/CIGTK subelement (see 9.6.13.20 (WNM Sleep Mode Response frame format)). If a GTK/IGTK/BIGTK/CIGTK update is in progress for one or more links, the pending GTK, IGTK when management frame protection is negotiated, BIGTK when beacon protection is negotiated, and CIGTK when control frame protection is negotiated for each of the affected AP(s) shall be included in the WNM Sleep Mode Response frame using the WNM Sleep Mode MLO GTK/IGTK/BIGTK/CIGTK subelement (see 9.6.13.20 (WNM Sleep Mode Response frame format)). A non-AP MLD identifies the corresponding link to which the GTK/IGTK/BIGTK/CIGTK belongs based on the value of the Link ID subfield included in the subelement of the Key Data field.
* If management frame protection is not negotiated for the MLDs, the current GTK for each setup link shall be sent to the non-AP MLD using a group key handshake (see 12.7.7 (Group key handshake)) immediately following the WNM Sleep Mode Response frame. If a GTK update is in progress for a setup link, the pending GTK for the setup link shall be sent to the STA using another group key handshake immediately after the current GTK of the setup link has been sent.

**TGmf Editor: *Instruction: Modify 11.3.4.4 as follows:***

11.3.4.4 Deauthentication—originating STA or MLD(#11be)

The originating STA (#11be)or MLD shall deauthenticate with the indicated STA or MLD, respectively, using the following procedure:

(…existing texts…)

e) The SME, upon receipt of an MLME-DEAUTHENTICATE.confirm primitive, shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA, CIGTKSA, WIGTKSA, WTKSA, and TPKSA (including temporal keys) held for communication with the indicated STA (#11be)or MLD by using the MLMEDELETEKEYS. request primitive (see 12.6.16 (RSNA security association termination)) and by generating an MLME-SETPROTECTION.request(None) primitive.

(…existing texts…)

**TGmf Editor: *Instruction: Modify 11.3.4.5 as follows:***

11.3.4.5 Deauthentication—destination STA or MLD(#11be)

Otherwise, upon receipt of a Deauthentication frame from a STA (#11be)or an MLD for which the state is State 1a, State 2, State 3, or State 4, the destination STA or MLD, respectively, shall deauthenticate with the originating STA or MLD, respectively, using the following procedure:

(…existing texts…)

b) Upon receiving an MLME-DEAUTHENTICATE.indication primitive, the SME shall

1) Delete any PTKSA, GTKSA, IGTKSA, BIGTKSA, CIGTKSA, WIGTKSA, WTKSA and TPKSA (including temporal keys) held for communication with the originating STA (#11be)or the originating MLD by using the MLME-DELETEKEYS.request primitive (see 12.6.16 (RSNA security association termination)) and by generating an MLME-SETPROTECTION.request(None) primitive.

(…existing texts…)

**TGmf Editor: *Instruction: Modify 11.3.5.2 as follows:***

11.3.5.2 Non-AP STA, non-AP MLD, and non-PCP STA association initiation procedures(#11be)

The SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA, WIGTKSA, WTKSA, CIGTKSA,(#M7) and TPKSA (including temporal keys) held for communication with the AP or PCP by using MLMEDELETEKEYS. request primitive (see 12.6.16 (RSNA security association termination)) before invoking MLME-ASSOCIATE.request primitive.

(#11be)The SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA, CIGTKSA and temporal keys held for communication with the AP MLD by using MLME-DELETEKEYS.request primitive (see 12.6.16 (RSNAsecurity association termination)) before invoking MLME-ASSOCIATE.request primitive.

(…existing texts…)

**TGmf Editor: *Instruction: Modify 11.3.5.7 as follows:***

11.3.5.7 Non-AP STA, non-AP MLD, and non-PCP STA disassociation receipt procedure(#11be)

Upon receipt of a Disassociation frame from an (#11be)AP, AP MLD, or PCP for which the state is State 3 or State 4, if management frame protection was not negotiated when the PTKSA(s) were created, or if management frame protection was negotiated when the PTKSA(s) were created and the frame is not discarded per management frame protection processing, a non-AP STA, non-AP MLD, and non-PCP STA, respectively, shall disassociate from the AP, AP MLD, or PCP using the following procedure:

(…existing texts…)

c) Upon receiving the MLME-DISASSOCIATE.indication primitive, the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA, CIGTKSA, WIGTKSA, WTKSA, and TPKSA (including temporal keys) held for communication with the (#11be)AP, AP MLD, or PCP by using the MLME-DELETEKEYS. request primitive (see 12.6.16 (RSNA security association termination)) and by invoking an MLME-SETPROTECTION.request(None) primitive. The MM-SME shall perform this process for each STA whose address was included in the MMS parameter of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive that established the association.

(…existing texts…)

**TGmf Editor: *Instruction: Modify 12.2.13 as follows:***

* Requirements for control frame protection(#M7)

The Control frames that are defined to be protected are:

* Individually addressed Trigger frames exchanged between an associated non-AP STA and an AP that have negotiated control frame protection.
* Group addressed Trigger frames transmitted by an AP that are addressed to at least one associated non-AP STA that has negotiated control frame protection
* Individually addressed Multi-STA BlockAck frames exchanged between an associated non-AP STA and an AP that have negotiated control frame protection.
* Group addressed Multi-STA BlockAck frames transmitted by an AP that are addressed to at least one associated non-AP STA that has negotiated control frame protection.
* Individually addressed Compressed BlockAckReq frames exchanged between an associated non-AP STA and an AP that have negotiated control frame protection.

Individually addressed Multi-TID BlockAckReq frames exchanged between an associated non-AP STA and an AP that have negotiated control frame protection.

When control frame protection is negotiated, individually addressed and group addressed Control frames that are defined to be protected shall be encapsulated using the procedure defined in 12.6.22 (Protection of Control frames(#M7)) after the PTKSA is established. When control frame protection is negotiated, the following frame from the peer STA with negotiated control frame protection shall be discarded after the PTKSA is established.

* Unprotected individually addressed Control frames that are defined to be protected
* Unprotected group addressed Control frames that are defined to be protected
* Ack frame
* Compressed BlockAck frame
* GCR BlockAck frame if GCR is supported
* GCR BlockAckReq frame if GCR is supported

NOTE- If a group addressed Control frame from an associated AP is not protected and a non-AP STA that negotiates control frame protection is not one of the recipient(s) of the group addressed Control frame (e.g. a Trigger frame from the associated AP that does not negotiate control frame protection with any of the recipients), then the non-AP STA is not required to discard the group addressed Control frame since that group addressed Control frame is not defined to be protected.

**TGmf Editor: *Instruction: Modify 12.5.5 as follows:***

* Control integrity protocol (CIP)(#M7)
* Overview

The control integrity protocol (CIP) provides integrity and replay protection for the Control frames that are defined to be protected.

The cipher suite that is used for individually addressed Control frames in CIP is determined based on the negotiated pairwise cipher suite for individually addressed Data and Management frames. When GCMP-256 is used as the pairwise cipher suite, GMAC-256 shall be used for CIP of individually addressed Control frames. CIP cannot be used if another pairwise cipher suite is negotiated.

GMAC-256 shall be used for CIP of group addressed Control frames.

NIST Special Publication 800-38D defines the GMAC algorithm. GMAC-256 for CIP uses AES with a 256-bit integrity key. The authentication tag is not truncated and shall be 128 bits (16 octets).

If CIP is used, the same TK is used both for protecting individually addressed Data and Management frames with GCMP-256 and for protecting individually addressed Control frames with GMAC-256. The 4 most significant bits of the PN for protecting individually addressed Data and Management frames shall be set to a value that is less than 15 and the 4 most significant bits of the PN for protecting individually addressed Control frames shall be set to all 1s.

CIP uses the TK to compute the MIC of individually addressed Control frames that are defined to be protected.

CIP uses the control integrity group temporal key (CIGTK) and CIGTK packet number (CIPN) delivered by the AP to compute the MIC of group addressed Control frames that are defined to be protected. In a multiple BSSID set, all APs in the multiple BSSID set deliver and use the same CIGTK.

* Encapsulation format

To provide integrity and replay protection, CIP utilizes the Key ID, PN, and MIC in the Control frames that are defined to be protected.

The frame format is described in 9.3.1.7 (BlockAckReq frame format), 9.3.1.8 (BlockAck frame format), and 9.3.1.22 (Trigger frame format).

* CIP AAD construction

The CIP additional authentication data (AAD) is constructed from the Control frame header. AAD construction is performed as follows without any bits masked out:

* Frame Control field
* Duration field
* RA field
* TA field

Figure 12-32 (CIP AAD construction) depicts the format of the AAD.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Frame Control | Duration | RA | TA |
| Octets: | 2 | 2 | 6 | 6 |
| * CIP AAD construction | | | | |

* Replay counters and packet numbers

When CIP is negotiated between an AP and a non-AP STA:

* For non-MLO, the non-AP STA and the AP shall maintain a single PN (48-bit counter) for each PTKSA for protecting individually addressed Control frames. For MLO, in each setup link, the corresponding non-AP STA affiliated with the non-AP MLD and the corresponding AP affiliated with the AP MLD shall maintain a single PN (48-bit counter) for each PTKSA for protecting individually addressed Control frames. The PN shall be implemented as a 48-bit strictly increasing integer. The 4 most significant bits of the PN shall be set to all 1s. The 44 least significant bits of the PN shall be initialized to 0.
* For non-MLO, the non-AP STA and the AP shall maintain a 48-bit replay counter for each PTKSA to check replay of individually addressed Control frames between them that are defined to be protected. For MLO, in each setup link, the corresponding non-AP STA affiliated with the non-AP MLD and the corresponding AP affiliated with the AP MLD shall maintain a 48-bit replay counter for each PTKSA to check replay of individually addressed Control frames that are defined to be protected
* For non-MLO, the AP shall maintain a single PN (48-bit counter) for each CIGTK. For MLO, each AP affiliated with the AP MLD shall maintain a single PN (48-bit counter) for each CIGTK. The PN shall be implemented as a 48-bit strictly increasing integer, initialized to 0 when the corresponding CIGTK is initialized. A single PN space shall be maintained for all APs in a multiple BSSID set.
* For non-MLO, the non-AP STA shall maintain a 48-bit replay counter for each CIGTK to check replay of group addressed Control frames from the AP that are defined to be protected. For MLO, in each setup link, the corresponding non-AP STA affiliated with the non-AP MLD shall maintain a 48-bit replay counter for each CIGTK to check replay of group addressed Control frames that are defined to be protected.
* Transmission

When a STA transmits a Control frame that is defined to be protected, it shall:

* Select the TK (if the Control frame is individually addressed) or CIGTK (if the Control frame is group addressed) currently active for transmission of individually addressed Control frames or group addressed Control frames, respectively.
* Increment the PN to obtain a fresh nonzero PN.
* Set the Key ID field to the corresponding key ID and set the PN0 field, the PN1 field, the PN2 field, the PN3 field, the PN4 field, and the PN5 field based on the corresponding PN. Concatenate the TA field and the and the non-negative integer value of the PN to form a nonce, i.e., the initialization vector.
* Compute the AAD as specified in 12.5.5.3 (CIP AAD construction).
* For the Trigger frame, compute an integrity value over the concatenation of the AAD and its contents after the TA field up to and including the last User Info field that precedes the first User Info field that carries the MIC (see Figure 9-107 (Formats of User Info fields with AID12 subfield equal to 2010(#M7))) and excluding anything from any of the User Info fields that carry the MIC or other subsequent User Info fields (if any) that follow the User Info fields that carry the MIC. Otherwise, compute an integrity value over the concatenation of the AAD and contents after the TA field and before the MIC field. Insert the output into the MIC field.
* Include padding if needed to satisfy the padding requirement(s) of the intended recipient(s).
* Transmit the frame.
* Reception

When a STA receives a Control frame that is defined to be protected, it shall:

* Identify the appropriate TK (if the Control frame is individually addressed) or CIGTK (if the Control frame is group addressed) and associated state based on the Key ID field. If no such TK or CIGTK exists, silently discard the frame and terminate CIP processing for this reception.
* Perform replay protection on the received frame. The receiver shall interpret the PN field (constructed from the PN0 field, the PN1 field, the PN2 field, the PN3 field, the PN4 field, and the PN5 field if needed) as a 48-bit unsigned integer. The receiver shall compare the PN to the value of the replay counter identified by the Key ID field and the RA field. If the value from the PN field is less than or equal to the replay counter value, the receiver shall discard the frame and increment the dot11RSNAStatsCIPReplays counter by 1.
* Concatenate the TA field and the non-negative integer inserted into the PN field to form a nonce, i.e., the initialization vector.
* Compute the AAD as specified in 12.5.5.3 (CIP AAD construction).
* Extract and save the received MIC value. If the MIC value does not exist, silently discard the frame and terminate CIP processing for this reception. For the Trigger frame, compute a verifier over the concatenation of the AAD and its contents after the TA field up to and including the last User Info that precedes the first User Info field that carries the MIC (see Figure 9-107 (Formats of User Info fields with AID12 subfield equal to 2010(#M7))) and excluding anything from any of the User Info fields that carry part of the MIC value or other subsequent User Info fields (if any) that follow the User Info fields that carry the MIC value. Otherwise, compute a verifier over the concatenation of the AAD and contents following the TA field and before the MIC field. If the computed verifier is not equal to the received MIC value, then the receiver shall discard the frame, increment the dot11RSNAStatsCIPMICErrors counter by 1, and terminate CIP processing for this reception.
* Update the corresponding replay counter identified by the Key ID field and the RA field with the value of the PN field.
* Padding

A STA transmitting a BCC-encoded PPDU that contains a protected Control frame shall ensure that for each target STA, the number of bits in the PSDU following *Vlast* is at least *MPAD,MAC* as defined in Equation (12-5), which is based on the MIC padding delay indicated by the target STA (see 9.4.2.316 (CIP Capabilities element(#M7))), where *Vlast* is:

* MIC[127] if the frame is a BlockAckReq frame or a Multi-STA BlockAck frame.
* The last bit of the User Info field containing MIC[127] (see Figure 9-107 (Formats of User Info fields with AID12 subfield equal to 2010(#M7))) if the frame is a Trigger frame.
* *MPAD,MAC* = *NDBPSMPAD*

where

*NDBPS* is defined in Table 17-4 (Modulation-dependent parameters) for a non-HT and non-HT duplicate PPDU, Table 19-7 (Frequently used parameters) for an HT PPDU, Table 21-6 (Frequently used parameters) for a VHT PPDU, and Table 27-16 (Frequently used parameters) for an HE PPDU. If the protected Control frame is carried in an HE MU PPDU, *NDBPS* is replaced by *NDBPS,u* of the target user in Equation (12-5).

*MPAD* is defined as follows:

—  For a non-HT PPDU, HT PPDU, or VHT PPDU, *MPAD* is:

- 0 if the MIC padding delay is 0 µs.

- 1 if the MIC padding delay is 4 µs.

- 2 if the MIC padding delay is 8 µs.

- 3 if the MIC padding delay is 12 µs.

- 4 if the MIC padding delay is 16 µs.

- 5 if the MIC padding delay is 20 µs.

- 6 if the MIC padding delay is 24 µs.

- 7 if the MIC padding delay is 28 µs.

- 8 if the MIC padding delay is 32 µs.

—  For an HE PPDU, *MPAD* is:

- 0 if the MIC padding delay is 0 µs.

- 1 if the MIC padding delay is less than or equal to 16 µs.

- 2 if the MIC padding delay is less than or equal to 32 µs.

Define *VProc* as the duration of the PPDU that is after the OFDM symbol containing the last coded bit of the LDPC codeword that encodes *Vlast* minus *TPE, nominal* defined in 27.3.13 (Packet extension) for an HE PPDU.

A STA transmitting an LDPC-encoded PPDU that contains a protected Control frame shall ensure that for each target STA, *VProc* is greater than or equal to the MIC padding delay indicated by the target STA (see 9.4.2.316 (CIP Capabilities element(#M7))).

In an A-MPDU, a STA shall not use other MPDUs that are different from the protected Control frame as the padding to satisfy the requirements of the MIC padding delay.

A STA transmitting a BCC-encoded PPDU that contains the last frame soliciting a protected Control frame shall ensure that for each target STA, the number of bits in the PSDU following *Clast* is at least *MPAD,MAC*, which is based on the MIC padding delay indicated by the target STA (see 9.4.2.316 (CIP Capabilities element(#M7))), where *Clast* is:

* The last bit of the FCS of the frame if the frame is not a protected Control frame
* *Vlast* if the frame is a Trigger frame (see 9.3.1.22.1 (General)) or a BlockAckReq frame (see 9.3.1.7.1 (Overview))

Define *CProc* as the duration of the PPDU that is after the OFDM symbol containing the last coded bit of the LDPC codeword that encodes *Clast* of the frame soliciting a protected Control frame minus *TPE, nominal* defined in 27.3.13 (Packet extension) for an HE PPDU.

A STA transmitting an LDPC-encoded PPDU that contains the last frame soliciting a protected Control frame shall ensure that for each target STA, *CProc* is greater than or equal to the MIC padding delay indicated by the target STA (see 9.4.2.316 (CIP Capabilities element(#M7))).

Except for the exception specified in this subclause, a STA may use any type of padding to satisfy the requirements, such as using the Padding field in a Trigger frame, a Compressed BlockAckReq frame or a Multi-TID BlockAckReq frame, using one or more Per-AID TID Info subfields with the AID11 subfield equal to 2047 in a Multi-STA BlockAck frame, using pre-EOF A-MPDU padding, using post-EOF A-MPDU padding, or aggregating other MPDUs in the A-MPDU.

**TGmf Editor: *Instruction: Modify 12.6.1.1.14 as follows:***

* CIGTKSA(#M7)

An Authenticator's SME creates a CIGTKSA when control frame protection is negotiated for the first time. An AP MLD’s SME creates a CIGTKSA for a link when control frame protection is negotiated for the first time on that link. A CIGTKSA has the same lifetime as the BSS, unless superseded.

A Supplicant's SME creates a CIGTKSA when control frame protection is negotiated, upon receiving a CIGTK from its Authenticator. A non-AP MLD’s SME creates a CIGTKSA for each of its setup link when control frame protection is negotiated.

A CIGTKSA consists of the following:

* Direction vector (whether the CIGTK is used to transmit or receive)
* Key ID
* CIGTK
* Authenticator MAC address
* For MLO, the MAC address of the AP operating on the link corresponding to the CIGTKSA.

**TGmf Editor: *Instruction: Modify 12.6.19 as follows:***

* RSNA rekeying

(…existing texts…)

(#11be)For MLO, the AP MLD’s Authenticator manages packet number assignment for the PTKSA with a non-AP MLD. For a given link, the affiliated AP’s Authenticator manages packet number assignment for the IGTKSA, GTKSA, BIGTKSA, or CIGTKSA. If an IGTKSA, GTKSA, BIGTKSA, or CIGTKSA update is triggered, the affiliated AP updates group keys for the given link through a group key handshake between the AP MLD and non-AP MLD.

(…existing texts…)

**TGmf Editor: *Instruction: Modify 12.6.22 as follows:***

* Protection of Control frames(#M7)

This subclause defines rules that shall be followed by an RSNA non-AP STA on a link with an associated AP.

Control frame protection is an optional feature. A STA that supports control frame protection has dot11CIPActivated equal to true. If both the associated non-AP STA and AP have set the CIP Supported field to 1 in the RSNXE, then control frame protection is negotiated on the corresponding link.

NOTE – For MLO, all STAs affiliated with a MLD advertise the same RSNXE. Hence, between an associated non-AP MLD and AP MLD, control frame protection is negoatied either on all the setup links or none of the setup links.

Protection of group addressed Control frames that are defined to be protected shall be provided by a service in the MLME as described in 11.55 (Group addressed control frame protection procedures(#M7)). Protection of individually addressed Control frames that are defined to be protected shall be provided by a service in the MLME (see 12.2.4 (RSNA establishment)).

A non-AP STA (for non-MLO) or a non-AP STA affiliated with a non-AP MLD indicates in the CIP Capabilities element included in the (Re)Association Request frame the padding duration of the protected Control frames and PPDUs that solicit protected Control frames. An AP (for non-MLO) or an AP affiliated with an AP MLD indicates in the CIP Capabilities element included in the (Re)Association Response frame the padding durations of the protected Control frames and PPDUs that solicit protected Control frames.

A STA shall use a protected Multi-STA BlockAck frame to provide acknowledgement of individually addressed frames that solicit an acknowledgement to another STA if the STAs have negotiated control frame protection.

A protected GCR MU-BAR Trigger frame shall solicit a protected Multi-STA BlockAck frame instead of a GCR BlockAck frame. A non-AP STA that supports GCR and that has negotiated control frame protection shall include a protected Multi-STA BlockAck frame, instead of a GCR BlockAck frame, in the TB PPDU that is sent in response to a protected GCR MU-BAR Trigger frame (see 9.3.1.22.11 (GCR MU-BAR Trigger frame format)). An AP shall not send a GCR BlockAckReq frame to a non-AP STA that supports GCR and that has negotiated control frame protection.

A protected MU-BAR Trigger frame shall solicit a protected Multi-STA BlockAck frame. A non-AP STA that has negotiated control frame protection shall include a protected Multi-STA BlockAck frame in the TB PPDU that is sent in response to a protected MU-BAR Trigger frame (see 9.3.1.22.8 (MU-BAR Trigger frame format)).

**TGmf Editor: *Instruction: Modify 12.7.2 as follows:***

* EAPOL-Key frames

(…existing texts…)

|  |  |  |
| --- | --- | --- |
| * KDE selectors | | |
| OUI | Data type | Meaning |
| … |  |  |
| 00-0F-AC(#11be) | 16 | MLO GTK KDE |
| 00-0F-AC(#11be) | 17 | MLO IGTK KDE |
| 00-0F-AC(#11be) | 18 | MLO BIGTK KDE |
| 00-0F-AC(#11be) | 19 | MLO Link KDE |
| 00-0F-AC(#11bh) | 20 | Device ID KDE |
| 00-0F-AC(#11bh) | 21 | IRM KDE |
| 00-0F-AC(#11bh) | 22 | PASN ID KDE |
| 00-0F-AC(#Ed) | 23 | Reserved |
| 00-0F-AC(#M7) | 24 | CIGTK KDE |
| 00-0F-AC | <ANA> | MLO CIGTK KDE |
| 00-0F-AC | 25(#M7)–255 | Reserved |
| Other OUI or CID | Any | Vendor specific |

(…existing texts…)

The format of the CIGTK KDE is shown in Figure 12-55 (CIGTK KDE(#M7)).(#M7)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Key Info | CIPN | CIGTK |
| Octets: | 1 | 6 | 32 |
| * CIGTK KDE(#M7) | | | |

The Key Info field is defined in Figure 12-56 (CIGTK KDE's Key Info field format(#M7)).

|  |  |  |
| --- | --- | --- |
|  | B0 | B1 B7 |
|  | Key ID | Reserved |
| Bits: | 1 | 7 |
| * CIGTK KDE's Key Info field format(#M7) | | |

(#M7)

The Key ID field contains the CIGTK key ID.(#M7)

The CIPN field contains the CIPN used to protect the last protected group addressed Control frame. It is used by the receiver as the initial value for the replay counter for the CIGTK.(#M7)

The CIGTK field contains the CIGTK.(#M7)The format of the Data field of the MLO CIGTK KDE is shown in Figure 12-xx (MLO CIGTK KDE).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Key Info | | CIPN | | Reserved | LinkID | CIGTK |
| Bits | 8 | | 48 | | 4 | 4 | 256 |
|  | |  | | Figure 12-xx MLO CIGTK KDE | | | |

The definitions of the Key Info and CIGTK fields are the same as in the CIGTK KDE described above.

The CIPN field contains the CIPN used to protect the last protected group addressed Control frame corresponding to the LinkID field. It is used by the receiver as the initial value for the replay counter for the CIGTK.

The LinkID field contains the link identifier that corresponds to the link this CIGTK applies.

**TGmf Editor: *Instruction: Modify 12.7.4 as follows:***

* EAPOL-Key PDU notation

The following notation is used throughout the remainder of 12.7 (Keys and key distribution) and 13.4 (FT initial mobility domain association) to represent EAPOL-Key PDUs:

EAPOL-Key(S, M, A, I, K, Reserved, RSC, ANonce/SNonce, MIC, {Key Data})

where

S is the Secure bit of the Key Information field

M means the MIC is available in message. This should be set in all messages except message 1 of a 4-way handshake. This is the Key MIC Present bit of the Key Information field. When using an AEAD cipher, this Key MIC Present bit is set to 0 regardless of the M parameter value.

A means a response is required to this message. This is used when the receiver should respond to this message. This is the Key Ack bit of the Key Information field.

I is the Install bit: indicates whether to install (1) or not install (0) for the pairwise key. This is the Install bit of the Key Information field.

K is the key type: P (Pairwise), G (Group); this is the Key Type bit of the Key Information field

Reserved is reserved

RSC is the RSC; this is the RSC field

ANonce/SNonce is the Authenticator or Supplicant nonce, respectively. This is the Key Nonce field.

MIC is the integrity check, which is generated using the PTK-KCK. This is the Key MIC field. When using an AEAD cipher, this parameter is ignored, and no Key MIC field is included in the EAPOL-Key PDU.

{Key Data} is a sequence of zero or more elements and KDEs, concatenated and contained in the Key Data field, where

RSNE is the RSNE, described in 9.4.2.23 (RSNE)

RSNE[KeyName] is the RSNE, with the PMKID List field set to KeyName

GTK[N] is the GTK KDE, with the Key ID field set to N (The key ID specifies which index is used for this GTK. Indices 0 and 3 shall not be used for GTKs)

(#11be)MAC Address is the MAC address KDE, containing the MAC address of the MLD with which the transmitting STA is affiliated

FTE is the FTE, described in 9.4.2.46 (FTE)

MDE is the MDE, described in 9.4.2.45 (MDE)

TIE[IntervalType] is a TIE of type IntervalType, as described in 9.4.2.47 (TIE), containing e.g., for type KeyLifetime, the lifetime of the FT key hierarchy

IGTK[M] is the IGTK KDE, with the Key ID field set to M

IPN is the last IPN, as provided by the IGTK KDE

BIGTK[Q] is the BIGTK KDE, with the Key ID field set to Q

BIPN is the last BIPN, as provided by the BIGTK KDE

WIGTK[R] is the WIGTK KDE, with the Key ID field set to R

WIPN is the last WIPN, as provided by the WIGTK KDE

CIGTK[S](#M7) is the CIGTK KDE, with the Key(#Ed) ID field set to S

CIPN(#M7) is the last CIPN, as provided by the CIGTK KDE

PMKID is the PMKID KDE and is the PMK identifier used during the 4-way handshake for PMK identification

OCI is the OCI KDE

Device ID(#11bh) is the Device ID KDE, described in 9.4.2.317 (Device ID element(#11bh))

IRM(#11bh) is the IRM KDE, described in 9.4.2.318 (IRM element(#11bh))

PASN ID(#11bh) is the PASN ID KDE, described in 9.4.2.321 (PASN ID element(#11bh))

RSNXE is the RSNXE, described in 9.4.2.240 (RSNXE)

PMKID is the PMK identifier for the PMKSA selected by the Authenticator

SSID is the SSID element, described in 9.4.2.2 (SSID element)

[a] means that a is optionally or conditionally present in {Key Data}

(#11be)MLO GTK is the GTK for the AP affiliated with the AP MLD for the link specified by the value in the LinkID field.

(#11be)MLO IGTK is the IGTK for the AP affiliated with the AP MLD for the link specified by the value in the LinkID field.

(#11be)MLO BIGTK is the BIGTK for the AP affiliated with the AP MLD for the link specified by the value in the LinkID field.

MLO CIGTK is the CIGTK for the AP affiliated with the AP MLD for the link specified by the value in the LinkID field.

(#11be)MLO Link is the MAC address, RSNE, and RSNEX, if advertised, for the STA affiliated with the MLD specified by the value in the LinkID field.

(#11be)“an” means that the KDE could occur multiple times in the field for *n* links.

**TGmf Editor: *Instruction: Modify 12.7.6.1 as follows:***

* General

RSNA defines a protocol using EAPOL-Key PDUs called the *4-way handshake*. The handshake completes the IEEE 802.1X authentication process. The information flow of the 4-way handshake is as follows:

Message 1: Authenticator ® Supplicant: OCI(0 or 1, 0, 1, 0, P, 0, 0, ANonce, 0, {[PMKID][, MAC Address])(#11be)

Message 2: Supplicant ® Authenticator: EAPOL-Key(0 or 1, 1, 0, 0, P, 0, 0, SNonce, MIC, {RSNE [, RSNXE] [, OCI] [, MAC Address, MLO Linkn] [, Device ID](#11be)(#11bh)})

Message 3: Authenticator®Supplicant:   
EAPOL-Key(1, 1, 1, 1, P, 0, RSC, ANonce, MIC, {RSNE [, RSNXE] [, OCI], GTK(N) [, IGTK(M, IPN)] [, BIGTK(Q, BIPN)] [, WIGTK(R, WIPN)] [, CIGTK(S, CIPN)] [, MAC Address, MLO Linkm, MLO GTKm] [, MLO IGTKn] [, MLO BIGTKn] [, MLO CIGTKn] [, SSID] [, Device ID] [, IRM] [, PASN ID](#11be)(#11bh)})

Message 4: Supplicant ® Authenticator: EAPOL-Key(1, 1, 0, 0, P, 0, 0, 0, MIC, {[MAC Address] [, IRM KDE] [,IRM](#11be)(#11bh)}).

(…existing texts…)

**TGmf Editor: *Instruction: Modify 12.7.6.4 as follows:***

* 4-way handshake message 3

(…existing texts…)

Key Data =

* For PTK generation for the current operating band, the AP’s Beacon/Probe Response frame’s RSNE for the current operating band, and, optionally, a second RSNE that is the Authenticator’s pairwise cipher suite assignment for the current operating band, and, if a group data cipher has been negotiated, the GTK and the GTK’s key ID (see 12.7.2 (EAPOL-Key frames)) for the current operating band, and if management frame protection is negotiated, the IGTK KDE, and if beacon protection is enabled, the BIGTK KDE, and if control frame protection is negotiated, the CIGTK KDE, and if WUR frame protection is negotiated, the WIGTK KDE, and when this message 3 is part of a fast BSS transition initial mobility domain association or an association started through the FT protocol, the PMKR1Name calculated according to the procedures of 12.7.1.6.4 (PMK-R1) in the PMKID List field of the RSNE and the FTE with the same contents as in the (Re)Association Response frame, the MDE with the same contents as in the (Re)Association Response frame, the reassociation deadline timeout set to the minimum  of dot11FTReassociationDeadline and the key lifetime in the TIE[ReassociationDeadline], and the PTK lifetime in the TIE[KeyLifetime]; or
* (#11be)For MLO, the MLO GTK KDE for each setup link (see 35.3.5.1 (ML (re)setup procedure)). If management frame protection is negotiated, the MLO IGTK KDE for each setup link. If beacon protection is enabled, the MLO BIGTK KDE for each setup link. If control frame protection is negoatied, the MLO CIGTK KDE for each setup link. When this message 3 is part of a fast BSS transition initial mobility domain association or an association started through the FT protocol, the PMKR1Name calculated according to the procedures of 12.7.1.6.4 (PMK-R1) in the PMKID List field of the RSNE and the FTE with the same contents as in the (Re)Association Response frame, the MDE with the same contents as in the (Re)Association Response frame, the reassociation deadline timeout set to the minimum of dot11FTReassociationDeadline and the key lifetime in the TIE[ReassociationDeadline], and the PTK lifetime in the TIE[KeyLifetime]; or

(…existing texts…)

**TGmf Editor: *Instruction: Modify 12.7.7 as follows:***

* Group key handshake
* General

The Authenticator uses the Group key handshake to send a new GTK and, if management frame protection is negotiated, a new IGTK, and if beacon protection is enabled, a new BIGTK, and if WUR frame protection is negotiated, a new WIGTK, to the Supplicant. (#11be)When the Authenticator is an AP MLD and the Supplicant is a non-AP MLD, the Authenticator may also use the Group key handshake to send new GTK(s) for any of the setup links and, if management frame protection is negotiated, new IGTK(s) for any of the setup links, and if beacon protection is enabled, new BIGTK(s) for any of the setup links to the Supplicant.

The Authenticator may initiate the exchange at any time when a Supplicant is disassociated or deauthenticated.

Message 1: Authenticator ® Supplicant:

EAPOL-Key(1, 1, 1, 0, G, 0, RSC, 0, MIC, {GTK(N) [, OCI} [, IGTK(M, IPN)] [, BIGTK(Q, BIPN)] [, WIGTK(R, WIPN)] [, CIGTK(S, CIPN)] [, MLO GTKn] [, MLO IGTKn] [, MLO BIGTKn] [, MLO CIGTKn] (#11be)(#M7)})

Message 2: Supplicant ® Authenticator: EAPOL-Key(1, 1, 0, 0, G, 0, 0, 0, MIC, {[OCI]})

NOTE 1—Elements and KDEs in the key data field can be included in any order.

The following apply:

* RSC denotes the last TSC or packet number sent using the GTK.
* GTK[N] denotes the GTK with its key ID as encapsulated using the KDE defined in 12.7.2 (EAPOL-Key frames) using the PTK-KEK defined in 12.7.1.3 (Pairwise key hierarchy) and associated IV.
* IGTK[M], when present, denotes the IGTK with its key ID as encapsulated using the KDE defined in 12.7.2 (EAPOL-Key frames) using the PTK-KEK defined in 12.7.1.3 (Pairwise key hierarchy) and associated IV.
* BIGTK[Q], when present, denotes the BIGTK with its key ID as encapsulated using the KDE as defined in 12.7.2 (EAPOL-Key frames) using the PTK-KEK defined in 12.7.1.3 (Pairwise key hierarchy) and associated IV.
* WIGTK[R], when present, denotes the WIGTK with its key ID as encapsulated using the KDE as defined in 12.7.2 (EAPOL-Key frames) using the PTK-KEK defined in 12.7.1.3 (Pairwise key hierarchy) and associated IV.
* CIGTK[S], when present, denotes the CIGTK with its key ID as encapsulated using the KDE as defined in 12.7.2 (EAPOL-Key frames) using the PTK-KEK defined in 12.7.1.3 (Pairwise key hierarchy) and associated IV.(#M7)
* The MIC is computed over the body of the EAPOL-Key PDU (with the MIC field zeroed for the computation) using the PTK-KCK defined in 12.7.1.3 (Pairwise key hierarchy).
* The OCI represents the current operating channel information using which the EAPOL-Key PDU is sent. OCI is included when dot11RSNAOperatingChannelValidationActivated is true on the STA sending the message.
* (#11be)MLO GTKn, when present, denotes the GTK for the AP affiliated with the AP MLD for the link specified by LinkID n as defined in 12.7.2 (EAPOL-Key frames).
* (#11be)MLO IGTKn, when present, denotes the IGTK for the AP affiliated with the AP MLD for the link specified by LinkID n as defined in 12.7.2 (EAPOL-Key frames).
* (#11be)MLO BIGTKn, when present, denotes the GTK for the AP affiliated with the AP MLD for the link specified by LinkID n as defined in 12.7.2 (EAPOL-Key frames).
* MLO CIGTKn, when present, denotes the CIGTK for the AP affiliated with the AP MLD for the link specified by LinkID n as defined in 12.7.2 (EAPOL-Key frames).

(…existing texts…)

* Group key handshake message 1

Message 1 uses the following values for each of the EAPOL-Key PDU fields:

Descriptor Type **=** N – see 12.7.2 (EAPOL-Key frames)

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA-1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0

Key Type = 0 (Group)

Install = 0

Key Ack = 1

Key MIC Present = 0 when using an AEAD cipher or 1 otherwise

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 1

Reserved = 0

Key Length = 0

Key Replay Counter = *m*, greater than in the last EAPOL-Key PDU transmitted that was not an EAPOL-Key request frame

Key Nonce = 0

EAPOL-Key IV = 0 (Version 2) or random (Version 1)

RSC = last TSC or PN for the GTK (#11be)for non-MLO. 0 for MLO.

Key MIC = MIC(PTK-KCK, EAPOL); not present when using an AEAD cipher

Key Data Length = length of Key Data field in octets

Key Data =

* (#11be)For non-MLO, GTK and the GTK’s key ID (see 12.7.2 (EAPOL-Key frames))
* (#11be)For non-MLO, when present, IGTK, IGTK’s key ID, and IPN (see 12.7.2 (EAPOL-Key frames))
* (#11be)For non-MLO, when present, BIGTK, BIGTK’s key ID, and BIPN (see 12.7.2 (EAPOL-Key frames))
* (#11be)For non-MLO, when present, WIGTK, WIGTK’s key ID, and WIPN (see 12.7.2 (EAPOL-Key frames))
* For non-MLO, when present, CIGTK, CIGTK's key ID, and CIPN (see 12.7.2 (EAPOL-Key frames))(#M7)
* OCI KDE when dot11RSNAOperatingChannelValidationActivated is true on the Authenticaton.
* (#11be)For MLO, the MLO GTK KDE (see 12.7.2 (EAPOL-Key frames)) for each of the setup links with a new GTK.
* (#11be)For MLO, when present, the MLO IGTK KDE (see 12.7.2 (EAPOL-Key frames)) for each of the setup links with a new IGTK.
* (#11be)For MLO, when present, the MLO BIGTK KDE (see 12.7.2 (EAPOL-Key frames)) for each of the setup links with a new BIGTK.
* For MLO, when present, the MLO CIGTK KDE (see 12.7.2 (EAPOL-Key frames)) for each of the setup links with a new CIGTK.

(…existing texts…)

**TGmf Editor: *Instruction: Modify 12.11.2.6.3 as follows:***

* (Re)Association Response for FILS key confirmation

The (#11be)FILSR constructs a (Re)Association Response frame for FILS authentication per 9.3.3.6 (Association Response frame format) and 9.3.3.8 (Reassociation Response frame format). As with the (Re)Association Request frame, hash algorithms are used to generate the FILS Key Confirmation element and the specific hash algorithm depends on the negotiated AKM (see 9.4.2.23.3 (AKM suites)).

The (#11be)FILSR constructs a Key Delivery element indicating the current GTK and GTK PN, and the current IGTK and IPN if management frame protection is enabled, and the current BIGTK and BIPN if beacon protection is enabled, and the current WIGTK and WIPN if WUR frame protection is enabled, and the current CIGTK and CIPN if control frame protection is negotiated. For non-MLO, the GTK is carried in a GTK KDE. The IGTK and IPN are carried in an IGTK KDE, the BIGTK and BIPN are carried in a BIGTK KDE, and the WIGTK and WIPN are carried in a WIGTK KDE. For MLO, the GTKs for all setup links are carried in MLO GTK KDEs, the IGTKs in MLO IGTK KDEs, the BIGTKs in MLO BIGTK KDEs, and the CIGTKs in MLO CIGTK KDEs. The FILSR puts this element into the (Re)Association Response frame.

(…existing texts…)

**TGmf Editor: *Instruction: Modify 13.2.2 as follows:***

* + 1. Authenticator key holders

(…existing texts…)

The R1KH shall meet the following requirements:

* The R1KH-ID shall be set to a MAC address of the physical entity that stores the PMK-R1 and uses it to generate the PTK. That same MAC address shall be used to advertise the PMK-R1 identity to the STA (#11be)or non-AP MLD and the R0KH.
* (#11be)For non-MLO, the R1KH shall derive and distribute the GTK and IGTK to all connected STAs. For MLO, the R1KH shall distribute the GTKs and IGTKs for setup links to all connected non-AP MLDs.
* If WUR frame protection is enabled, the R1KH shall derive and distribute the WIGTK and WIPN to all WUR non-AP STAs with which the R1KH has negotiated WUR frame protection.
* For non-MLO, if control frame protection is negotiated, the R1KH shall derive and distribute the CIGTK and CIPN to all non-AP STAs with which the R1KH has negotiated control frame protection.(#M7) For MLO, the R1KH shall derive and distribute the CIGTKs and CIPNs for setup links to all connected non-AP MLDs with which the R1KH has negotiated control frame protection.
* (#11be)For non-MLO, if beacon protection is enabled, the R1KH shall derive and distribute the BIGTK and BIPN to all connected STAs. For MLO, the R1KH shall derive and distribute the BIGTKs and BIPNs for setup links to all connected non-AP MLDs.
* When the PMK-R1 lifetime expires, the R1KH shall delete the PMK-R1 PMKSA and shall revoke all PTKSAs derived from the PMK-R1 using the MLME-DELETEKEYS primitive.
* The R1KH shall not expose the PMK-R1 to other parties.

(…existing texts…)

**TGmf Editor: *Instruction: Modify 13.4.2 as follows:***

* FT initial mobility domain association in an RSN

(…existing texts…)

(#11be)Between a STA and an AP, the FT 4-way handshake is as follows:

R1KH®S1KH: EAPOL-Key(0, 0, 1, 0, P, 0, 0, ANonce, 0, {})

S1KH®R1KH: EAPOL-Key(0, 1, 0, 0, P, 0, 0, SNonce, MIC, {RSNE(PMKR1Name) [, RSNXE], MDE, FTE [, Device ID](#11bh)})

R1KH®S1KH: EAPOL-Key(1, 1, 1, 1, P, 0, 0, ANonce, MIC, {RSNE(PMKR1Name) [, RSNXE], [, OCI], MDE, FTE, TIE(ReassociationDeadline), TIE(KeyLifetime), GTK(N) [, IGTK(M, IPN)] [, BIGTK(Q, BIPN)] [, WIGTK(R, WIPN)] [, CIGTK(S, CIPN)] [, Device ID] [, IRM] [, PASN ID](#11bh)})

S1KH®R1KH: EAPOL-Key(1, 1, 0, 0, P, 0, 0, 0, MIC, {[IRM](#11bh)})

(#11be)Between a non-AP MLD and an AP MLD, the FT 4-way handshake is as follows:

R1KH®S1KH: EAPOL-Key(0, 0, 1, 0, P, 0, 0, ANonce, 0, {MAC Address})

S1KH®R1KH: EAPOL-Key(0, 1, 0, 0, P, 0, 0, SNonce, MIC, {RSNE(PMKR1Name), MDE, FTE, [RSNXE] [, OCI], MAC Address, MLO Linkn})

R1KH®S1KH: EAPOL-Key(1, 1, 1, 1, P, 0, 0, ANonce, MIC, {MAC Address, MLO Linkm(RSNE(PMKR1Name)), MLO GTKn [, MLO IGTKn] [, MLO BIGTKn] [, MLO IGTKn], MDE, FTE, TIE(ReassociationDeadline), TIE(KeyLifetime) [, WIGTK(R, WIPN)]})

S1KH®R1KH: EAPOL-Key(1, 1, 0, 0, P, 0, 0, 0, MIC, {[MAC Address]})

where MLO GTKn, MLO IGTKn, MLO BIGTKn,  and MLO CIGTKn are defined in 12.7.4 (EAPOL-Key PDU notation).

(…existing texts…)

**13.5.2 Over-the-air FT protocol authentication in an RSN**

**TGmf Editor: *Instruction: Change Figure 13-5 as follows:***

******

**Figure 13-5—Over-the-air FT protocol in an RSN**

**13.5.3 Over-the-DS FT protocol in an RSN**

**TGmf Editor: *Instruction: Change Figure 13-6 as follows:***

******

**Figure 13-6—Over-the-DS FT protocol in an RSN**

**13.6.2 Over-the-air fast BSS transition with resource request**

**TGmf Editor: *Instruction: Change Figure 13-10 as follows:***

****

**Figure 13-10—Over-the-air FT resource request protocol in an RSN**

**13.6.3 Over-the-DS fast BSS transition with resource request**

**TGmf Editor: *Instruction: Change Figure 13-12 as follows:***

****

**Figure 13-12—Over-the-DS FT resource request protocol in an RSN**

**TGmf Editor: *Instruction: Modify 13.7.1 as follows:***

* FT reassociation in an RSN

If the FTO does not send a Reassociation Request frame to the target (#11be)FTR within the reassociation deadline interval received during the FT initial mobility domain association, the target FTR may delete the PTKSA, and the FTO shall abandon this transition attempt.

The FTO shall perform a reassociation directly with the target (#11be)FTR via the following exchange:

FTO®Target FTR: Reassociation Request(RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Request, RSNXE, Basic Multi-Link element)

Target FTR®FTO: Reassociation Response(RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID, GTK[N], IGTK[M], BIGTK[Q], WIGTK[R], CIGTK[S], MLO GTKn, MLO IGTKn, MLO BIGTKn, MLO CIGTKn(#M7)], RIC‑Response, RSNXE, Basic Multi-Link element)

where

* MLO GTK is the MLO GTK subelement for the AP affiliated with the AP MLD for the link specified by the value in the Link ID field,
* MLO IGTK is the MLO IGTK subelement for the AP affiliated with the AP MLD for the link specified by the value in the Link ID field,
* MLO BIGTK is the MLO BIGTK subelement for the AP affiliated with the AP MLD for the link specified by the value in the Link ID field.
* MLO CIGTK is the MLO CIGTK subelement for the AP affiliated with the AP MLD for the link specified by the value in the Link ID field.
* The GTK[N], IGTK[M], and BIGTK[Q] are present when the FTR is an AP.
* The MLO GTKn, MLO IGTKn, MLO BIGTKn, and the Basic Multi-Link element are present when the FTR is an AP MLD.
* The MLO CIGTKn are present when the FTR is an AP MLD and control frame protection is negotiated.

(…existing texts…)

**TGmf Editor: *Instruction: Modify 13.8.5 as follows:***

* FT authentication sequence: contents of fourth message

(…existing texts…)

If present, the FTE shall be set as follows:

* ANonce, SNonce, R0KH-ID, and R1KH-ID shall be set to the values contained in the second message of this sequence.
* The Element Count subfield of the MIC Control field shall be set to the number of elements protected in this frame (variable).
* The RSNXE Used subfield of the MIC Control field shall be set to 1 if the target AP (#11be)or an AP affiliated with the target AP MLD includes an RSNXE in its Beacon and Probe Response frames; otherwise, this subfield shall be set to 0.
* When the negotiated AKM is 00-0F-AC:25, the MIC Length subfield of the MIC Control field shall be set to indicate the length of the MIC field.
* If dot11RSNAOperatingChannelValidationActivated is true and Supplicant indicates OCVC, the Authenticator shall include FT OCI subelement in FTE.
* When this message of the authentication sequence appears in a Reassociation Response frame, the Optional Parameter(s) field in the FTE may include the GTK, IGTK, BIGTK, WIGTK, CIGTK(#M7) subelements (#11be)or MLO GTK, MLO IGTK, MLO BIGTK, and MLO CIGTK subelements. If a GTK, an IGTK, a BIGTK, a WIGTK, a CIGTK,(#M7, #Ed), an MLO GTK, an MLO IGTK, an MLO BIGTK, or an MLO CIGTK are included, the Key field of the subelement shall be wrapped using PTK-KEK or KEK2 and the appropriate key wrap algorithm, as specified in Table 12-11 (Integrity and key wrap algorithms) and 12.7.2 (EAPOL-Key frames). The padding consists of appending a single octet 0xdd followed by zero or more 0x00 octets. When processing a received message, the receiver shall ignore this trailing padding. Addition of padding does not change the value of the Key Length field. Note that the length of the encrypted Key field can be determined from the length of the GTK, IGTK, BIGTK, WIGTK, CIGTK(#M7), MLO GTK, MLO IGTK, MLO BIGTK, or MLO CIGTK subelement.

(…existing texts…)