### IEEE P802.11Wireless LANs

|  |
| --- |
| Trigger, BA, BAR Protection |
| Date: 2025-03-05 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Po-Kai Huang | Intel |  |  |  |
| Ido Ouzieli |  |  |  |
| Danny Alexander |  |  |  |
| Danny Ben-Ari |  |  |  |
| Johannes Berg |  |  |  |
| Laurent Cariou |  |  |  |
| Alfred Asterjadhi | Qualcomm |  |  |  |
| Abhisehek Patil |  |  |  |
| Liwen Chu | NXP |  |  |  |
| Li-Hsiang Sun | Mediatek |  |  |  |
| Nehru Bhandaru | Broadcom |  |  |  |
| Yanjun Sun  | Apple |  |  |  |
| Jarkko Kneckt |  |  |  |

Abstract

This document proposes Trigger, BA, BAR protection.

Revisions:

* Rev 0: Initial version of the document.
* Rev 1: Fix wrong reference.
* Rev 2: Adding missing parts based on the suggestion from Jouni.
* Rev 3: Fix Figure reference. Editorial fix.
* Rev 4: Editorial fix.
* Rev 5: Editorial fix. Reorder clause based on suggestion from Stephen. Clarification revision. Simplify padding indication to one value.
* Rev 6: Revision based on the feedback. Changes are marked with green.
	+ Add missing parts in 6.5.16.1.2, 6.5.17.2.2, 11.xx Group addressed control frame protection procedures, 12.2.4, 12.2.x Requirements for control frame protection, 12.6.xx Protection of Control frames
	+ Improve language in 9.3.1.22.1, 12.5.x.1, 12.5.x.4 Replay counters and packet numbers, 12.8.x Mapping CIGTK to CIP key
	+ Move texts in 12.5.x.2 to 12.6.xx Protection of Control frames
	+ Clarify size of Trigger dependent User Info field for User Info field carrying PN and MIC in 9.3.1.22.1.
* Rev 7: Editorial change based on the comment from Mark. Changes are marked with blue.

**Discussion:**

Trigger, BA, BAR protection has been discussed in 802.11 [1-19]. The main idea is to address the attacks based on

* unprotected Trigger frame to wake up due to SM power save and to have unnecessary solicited transmission
* BlockAck frame to move the transmission window
* BlockAckReq frame to move the received reordering buffer

Revme D7.0 has PBAC to address BAR attacks, but the capability disallows BlockAckReq frame to move window and needs to rely on additional Management frame to move the window, which loses the original benefits of BlockAckReq frame to move window and solicit acknowledgement at the same time.

In this document, the proposal is to protect the following Control frame by introducing capability bit in RSNXE and Protected Control bit, Key ID field, PN, field and MIC field to the corresponding Control frame.

* Individually and group addressed Trigger frames
* Individually and group addressed Multi-STA BlockAck frames
* Individually addressed Compressed BlockAckReq frames
* Individually addressed Multi-TID BlockAckReq frames

GMAC-256 will be used to reduce the number of modes. Mandate to use Multi-STA BlockAck when control frame protection is negotiated. to reduce the number of modes. GCR can be done in GCR MU BAR to reduce the number of modes.

**Interpretation of a Motion to Adopt**

A motion to approve this submission means that the editing instructions and any changed or added material are actioned in the TGmf Draft. This introduction is not part of the adopted material.

***Editing instructions formatted like this are intended to be copied into the TGmf Draft (i.e. they are instructions to the 802.11 editor on how to merge the text with the baseline documents).***

***TGmf Editor: Editing instructions preceded by “TGmf Editor” are instructions to the TGmf editor to modify existing material in the TGmf draft. As a result of adopting the changes, the TGmf editor will execute the instructions rather than copy them to the TGmf Draft.***

***TGmf editor: Add new acronyms in 3.4 as follows:***

**3.4 Acronyms and abbreviations**

CIGTK control integrity group temporal key

CIP control integrity protocol

CIPN CIGTK packet number

***TGmf editor: Modify 6.5.14 as follows:***

* + 1. **SetKeys**
			1. **MLME-SETKEYS.request**
				1. **Semantics of the service primitive**

The primitive parameter is as follows:

MLME-SETKEYS.request(

Keylist
)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Valid range** | **Description** |
| Keylist | A set of SetKeyDescriptors | N/A | The list of keys to be used by the MAC. |

Each SetKeyDescriptor consists of the following parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Valid range** | **Description** |
| Key | Bit string | N/A | The temporal key value |
| Length | Integer | N/A | The number of bits in the Key to be used. |
| Key ID | Integer | 0–3 shall be used with (#3056)TKIP, CCMP, and GCMP; 4–5 with BIP for IGTK; 6-7 with BIP for BIGTK; (11ba)8–9 with BIP for WIGTK; 0-1 with CIP for CIGTK;and 10–4095 are reserved | Key identifier |
| Key Type | Enumeration | Group, Pairwise, PeerKey, IGTK, BIGTK, WIGTK(11ba), CIGTK | Defines whether this key is a GTK, TK, TPK-TK, IGTK, BIGTK, WIGTK, or CIGTK respectively.(11ba)(#1521) |
| Address | MAC address | (#1653)Any valid -individual address | This parameter is valid only when the Key Type value is one of:(#1392) * Pairwise,
* Group and the STA is in an IBSS or PBSS (but not an MBSS),
* PeerKey.
 |
| Receive Sequence Counter(#1406) | 8 octets | N/A | (#1406)Initialization value of the replay counter(s).This parameter is valid only when the Key Type is Group, IGTK, BIGTK, WIGTK, or CIGTK.(11ba) |

* + - * 1. **Effect of receipt**

When the Key Type is Group, IGTK, BIGTK, WIGTK(11ba) or CIGTK, and the key matches the GTK, IGTK, BIGTK, WIGTK(11ba), CIGTK, if any, installed as a result of (#1836)EAPOL-Key PDUs (see 12.7.7.4 (Group key handshake implementation considerations)) or exiting WNM sleep mode (see 11.2.3.15.1 (WNM sleep mode capability)) receipt of this primitive shall have no effect except updating the RSC(s) when they are greater than those currently stored. (#3407)Otherwise, irrespective of the Key Type parameter, when the Key parameter is the same as a key installed as a result of EAPOL-Key PDUs or exiting WNM sleep mode, receipt of this primitive shall have no effect.(#1679) Otherwise, receipt of this primitive causes the MAC to apply the keys as follows, subject to the MLME-SETPROTECTION.request primitive:

* The MAC uses the key and key ID for the transmission of subsequent frames to which the key and key ID apply (as defined by the Key Type and Address parameters).(#3711)(#4332)
* NOTE—If more than one key (and key ID) can be used for transmission for a particular key type and address, the MAC chooses the key and key ID to use.(#4332)
* The MAC installs the key with the associated (#3493)key ID such that received frames for that cipher, of the appropriate type, and containing the matching (#3493)key ID are processed using that key and its associated state information(#1661).
* When the Key Type parameter is Pairwise or PeerKey, and the Key, Key ID, and Address (where valid) parameters identify a new key to be set, the MAC shall initialize the transmitter TSC/PN counter and the receiver replay counter(s) to 0. When the Key Type parameter is not Pairwise, PeerKey, or BIGTK,(#6054) and the Key, Key ID, and Address (where valid) parameters identify a new key to be set, the MAC shall initialize, depending on the direction of the traffic, the transmitter TSC/PN/IPN/WIPN/CIPN(#6054) counter to 0 or 1 (see Clause 12 (Security) and Clause 29 (Wake-Up Radio (WUR) MAC specification(11ba))) or the receiver replay counter(s) to the value in the Receive Sequence Count parameter. (#6054)When the Key Type parameter is BIGTK, and the Key and Key ID parameters identify a new key to be set, the MAC shall initialize, depending on the direction of the traffic, the transmitter BIPN counter as specified in 12.5.3.4 (BIP replay counters and packet numbers(#3573)) or the receiver replay counter to the value in the Receive Sequence Count parameter. When the Key Type, Key, Key ID, and Address (where valid) parameters identify an existing key, the MAC shall not change the transmitter TSC/PN/IPN/BIPN/WIPN/CIPN counter or the receiver replay counter(s) associated with that key.(#1661)(#1505)

(#3292)NOTE—A new key is identified by virtue of having a different Key parameter than any of the currently configured keys used by the MAC. Additionally, it ought to have a different Key parameter than any key that has ever been used by the MAC for the current link.(#4272)

***TGmf editor: Modify 6.5.15 as follows:***

* + - * 1. **When generated**

This primitive is generated by the SME at any time when keys for a security association are to be deleted in the MAC.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Valid range** | **Description** |
| Key ID | Integer | 0–3 shall be usedwith (#3056)TKIP,CCMP, andGCMP;4–5 with BIP for IGTK; 6-7 with BIP for BIGTK; (11ba)8–9 with BIP for WIGTK; 0-1 with CIP for CIGTK; and10–4095 are reserved | Key identifier. |
| Key Type | Enumeration | Group, Pairwise, PeerKey, IGTK, BIGTK, WIGTK(11ba), CIGTK | Defines whether this key is a GTK, TK, TPK-TK, IGTK, BIGTK, WIGTK, or CIGTK respectively.(11ba)(#1521) |
| Address | MAC address | (#1653)Any valid individual address | This parameter is valid only when the Key Type value is one of:(#1392) * Pairwise,
* Group and the STA is in an IBSS or PBSS (but not an MBSS),
* PeerKey.
 |
| Encapsulation Mode(#6054) | Enumeration | Normal, BCE | This parameter is valid only when the Key Type value is BIGTK |

***TGmf editor: Modify 6.5.16.1.2 as follows: (Track change on)***

* Semantics of the service primitive

The primitive parameter is as follows:

MLME-SETPROTECTION.request(

Protectlist
)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid range | Description |
| Protectlist | A set of ProtectDescriptors | N/A | The list of how each key is being used -currently. |

Each ProtectDescriptor consists of the following parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid range | Description |
| Address | MAC address | (#1653)Any valid individual address | This parameter is valid only when the Key Type value is one of:(#1392) * Pairwise,
* Group and the STA is in an IBSS or PBSS (but not an MBSS),
* PeerKey.
 |
| ProtectType | Enumeration | None, Rx, Tx, Rx\_Tx | The protection value for this MAC. |
| Key Type | Enumeration | Group, Pairwise, PeerKey, IGTK, BIGTK, WIGTK(11ba), CIGTK | Defines whether this key is a GTK, TK, TPK-TK, IGTK, BIGTK, WIGTK, or CIGTK respectively.(11ba)(#1521) |

***TGmf editor: Modify 6.5.17.2.2 as follows: (Track change on)***

* Semantics of the service primitive

The primitive parameters are as follows:

MLME-PN-EXHAUSTION.indication(

Key ID,

Key Type,

Address

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid range | Description |
| Key ID | Integer | 0-3 shall be used with CCMP and GCMP; 4-5 with BIP for IGTK; 6-7 with BIP for BIGTK; (11ba)8–9 with BIP for WIGTK; 0-1 with CIP for CIGTK; and 10-4095 are reserved | Key identifier. |
| Key Type | Enumeration | Group, Pairwise, PeerKey, IGTK, BITGK, WIGTK(11ba), CIGTK | Defines whether this key is a GTK, TK, TPK-TK, IGTK, BIGTK, WIGTK, or CIGTK respectively.(11ba)(#1521) |
| Address | MAC address | (#1653)Any valid individual address | This parameter is valid only when the Key Type value is one of:(#1392) * Pairwise,
* Group and the STA is in an IBSS or PBSS (but not an MBSS),
* PeerKey.
 |

***TGmf editor: Modify 9.3.1.7.1 as follows: (Track change on)***

* BlockAckReq frame format
* Overview

The frame format of the BlockAckReq frame is defined in Figure 9-46 (BlockAckReq frame format).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Frame Control | Duration | RA | TA | BAR Control | BAR Information | Control MIC | Padding | FCS |
| Octets: | 2 | 2 | 6 | 6 | 2 | variable | 0 or 22 | variable | 4 |
| * BlockAckReq frame format
 |

(…existing texts….)

The BAR (for block acknowledgment request) Control field is shown in Figure 9-47 (BAR Control field format(11ax)).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | B0 | B1 B4 | B5 | B6 | B7 B11 | B12 B15 |
|  | Reserved | BAR Type | Protected Control | Key ID | Reserved | TID\_INFO |
| Bits: | 1 | 4 | 1 | 1 | 5 | 4 |
|  |  | * BAR Control field format(11ax)
 |

 (11ax)The BAR Type subfield indicates the BlockAckReq frame variant, as defined in Table 9-36 (BlockAckReq frame variant encoding(11ax)).

|  |
| --- |
| * BlockAckReq frame variant encoding(11ax)
 |
| BAR Type  | BlockAckReq frame variant |
| 0 | Reserved |
| 1 | Extended Compressed |
| 2 | Compressed |
| 3 | Multi-TID |
| 4–5 | Reserved |
| 6 | GCR |
| 7–9 | Reserved |
| 10 | GLK-GCR |
| 11–15 | Reserved |

DMG STAs use only the Compressed BlockAckReq variant and the Extended Compressed BlockAckReq variant.

If control frame protection is negotiated, the Protected Control field is equal to 1 if the BlockAckReq frame is a Compressed BlockAckReq or a Multi-TID BlockAckReq frame and contains information that has been processed with a message integrity check algorithm as defined in 12.5.X (Control integrity protocol (CIP) and is equal to 0 if the BlockAckReq frame is a Compressed BlockAckReq or a Multi-TID BlockAckReq frame and does not contain information that has been processed with a message integrity check algorithm. Otherwise, the Protected Control field is reserved.

The Key ID field contains the key ID when the Protected Control field is 1. Otherwise, the Key ID field is reserved.

The meaning of the TID\_INFO subfield of the BAR Control field depends on the BlockAckReq frame variant type. The meaning of this subfield is explained within the subclause for each of the BlockAckReq frame variants.

The meaning of the BAR Information field of the BlockAckReq frame depends on the BlockAckReq frame variant type. The meaning of this field is explained within the subclause for each of the BlockAckReq frame variants.

NOTE—Reference to “a BlockAckReq” frame without any other qualification from other subclauses applies to any of the variants, unless specific exclusions are called out.

The Control MIC field provides integrity protection for the BlockAckReq frame. The Control MIC field is present if the Protected Control field is equal to 1; otherwise, the Control MIC field is not present.

The format of the Control MIC field is shown in Figure 9-XY (Control MIC field format).

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

Figure 9-XY----Control MIC field format

The PN field contains the PN corresponding to the integrity key (see 12.5.x (Control ~~frame~~ integrity protocol (CIP)) indicated by the Key ID field. The PN field format is the same as defined in Figure 9-1029 (PN field format).

The MIC field contains a message integrity check calculated over the BlockAckReq frame as defined in 12.5.x (Control integrity protocol (CIP)).

The Padding field is optionally present if the Protected Control ~~sub~~field is equal to 1; otherwise, the Padding field is not present. The Padding field extends the frame length to give the recipient STA enough time to perform a message integrity check and to prepare the response for transmission a SIFS after the frame is received and validated.

***TGmf editor: Modify 9.3.1.8.1 as follows: (Track change on)***

**9.3.1.8.1 Overview**

(…existing texts….)

The BA Control field is defined in Figure 9-53 (BA Control field format(11ax)(11ay)).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | B0 | B1 B4 | B5 | B6 | B7 B8 | B9 | B10 | B11 | B12 B15 |
|  | Reserved | BA Type | Protected Control | Key ID | Reserved | No Memory Kept | Memory Configuration Tag | Management Ack | TID\_INFO |
| Bits:  | 1 | 4 | 1 | 1 | 2 | 1 | 1 | 1 | 4 |
|  |  | * BA Control field format(11ax)(11ay)
 |

The BA Type subfield in the BA Control field indicates the BlockAck frame variant, as defined in Table 9-37 (BlockAck frame variant encoding)).

|  |
| --- |
| * BlockAck frame variant encoding
 |
| BA Type | BlockAck frame variant |
| 0 | Reserved |
| 1 | Extended Compressed |
| 2 | Compressed |
| 3 | Reserved |
| 4–5 | Reserved |
| 6 | GCR |
| 7 | EDMG Multi-TID |
| 8 | EDMG Compressed |
| 9 | Reserved |
| 10 | GLK-GCR |
| 11 | Multi-STA |
| 12–15 | Reserved |

NOTE—Reference to “a BlockAck” frame without any other qualification from other subclauses applies to any of the variants, unless specific exclusions are called out.

The GCR BlockAck frame is used in response to a GCR BlockAckReq frame, and the GLK-GCR BlockAck frame is used in response to a GLK-GCR BlockAckReq frame.(11ax)

If control frame protection is negotiated with the recipient(s), the Protected Control field is equal to 1 in a Multi-STA BlockAck frame to indicate that the frame is protected and is equal to 0 in a Multi-STA BlockAck frame to indicate that the frame is not protected. Otherwise, the Protected Control field is reserved.

The Key ID field contains the key ID when the Protected Control field is 1. Otherwise, the Key ID field is reserved.

……

***TGmf editor: Modify 9.3.1.8.6 as follows: (Track change on)***

**9.3.1.8.6 Multi-STA BlockAck variant**

……

If the AID11 subfield of the AID TID Info subfield is not 2045, 2009, or 2047, then the Per AID TID Info subfield has the format shown in Figure 9-60 (Per AID TID Info subfield format if the AID11 subfield is not 2045(11ax), 2009, or 2047).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | 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 not 2045(11ax), 2009, or 2047
 |

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-XX (Per AID TID Info subfield format if the AID11 subfield is equal to 2009). 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 in Table 9-40 (Fragment Number subfield encoding for the Multi-STA BlockAck variant) for the length of the Block Ack Bitmap subfield to indicate the length of the PN And MIC ~~sub~~field as described in Figure 9-XX (Per AID TID Info subfield format if the AID11 subfield is equal to 2009).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | AID TID Info | Block Ack Starting Sequence Control | PN And MIC |
| Octets: | 2 | 2 | 32 |

Figure 9-XX----Per AID TID Info subfield format if the AID11 subfield is equal to 2009

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

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

Figure 9-XX----PN And MIC subfield format

The PN field contains the PN corresponding to the integrity key (see 12.5.x (Control ~~frame~~ integrity protocol (CIP)) indicated by the Key ID field. The PN subfield format is the same as defined in Figure 9-1029 (PN field format).

The MIC field contains a message integrity check calculated over the BlockAck frame as defined in 12.5.x (Control integrity protocol (CIP)).

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-XX (Per AID TID Info subfield format if the AID11 subfield is equal to 2047). 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 9-39) is reserved and the Fragment Number subfield of the Block Ack Starting Sequence Control subfield (if present as defined in Table 9-39) is set as defined in Table 9-40 (Fragment Number subfield encoding for the Multi-STA BlockAck variant) for the length of the Block Ack Bitmap subfield to indicate the length of the Padding ~~sub~~field.

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

Figure 9-XX----Per AID TID Info subfield format if the AID11 subfield is equal to 2047

If the AID11 subfield is not 2045, then the context and the presence of each optional subfield in a Per AID TID Info subfield in a Multi-STA BlockAck frame is defined in Table 9-39 (Context of the Per AID TID Info subfield and presence of optional subfields if the AID11 subfield is not 2045(11ax)).

|  |
| --- |
| * Context of the Per AID TID Info subfield and presence of optional subfields if the AID11 subfield is not 2045(11ax)
 |
| Ack Type subfield values | TID subfield values | Presence of Block Ack Starting Sequence Control subfield and Block Ack Bitmap/PN And MIC/Padding (sub)fields | Context of a Per AID TID Info subfield in a Multi-STA BlockAck frame |
| 0 | 0 | Present  | Block acknowledgment context:Sent as an acknowledgment to QoS Data frames that solicit a BlockAck frame response or to a BlockAckReq frame if AID11 subfield is not 2009 or 2047.PN and MIC context if AID11 subfield is equal to 2009.Padding context if AID11 subfield is equal to 2047. |
| 0 | 1–7 | Present | Block acknowledgment context:Sent as an acknowledgment to QoS Data frames that solicit a BlockAck frame response or to a BlockAckReq frame. |
| 1 | 0 | Not present | Acknowledgment context:Sent as an acknowledgment to a QoS Data or QoS Null frame that solicits an Ack frame response if AID11 subfield is not equal to 2047.Padding context if AID11 subfield is equal to 2047. |
| 1 | 1–7 | Not present | Acknowledgment context:Sent as an acknowledgment to a QoS Data or QoS Null frame that solicits an Ack frame response.  |
| 0 or 1 | 8–13 | N/A | Reserved |
| 0 | 14 | N/A | Reserved |
| ~~0~~ | ~~0~~ | ~~Present~~ | ~~PN and MIC context if AID11 subfield is equal to 2009.~~~~Padding context if AID11 subfield is equal to 2047.~~ |
|  |  |  |  |
| 1 | 14 | Not present | All ack context:Sent as an acknowledgment to an A-MPDU that contains an MPDU that solicits an immediate response and all MPDUs contained in the A-MPDU are received successfully. |
| 0 | 15 | N/A | Reserved |
| 1 | 15 | Not present | Management/PS-Poll frame acknowledgment context:Sent as an acknowledgment to a Management or PS-Poll frame. |
| NOTE 1—Additional rules for acknowledgment, block acknowledgment and the all ack context are defined in 26.4.2 (Acknowledgment context in a Multi-STA BlockAck frame) for a multi-TID A-MPDU.NOTE 2—As HE STAs do not use HCCA (see 10.23.1 (General)), TID values from 8 to 15 are not used in QoS Data frames. |

……

***TGmf editor: Modify 9.3.1.22.1 as follows: (Track change on)***

* **Trigger frame format(11ax)**
* **General**

A Trigger frame allocates resources for and solicits one or more HE TB PPDU transmissions. The Trigger frame also carries other information required by the responding STA to send an HE TB PPDU.

The format for the Trigger frame is defined in Figure 9-90 (Trigger frame format(11ax)(#1097)).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | Frame Control | Duration | RA | TA | Common Info | User Info List | Padding | FCS |
| Octets: | 2 | 2 | 6 | 6 | 8 or more | variable | variable | 4 |
|  |

The Duration field is set as defined in 9.2.5 (Duration/ID field (QoS STA)).

The RA field is set as follows:

* For a Trigger frame that is not a GCR MU-BAR, NFRP or MU-RTS Trigger frame, and that has one User Info field and the AID12 subfield of the User Info field contains the AID of a non-AP STA, the RA field is set to the address of that STA
* For a Trigger frame that has at least one User Info field with the AID12 subfield that allocates an RA-RU, the RA field is set to the broadcast address
* For a Trigger frame that is not a GCR MU-BAR Trigger frame and that has more than one User Info field, the RA field is set to the broadcast address
* For a Trigger frame that is an NFRP Trigger frame or MU-RTS Trigger frame, the RA field is set to the broadcast address
* For a Trigger frame that is a GCR MU-BAR Trigger frame, the RA field is set to the MAC address of the group for which reception status is being requested

The TA field is the address of the STA transmitting the Trigger frame if the Trigger frame is addressed to STAs that belong to a single BSS. The TA field is the transmitted BSSID if the Trigger frame is addressed to STAs from at least two different BSSs of the multiple BSSID set. The rules for setting of the TA field are defined in 26.5.2.2.4 (Allowed settings of the Trigger frame fields and TRS Control subfield).

The Common Info field is defined in Figure 9-91 (Common Info field format(11ax)).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | B0    B3 | B4   B15 | B16 | B17 | B18 B19 | B20   B21 | B22 | B23                B25 |
|  | Trigger Type | UL Length | More TF | CS Required | UL BW | GI And HE-LTF Type | MU-MIMO HE-LTF Mode | Number Of HE-LTF Symbols And Midamble Periodicity |
| Bits: | 4 | 12 | 1 | 1 | 2 | 2 | 1 | 3 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | B26 | B27 | B28   B33 | B34   B35 | B36 | B37    B52 | B53 | B54    B62 |
|  | UL STBC | LDPC Extra Symbol Segment | AP Tx Power | Pre-FEC Padding Factor | PEDisambiguity | UL Spatial Reuse | Doppler | UL HE-SIG-A2 Reserved |
| Bits: | 1 | 1 | 6 | 2 | 1 | 16 | 1 | 9 |

|  |  |  |
| --- | --- | --- |
|  | B63 |  |
|  | Reserved | Trigger Dependent Common Info |
| Bits: | 1 | variable |
| * **Common Info field format(11ax)**
 |

The Trigger Type subfield identifies the Trigger frame variant and its encoding is defined in Table 9-47 (Trigger Type subfield encoding(11ax)).

|  |
| --- |
| * **Trigger Type subfield encoding(11ax)**
 |
| **Trigger Type subfield value** | **Trigger frame variant** |
| 0 | Basic |
| 1 | Beamforming Report Poll (BFRP) |
| 2 | MU-BAR |
| 3 | MU-RTS |
| 4 | Buffer Status Report Poll (BSRP) |
| 5 | GCR MU-BAR |
| 6 | Bandwidth Query Report Poll (BQRP) |
| 7 | NDP Feedback Report Poll (NFRP) |
| 8(11az) | Ranging |
| 9–15 | Reserved |

(…existing texts….)

The UL HE-SIG-A2 Reserved subfield of the 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. An HE AP sets the UL HE-SIG-A2 Reserved subfield to all 1s except when control frame protection is negotiated with the recipient(s), where B61 and B62 are set as follows:

~~If control frame protection is negotiated,~~ 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.X (Control ~~frame~~ integrity protocol (CIP) and is equal to 0 otherwise~~, if the Trigger frame is not protected~~. If the Protected Control subfield is equal to 1, B62 of the Common Info field is the Key ID ~~sub~~field. Otherwise, B62 is reserved.

~~When the Protected Control subfield is equal to 1, the Trigger frame is protected using the MIC algorithm as defined in 12.5.X (Control frame integrity protocol (CIP).~~

~~If control frame protection is negotiated and the Protected Control subfield is equal to 1, B62 of the Common Info field is the Key ID subfield. If control frame protection is negotiated but the Protected Control Subfield is equal to 0, B2 is reserved.~~

(…existing texts….)

The Trigger Control MIC field provides integrity protection for the Trigger frame. The Trigger Control MIC field is present if the Protected Control subfield is equal to 1; otherwise, the Trigger Control MIC field is not present.

The Trigger Control MIC field contains 8 User Info fields. The first and the second User Info fields of the Trigger Control MIC field contain the PN corresponding to the integrity key indicated by the Key ID ~~sub~~field, and each User Info field has AID12 subfield equal to 2009. of the first and the second User Info field

~~the PN subfield and the MIC subfield as shown in Figure 9-xxx (Formats of User Info fields with AID12 subfield equal to 2009) and Figure 9-xxx (Formats of User Info fields with AID12 subfield equal to 2010).~~

~~The format of the User Info fields with AID12 subfield equal to 2009 is shown in Figure 9-xxx (Formats of User Info fields with AID12 subfield equal to 2009). The Trigger Dependent User Info field (if present) is equal to 0.~~

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | B0   B11 | B12    B15 | B16 B23 | B24   B31 | B32 B39  |  |
| First User Info field of the Trigger Control MIC field | AID12 | Reserved | PN0 | PN1 | PN2 | TriggerDependentUser Info |
| Bits: | 12 | 4 | 8 | 8 | 8 | variable |
|  | B0   B11 | B12    B15 | B16 B23 | B24   B31 | B32 B39  |  |
| Second User Info field of the Trigger Control MIC field | AID12 | Reserved | PN3 | PN4 | PN5 | TriggerDependentUser Info |
| Bits: | 12 | 4 | 8 | 8 | 8 | variable |

Figure 9-xxx- Formats of User Info fields with AID12 subfield equal to 2009

The third, the fourth, the fifth, the sixth, the seventh, and the eighth User Info field of the Trigger Control MIC field contain a message integrity check calculated over the Trigger frame as defined in 12.5.x (Control integrity protocol (CIP)), and each User Info fields has~~. The MIC field is carried in five contiguous User Info fields, each with~~ AID12 subfield equal to 2010. The format of the third, the fourth, the fifth, the sixth, the seventh, and the eighth User Info fields with AID12 subfield equal to 2010 is shown in Figure 9-xxx (Formats of User Info fields with AID12 subfield equal to 2010). ~~The Trigger Dependent User Info field (if present) is equal to 0.~~

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | B0 B11 | B12  B15 |  B16 B39  |  |  |
| Third User Info field of the Trigger Control MIC field | AID12 | Reserved | MIC[0:23] | TriggerDependentUser Info |
| Bits: | 12 | 4 | 24 | variable |
|  | B0 B11 | B12 B15 |  B16 B39  |  |
| Fourth User Info field of the Trigger Control MIC field | AID12 | Reserved | MIC[24:47] | TriggerDependentUser Info |
| Bits: | 12 | 4 | 24 | variable |
|  | B0 B11 | B12 B15 |  B16 B39  |  |
| Fifth User Info field of the Trigger Control MIC field | AID12 | Reserved | MIC[48:71] | TriggerDependentUser Info |
| Bits: | 12 | 4 | 24 | variable |
|  | B0 B11 | B12 B15 |  B16 B39  |  |
| Sixth User Info field of the Trigger Control MIC field | AID12 | Reserved | MIC[72:95] | TriggerDependentUser Info |
| Bits: | 12 | 4 | 24 | variable |
|  | B0 B11 | B12 B15 |  B16 B39  |  |
| Seventh User Info field of the Trigger Control MIC field | AID12 | Reserved | MIC[96:119] | TriggerDependentUser Info |
| Bits: | 12 | 4 | 24 | variable |
|  | B0 B11 | B12 B15 |  B16 B39  |  |
| Eighth User Info field of the Trigger Control MIC field | AID12 | Reserved | MIC[120:127] | Reserved | TriggerDependentUser Info |
| Bits: | 12 | 4 | 8 | 16 | variable |

Figure 9-xxx- Formats of User Info fields with AID12 subfield equal to 2010

For an MU-BAR Trigger frame, the Trigger Dependent User Info field of User Info fields with AID12 subfield equal to 2009 or 2010 is four octets in length, where the BAR Type field (see Figure 9-98—Trigger Dependent User Info subfield format in the MU-BAR Trigger frame and Figure 9-47—BAR Control field format) is set to indicate the Compressed BlockAckReq frame variant and the other subfields are set to 0 ~~and all the subfields, except for the BAR Type subfield, are set to 0 in an MU-BAR Trigger frame. The BAR Type subfield is set to indicate a Compressed BAR in an MU BAR Trigger frame~~. For other variants, the Trigger Dependent User Info field (if present) is equal to 0.

The Trigger Control MIC field follows all other User Info fields in the Trigger frame that are addressed to STAs that have negotiated control frame protection.

**9.3.3.5. Association Request frame format**

***TGmf editor: Modify Table 9-64 as follows:***

**Table 9-64—Association Request frame body**

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| … |  |  |
| <Last assigned + 1> | CIP Capabilities ~~element~~ | The CIP Capabilities element is present if dot11CIPActivated is true; otherwise, it is not present. |

**9.3.3.6. Association Response frame format**

***TGmf editor: Modify Table 9-65 as follows:***

**Table 9-65—Association Response frame body**

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| … |  |  |
| <Last assigned + 1> | CIP Capabilities ~~element~~ | The CIP Capabilities element is present if dot11CIPActivated is true; otherwise, it is not present. |

**9.3.3.7. Reassociation Request frame format**

***TGmf editor: Modify Table 9-66 as follows:***

**Table 9-66—Reassociation Request frame body**

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| … |  |  |
| <Last assigned + 1> | CIP Capabilities | The CIP Capabilities element is present if dot11CIPActivated is true; otherwise, it is not present. |

* + - 1. **Reassociation Response frame format**

***TGmf editor: Modify Table 9-67 as follows:***

**Table 9-67—Reassociation Response frame body**

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| … |  |  |
| <Last assigned + 1> | CIP Capabilities | The CIP Capabilities element is present if dot11CIPActivated is true; otherwise, it is not present. |

* + - 1. **Probe Response frame format**

***TGmf editor: Modify Table 9-69 as follows:***

**Table 9-69—Probe Response frame body**

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| … |  |  |
| <Last assigned + 1> | CIP Capabilities | The CIP Capabilities element is present if dot11CIPActivated is true; otherwise, it is not present. |

***TGmf editor: Modify table 9-130 as follows:***

**9.4.2 elements**

* General

***Modify Table 9-130 (Element IDs) as follows:***

* Element IDs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element | Element ID | Element ID Extension | Extensible | Fragmentable |
| ..... |  |  |  |  |
| CIP Capabilities ~~element~~ (see 9.4.2.xx (CIP Capabilities element)) | 255 | <ANA> | Yes | No |
| ... |  |  |  |  |
| NOTE 1—See 10.28.6 (Element parsing) on the parsing of elements. |

***TGmf editor: Modify 9.4.2.46 as follows:***

* FTE

***Modify Table 9-221 (not all lines shown) as follows:***

* Subelement IDs

|  |  |
| --- | --- |
| Value | Subelement Name |
| … |  |
| <ANA> | CIGTK |

***Insert the following at the end of 9.4.2.46 (FTE):***

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Subelement ID | Length | Key Info~~ID~~ | CIPN  | Wrapped Key |
| Octets: | 1 | 1 | 1 | 6 | ~~32~~40 |

Figure 9-xxx- (CIGTK subelement format )

The Key Info field is defined in Figure 9-xxx—(CIGTK subelement’s Key Info subfield format).

 B0 B1 B7

|  |  |  |
| --- | --- | --- |
|  | Key ID | Reserved |
| Bits: | 1 | 7 |

Figure 9-xxx—(CIGTK subelement’s Key Info subfield format)

The Key ID field contains the CIGTK key ID. ~~Bits 0 of the Key ID field define a value in the range 0 to 1. Bits 1-7 of the Key ID field are reserved.~~

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

The Wrapped Key field contains the wrapped CIGTK being distributed.

***TGmf editor: Modify 9.4.2.80 as follows:***

* WNM Sleep Mode element

***Modify Table 9-266 (not all lines shown) as follows:***

* WNM Sleep Mode Response Status definition

|  |  |
| --- | --- |
| Value | Description |
| 0 | Enter/Exit WNM sleep mode Accept. |
| 1 | Exit WNM sleep mode Accept, GTK/IGTK/BIGTK/CIGTK update required. |

***TGmf editor: Insert the following new subclauses at the end of 9.4.2:***

**9.4.2.xx CIP Capabilities element**

The CIP Capabilities element contains fields that are used to advertise padding delay used with CIP.

The format of the CIP Capabilities element is shown in Figure 9-xxx (CIP Capabilities element).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Element ID | Length | Element ID Extension | Padding Delay |
| Octets: | 1 | 1 | 1 | 1 |
| Figure 9-xxx - CIP Capabilities element format |

The Element ID, Length and Element ID Extension fields are defined in 9.4.2.1 (General).

The Padding Delay field contains the MIC padding delay. The format of the Padding Delay field is shown in Figure 9-xxxx (Padding Delay field).

|  |  |  |
| --- | --- | --- |
|  | MIC Padding Delay | Reserved |
|  Bits |  4 |  4 |

  **Figure 9-xxx – Padding Delay field**

For a STA that sends the CIP Capabilities element, the MIC Padding Delay field indicates the minimum padding duration that is needed (see 12.5.x.7 (Padding)) within a PPDU that solicits a protected Control frame from the STA and/or the minimum padding duration that is needed within a protected Control frame that is addressed to the STA.

The MIC Padding Delay field is set as defined in Table 9-xxx (Encoding of the MIC Padding Delay field).

Table 9-xxx—Encoding of the MIC Padding Delay field

|  |  |
| --- | --- |
| MIC Padding Delay field value | MIC padding delay |
| 0 | 0 µs |
| 1 | 4 µs |
| 2 | 8 µs |
| 3 | 12 µs |
| 4 | 16 µs |
| 5 | 20 µs |
| 6 | 24 µs |
| 7 | 28 µs |
| 8 | 32 µs |
| ok9-15 | Reserved |

* RSNXE

***TGmf editor: Modify the Table 9-373 as follows:***

* Extended RSN Capabilities field

|  |  |  |
| --- | --- | --- |
| Bit | Information | Notes |
| … |  |  |
| <ANA> | CIP Supported | The CIP Supported field is set to 1 when dot11CIPActivated is true and is set to 0 otherwise.  |

***TGmf editor: Modify 9.6.13.20 as follows:***

* + - 1. **WNM Sleep Mode Response frame format**

(…existing texts….)

The Key Data field contains zero or more subelements that provide the current GTK, IGTK, BIGTK, and CIGTK to the STA. The format of these subelements is shown in Figure 9-1288 (WNM Sleep Mode GTK subelement format), Figure 9-1289 (WNM Sleep Mode IGTK subelement format), and Figure 9-1290 (WNM Sleep Mode BIGTK subelement format).(#155) The subelement IDs for these subelements are defined in Table 9- 540 (Optional subelement IDs for WNM Sleep Mode parameters). When management frame protection was not negotiated for the current association(#7049), the Key Data field is not present.

|  |
| --- |
| **Table 9-540 - Optional subelement IDs for WNM Sleep Mode parameters** |
| **Value** | **Contents of subelement** |
| <ANA> | CIGTK |

(…existing texts….)

The CIGTK subelement ~~contains the CIGTK~~is as shown in Figure 9-xxx (WNM Sleep Mode CIGTK subelement format).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Subelement ID | Length | Key Info~~ID~~ | CIPN | Key |
| Octets: | 1 | 1 | 1 | 6 |  32 |
| **Figure 9-xxx - WNM Sleep Mode 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).

The Key Info field is defined in Figure 9-xxx—(CIGTK subelement’s Key Info subfield format).

 B0 B1 B7

|  |  |  |
| --- | --- | --- |
|  | Key ID | Reserved |
| Bits: | 1 | 7 |

Figure 9-xxx—(WNM Sleep Mode CIGTK subelement’s Key Info subfield format)

The Key ID field contains the CIGTK key ID. ~~Bits 0 of the Key ID field define a value in the range 0 to 1. Bits 1-7 of the Key ID field are reserved.~~

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

The Key field is the CIGTK being distributed.

(…existing texts….)

***TGmf editor: Modify clause 11 as follows:***

* General

***Change the eighth paragraph as follows:***

WNM sleep mode enables an extended power save mode in which a non-AP STA needs not listen for every DTIM beacon, and does not need to perform GTK/IGTK/BIGTK/CIGTK updates. A STA in WNM sleep mode can transition to awake state as infrequently as once every WNM sleep interval to check whether its corresponding TIM bit is set or group addressed traffic is pending.

* WNM sleep mode
* WNM sleep mode capability

***Change the fourth paragraph as follows:***

WNM sleep mode enables an extended power save mode for non-AP STAs in which a non-AP STA need not listen for every DTIM beacon, and need not perform GTK/IGTK/BIGTK/CIGTK updates. A non-AP STA can sleep for extended periods as indicated by the WNM Sleep Interval field of the WNM Sleep Mode element, which is present in WNM Sleep Mode Request frames transmitted by the non-AP STA.

* WNM sleep mode non-AP STA operation

***Change the fifth paragraph as follows:***

The receipt of an MLME-SLEEPMODE.confirm primitive with a valid SleepMode parameter indicates to the STA's SME that the AP has processed the corresponding WNM Sleep Mode Request frame. The content of the WNM sleep mode parameter in the WNM Sleep Mode Response frame provides the status of WNM Sleep Mode elements processed by the AP. The non-AP STA shall delete the GTKSA if the response indicates success. If RSN is used with management frame protection, the non-AP STA shall delete the IGTKSA if the response indicates success, If RSN is used with beacon frame protection, the non-AP STA shall delete the BIGTKSA if the response indicates success. If RSN is used with control frame protection, the non-AP STA shall delete the CIGTKSA if the response indicates success.

* WNM sleep mode AP operation

(…existing texts..)

When WNM sleep mode is enabled for an associated STA, the AP shall stop sending all individually addressed MPDUs to the non-AP STA. The AP may disassociate or deauthenticate the STA at any time for any reason while the non-AP STA is in WNM sleep mode. An AP shall perform the TFS AP operation, as specified in 11.21.12 (TFS procedures) for non-AP STAs for which it has received TFS Request elements. The AP shall set the TIM bit corresponding to the AID of the associated STA for which the AP has queued either a TFS Notify frame or matching frame. An AP shall not perform GTK/IGTK/BIGTK/CIGTK updates for the STAs in WNM sleep mode.

(…existing texts..)

If RSN is used with management frame protection and a valid PTK is configured for the STA, the current GTK, IGTK, BIGTK, and CIGTK shall be included in the WNM Sleep Mode Response frame. If a GTK/IGTK/BIGTK/CIGTK update is in progress, the pending GTK, IGTK, BIGTK, and CIGTK shall be included in the WNM Sleep Mode Response frame.

(…existing texts..)

* Non-AP STA and non-PCP STA association initiation procedures

***Change the first paragraph as follows:***

The SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA(11ba), WIGTKSA, (#3344)WTKSA, CIGTKSA, and TPKSA (including temporal keys)(#205) 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.

* AP or PCP association receipt procedures

***Change the first paragraph as follows (not all lines shown):***

The following procedure shall be used by an AP or PCP upon receipt of an Association Request frame from a STA:

* If the ResultCode in the MLME-ASSOCIATE.response primitive is SUCCESS, the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA, WIGTKSA, WTKSA, CIGTKSA, and TPKSA (including temporal keys) held for communication with the STA by using the MLME-DELETEKEYS.request primitive (see 12.6.16 (RSNA security association termination)).
* Non-AP STA and non-PCP STA reassociation initiation procedures

***Change the first paragraph as follows:***

Except when the association is part of a fast BSS transition, the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA, WIGTKSA, WTKSA, CIGTKSA, and TPKSA (including temporal keys) held for communication with the AP, or PCP by using the MLME-DELETEKEYS.request primitive (see 12.6.16 (RSNA security association termination)) before invoking an MLME-REASSOCIATE.request primitive.

* AP, or PCP reassociation receipt procedures

***Change the first paragraph as follows (not all lines shown):***

The following procedure shall be used by an AP or PCP upon receipt of a Reassociation Request frame from a STA:

* If management frame protection is not in use, or the ResultCode in the MLME-REASSOCIATE.response primitive is SUCCESS and the reassociation is not part of a fast BSS transition, the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA, WIGTKSA, WTKSA, CIGTKSA, and TPKSA (including temporal keys) held for communication with the STA by using the MLME-DELETEKEYS.request primitive (see 12.6.16 (RSNA security association termination)).
* Non-AP STA and non-PCP STA disassociation initiation procedures

***Change the second paragraph as follows (not all lines shown):***

Upon receipt of an MLME-DISASSOCIATE.request primitive, a non-AP STA and non-PCP STA's MLME shall disassociate from an AP, or PCP, respectively, using the following procedure:

* Upon receiving an MLME-DISASSOCIATE.confirm primitive, the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA, WIGTKSA, WTKSA, CIGTKSA, and TPKSA (including temporal keys) held for communication with the AP 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. In the case of an MM-SME coordinated STA, the MLME shall perform this 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.
* Non-AP STA and non-PCP STA disassociation receipt procedure

***Change the first paragraph as follows (not all lines shown):***

Upon receipt of a Disassociation frame from an AP, 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 and non-PCP STA, respectively, shall disassociate from the AP or PCP using the following procedure:

* Upon receiving the MLME-DISASSOCIATE.indication primitive, the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA, WIGTKSA, WTKSA, CIGTKSA, and TPKSA (including temporal keys) held for communication with the AP 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.
* General

***Change the second paragraph as follows (not all lines shown):***

Upon receipt of an MLME-DISASSOCIATE.request primitive, an AP or PCP shall disassociate a STA (with respect to the AP or PCP) using the following procedure:

* Upon receiving an MLME-DISASSOCIATE.confirm primitive, the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA, WIGTKSA, WTKSA, CIGTKSA, and TPKSA (including temporal keys) held for communication with the STA 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.
* AP or PCP disassociation receipt procedure

***Change the first paragraph as follows (not all lines shown):***

Upon receipt of a Disassociation frame from a STA 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, the AP or PCP (with respect to the STA) shall disassociate the STA using the following procedure:

* Upon receiving an MLME-DISASSOCIATE.indication primitive the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA, WIGTKSA, WTKSA, CIGTKSA, and TPKSA (including temporal keys) held for communication with the STA 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.

***TGmf editor: Insert a new clause at the end of clause 11 as follows:***

11.xx Group addressed control frame protection procedures

When control frame protection is negotiated, the MLME shall provide an encapsulation service for group addressed Control frames that are defined to be protected. All group addressed Control frames shall be submitted to this service for encapsulation and transmission.

The group addressed control frame protection service shall take the following actions:

* Control frame protection for broadcast shall be set using the MLME-SETPROTECTION.request primitive with the Protectlist including a Key Type value of CIGTK. A non-AP STA shall also set the Protect Type value to Rx. An AP shall set the Protect Type value to Tx.
* The CIGTK shall be installed using the MLME-SETKEYS.request primitive with the value CIGTK for the Key Type parameter of the SetKeyDescriptor.
* The frames shall be encapsulated and protected using CIP (see 12.5.x (Control integrity protocol (CIP))).

***TGmf editor: Modify 12 as follows:***

* RSNA establishment

An SME establishes an RSNA in one of seven ways:

* If an RSNA uses authentication negotiated over IEEE Std 802.1X or FILS authentication in an infrastructure BSS, an SME establishes an RSNA as follows:

 7) If the STAs negotiate management frame protection, the SME programs the TK and pairwise cipher suite into the MAC for protection of individually addressed robust Management frames. It also installs the IGTK and IPN for protection of group addressed robust Management frames.

8) If beacon protection is enabled, the SME programs the BIGTK and BIPN into the MAC for the protection of Beacon frames.

8a) If control frame protection is negotiated, the SME programs the CIGTK and CIPN into the MAC for the protection of group addressed Control frames that are defined to be protected. The SME also programs the TK into the MAC for the protection of individually addressed Control frames that are defined to be protected.

* If an RSNA is based on a PSK or password in an infrastructure BSS, an SME establishes an RSNA as follows:
1. If the STAs negotiate management frame protection, the SME programs the TK and pairwise cipher suite into the MAC for protection of individually addressed robust Management frames.  It also installs the IGTK and IPN for protection of group addressed robust Management frames.
2. If beacon protection is enabled, the SME programs the BIGTK and BIPN into the MAC for the

 protection of Beacon frames.

 7a) If control frame protection is negotiated, the SME programs the CIGTK and CIPN into the MAC for the protection of group addressed Control frames that are defined to be protected. The SME also programs the TK into the MAC for the protection of individually addressed Control frames that are defined to be protected

* If an RSNA allows for confidentiality only (no authentication) in an infrastructure BSS, an SME establishes an RSNA as follows:

7) If beacon protection is enabled, it programs the BIGTK into the MAC for protection of Beacon frames.

7a) If control frame protection is negotiated, it programs the CIGTK into the MAC for protection of group addressed Control frames that are defined to be protected. It also programs TK into the MAC for protection of individually addressed Control frames that are defined to be protected

***TGmf editor: Add new clause at the end of 12.2 as follows:***

**12.2.x Requirements for control frame protection**

The Control frames that are defined to be protected are:

* Individually and group addressed Trigger frames
* Individually and group addressed Multi-STA BlockAck frames
* Individually addressed Compressed BlockAckReq frames
* Individually addressed Multi-TID BlockAckReq frames

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.xx (Protection of Control frames)).

***TGmf editor: Add new clause at the end of 12.5 as follows:***

**12.5.x Control integrity protocol (CIP)**

**12.5.x.1 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.

**12.5.x.2 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 9.3.1.22 (Trigger frame format), 9.3.1.7 (BlockAckReq frame format), and 9.3.1.8 (BlockAck frame format).

**12.5.x.3 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-xx (CIP AAD construction) depicts the format of the AAD.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Frame Control | Duration | RA | TA |
| Octets: | 2 | 2 | 6 | 6 |

Figure 12-xx—CIP AAD construction

**12.5.x.4 Replay counters and packet numbers**

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

* The non-AP STA and the AP 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.
* 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.
* The AP 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.
* 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.

**12.5.x.5 Transmission**

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

1. 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.
2. Increment the PN, to obtain a fresh nonzero PN
3. Set the Key ID field to the corresponding key ID and set the 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 PN field to form a nonce, i.e., the initialization vector.
4. Compute the AAD as specified in 12.5.x.3 (CIP AAD construction).
5. 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-xxx (Formats of User Info fields with AID12 subfield equal to 2010)) 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.
6. Include padding if needed to satisfy the padding requirement(s) of the intended recipient(s).
7. Transmit the frame.

**12.5.x.6 Reception**

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

1. 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.
2. 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.
3. Concatenate the TA field and the non-negative integer inserted into the PN field, to form a nonce, i.e., the initialization vector
4. Compute the AAD as specified in 12.5.x.3 (CIP AAD construction).
5. 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-xxx (Formats of User Info fields with AID12 subfield equal to 2010)) 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.
6. Update the corresponding replay counter identified by the Key ID field and the RA field with the value of the PN field.

**12.5.x.7 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 (12-x1), which is based on the MIC padding delay indicated by the target STA (see 9.4.2.xx (CIP Capabilities element)), 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-xxx (Formats of User Info fields with AID12 subfield equal to 2010)) if the frame is a Trigger frame

*MPAD,MAC* = *NDBPSMPAD (12-x1)*

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-x1).

*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.xx (CIP Capabilities element)).

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.xx (CIP Capabilities element)), 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 in 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, C Proc is greater than or equal to the MIC padding delay indicated by the target STA (see 9.4.2.xx (CIP Capabilities element)).

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: Modify 12 as follows:***

* General

***Change second paragraph (not all lines shown) by adding a sub-bullet as follows:***

A security association is a set of policy(ies) and key(s) used to protect information. The information in the security association is stored by each party of the security association, needs to be consistent among all parties, and needs to have an identity. The identity is a compact name of the key and other bits of security association information to fit into a table index or an MPDU. The following types of security associations are supported by an RSNA STA:

* BIGTKSA: A result of a successful group key handshake, successful 4-way handshake, successful

 FT 4-way handshake, the Reassociation Response frame of the fast BSS transition protocol, or successful FILS authentication.

* CIGTKSA: A result of a successful group key handshake, successful 4-way handshake, successful

 FT 4-way handshake, the Reassociation Response frame of the fast BSS transition protocol, or successful FILS authentication.

***Insert a new subclause at the end of 12.6.1.1 (Security association definitions) as follows:***

12.6.1.1.x CIGTKSA

An Authenticator's SME creates a CIGTKSA when control frame protection is negotiated for the first time. A CIGTKSA has the 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 CIGTKSA consists of the following:

* Direction vector (whether the CIGTK is used to transmit or receive)
* Key ID
* CIGTK
* Authenticator MAC address

12.6.1.2.2 Security association in an ESS

The MLME-DELETEKEYS.request primitive deletes the temporal key(s) established for the security association so that they cannot be used to protect subsequent IEEE 802.11 traffic. An SME uses this primitive when it deletes a PTKSA, GTKSA, IGTKSA, BIGTKSA, CIGTKSA, WIGTKSA or TPKSA.

* RSNA security association termination

***Change the second paragraph as follows:***

it deletes some security associations. In the case of an ESS, the non-AP STA's SME shall delete any PTKSA(s), GTKSA(s), IGTKSA(s), BIGTKSA(s), CIGTKSA(s), WIGTKSA(s), WTKSA(s), TPKSA(s), and the AP's SME shall delete the PTKSA.

* RSNA rekeying

(…existing texts…)

The IEEE 802.11 MAC shall issue an MLME-PN-WARNING.indication primitive when the packet number assignment for a particular PTKSA, IGTKSA, GTKSA, BIGTKSA, CIGTKSA or TPKSA(#205) reaches or exceeds the threshold that is defined in dot11PNWarningThresholdLow and dot11PNWarningThresholdHigh for the first

time. The indication shall be issued only once for a given PTKSA, IGTKSA, GTKSA, BIGTKSA, CIGTKSA or

TPKSA(#205). The SME may use the indication as a trigger to establish a new PTKSA, IGTKSA, GTKSA, BIGTKSA, CIGTKSA or TPKSA(#205) before the Packet Number space is exhausted.

(…existing texts…)

(#1382)An Authenticator may initiate a group key handshake for the purpose of GTK rekeying (with a GTKSA), IGTK keying (with an IGTKSA), BIGTK rekeying (with a BIGTKSA), WIGTK rekeying (with a WIGTKSA), or CIGTK rekeying (with a CIGTKSA).

***TGmf editor: Add new clause at the end of 12.6 as follows:***

**12.6.xx Protection of Control frames**

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

Control frame protection control frame protection

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.xx (Group addressed control frame protection procedures). 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)).

the the

***TGmf editor: Modify 12 as follows:***

* General

***Change the first paragraph as follows (not all lines shown):***

RSNA defines the following key hierarchies:

* WIGTK, a hierarchy consisting of a single key to provide integrity protection for broadcast and group addressed WUR Wake-up frames

fa) CIGTK, a hierarchy consisting of a single key to provide integrity protection for group addressed Control frames that are defined to be protected

***Insert the following new subclauses at the end of 12.7.1:***

**12.7.1.x Control integrity group temporal key (CIGTK) hierarchy**

The Authenticator shall select the CIGTK as a random value each time it is generated.

The Authenticator may update the CIGTK for any reason, including:

1. The disassociation or deauthentication of a STA.
2. An event within the SME that triggers a group key handshake.

The CIGTK is configured via the MLME-SETKEYS.request primitive; see 6.5.14 (SetKeys). The CIGTK configuration is described in the EAPOL-Key state machines; see 12.7.9 (RSNA Supplicant key management state machine) and 12.7.10 (RSNA Authenticator key management state machine).

The CIGTK packet number (CIPN) is used to provide replay protection.

**12.7.2 EAPOL-Key frames**

***Insert the following new row to Table 12-10 (KDE selectors) while maintaining the numerical order and updating the reserved range:***

**Table 12-10—KDE selectors**

|  |  |  |
| --- | --- | --- |
| **OUI** | **Data type** | **Meaning** |
| 00-0F-AC | <ANA> | CIGTK KDE |
| … | … | … |

***Insert the following:***

The format of the CIGTK KDE is shown in [Figure 12-50x (CIGTK KDE)](#_bookmark22).

|  |  |  |
| --- | --- | --- |
| Key Info | CIPN | CIGTK |

Octets: 1 6 32

**Figure 12-50c—CIGTK KDE**

The Key Info field is defined in Figure 9-xxx—(CIGTK KDE’s Key Info field format).

 B0 B1 B7

|  |  |  |
| --- | --- | --- |
|  | Key ID | Reserved |
| Bits: | 1 | 7 |

Figure 9-xxx—(CIGTK KDE’s Key Info field format)

The Key ID field contains the CIGTK key ID.

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.

The CIGTK field contains the CIGTK.

***Modify 12.7.4 as follows:***

**12.7.4 EAPOL-Key PDU notation**

…

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

…

WIGTK[R] is the (#3493)WIGTK KDE, with the Key ID field set to R

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

CIGTK[S] is the CIGTK KDE, with the key ID field set to S

CIPN is the last CIPN, as provided by the CIGTK KDE

***Modify 12.7.7 as follows:***

**12.7.7 Group key handshake**

**12.7.7 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, and if control frame protection is negotiated, a new CIGTK, 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)] )

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

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](#_bookmark18) [(EAPOL-Key frames)](#_bookmark18) 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)](#_bookmark18) 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)](#_bookmark18) 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)](#_bookmark18) 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)](#_bookmark18) using the PTK-KEK defined in 12.7.1.3 (Pairwise key hierarchy) and associated IV.
* 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).
* OCI KDE represents the current operating channel information using which the EAPOL-Key PDU is sent. OCI KDE is included when dot11RSNAOperatingChannelValidationActivated is true on the STA sending the message.

**12.7.7.2 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)](#_bookmark18)

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.

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 =

* GTK and the GTK’s key ID (see [12.7.2 (EAPOL-Key frames)](#_bookmark18))
* When present, IGTK, IGTK’s key ID, and IPN (see [12.7.2 (EAPOL-](#_bookmark18) [Key frames)](#_bookmark18))
* When present, BIGTK, BIGTK’s key ID, and BIPN (see [12.7.2](#_bookmark18) [(EAPOL-Key frames)](#_bookmark18))
* When present, WIGTK, WIGTK’s key ID, and WIPN (see [12.7.2](#_bookmark18) [(EAPOL-Key frames)](#_bookmark18))
* When present, CIGTK, CIGTK’s key ID, and CIPN (see [12.7.2 (EAPOL-Key frames)](#_bookmark18))
* OCI KDE when dot11RSNAOperatingChannelValidationActivated is true on the Authenticator

…

On reception of message 1, the Supplicant:

d) Uses the MLME-SETKEYS.request primitive to configure the GTK and, the IGTK when present, and the BIGTK if beacon protection is enabled at the non-AP STA, and the WIGTK if WUR frame protection is negotiated, and the CIGTK if control frame protection is negotiated, into the MAC.

***Modify 12.7.9.4 as follows:***

**12.7.9.4 Supplicant state machine procedures**

(…existing texts…)

When processing 4-way handshake message 3, the GTK, IGTK, BIGTK if present,

WIGTK if present, and CIGTK if present are decrypted from the EAPOL-Key PDU and installed. The PTK shall be installed before the GTK and IGTK.

(…existing texts…)

***Insert the following new subclauses at the end of 12.8:***

**12.8.x Mapping CIGTK to CIP key**

See 12.7.1.x (Control integrity group temporal key (CIGTK) hierarchy) for the definition of the CIGTK. A STA shall use ~~bits 0–255 of~~ the CIGTK as the GMAC-256 key for protecting group addressed Control frames.

**12.11.2.6.3 (Re)Association Response for FILS key confirmation**

(..existing texts…)

The AP constructs a Key Delivery element indicating the current GTK and (#1406)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(11ba), and the current WIGTK and WIPN if WUR frame protection is enabled, and the current CIGTK and CIPN if control frame protection is negotiated. The GTK is carried in a GTK KDE(#3492). The IGTK and IPN are carried in an IGTK KDE. The BIGTK and BIPN are carried in a BIGTK KDE. (11ba)The WIGTK and WIPN are carried in a WIGTK KDE. The CIGTK and CIPN are carried in a CIGTK KDE. The AP puts this element into the (Re)Association Response frame.

(..existing texts…)

Upon successful completion of the FILS authentication procedure, the STA shall process the Key Delivery element in the (Re)Association Response frame. The STA installs the GTK and (#1406)GTK RSC, and IGTK and IGTK RSC if management frame protection is enabled, and BIGTK and BIGTK RSC if present, and CIGTK and CIGTK RSC if control frame protection is negotiated in the Key Delivery element and dot11BeaconProtectionEnabled is true(11ba), and WIGTK and WIGTK RSC if present in the (#1488)Key Delivery element and dot11RSNAWURFrameProtectionActivated is true.

***TGmf editor: Modify 13 as follows:***

* Authenticator key holders

***Change the seventh paragraph as follows***

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.
* the R1KH shall derive and distribute the GTK and IGTK to all connected STAs.
* If WUR frame protection is enabled, the R1KH shall derive and distribute the IWGTK and WIPN to all WUR non-AP STAs with which the R1KH has negotiated WUR frame protection.
* 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.
* FT resource request protocol
* Over-the-air fast BSS transition with resource request

***Change the twelfth paragraph as follows:***

In an RSN, on successful completion of the FT authentication exchange of the FT resource request protocol, the PTKSA has been established and proven live. The key replay counter shall be initialized to 0, and the subsequent EAPOL-Key PDUs (e.g., GTK, IGTK, BIGTK, WIGTK, and CIGTK updates) shall use the key replay counter to detect and discard replays. The PTKSA shall be deleted by the target FTR if it does not receive a Reassociation Request frame from the FTO within the reassociation deadline timeout value.

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

***Change the twelfth paragraph as follows:***

In an RSN, on successful completion of the FT Confirm/Acknowledgment frame exchange, the PTKSA has been established and proven live. The key replay counter shall be initialized to 0, and the subsequent EAPOL-Key frames (e.g., GTK, IGTK, BIGTK, WIGTK, and CIGTK updates) shall use the key replay counter to detect and discard replays. The PTKSA shall be deleted by the target FTR if it does not receive a Reassociation Request frame from the FTO within the reassociation deadline timeout value. Resource request procedures are specified in 13.11 (Resource request procedures).

* FT reassociation
* FT reassociation in an RSN

***Change the second paragraph as follows:***

The FTO shall perform a reassociation directly with the target 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]], RIC-Response, RSNXE, Basic Multi-Link element)

* FT authentication sequence
* FT authentication sequence: contents of fourth message

***Change the fifth paragraph as follows:***

If present, the FTE shall be set as follows:

* 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, or CIGTK subelements. If a GTK, an IGTK, a BIGTK, WIGTK, or 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, or CIGTK subelement.

**C.3 MIB detail**

***TGmf editor: Modify C.3 as follows:***

**Dot11StationConfigEntry ::= SEQUENCE**

**{**

**…**

dot11CIPActivated TruthValue,

**}**

**dot11CIPActivated OBJECT-TYPE**

**SYNTAX TruthValue**

**MAX-ACCESS read-write**

**STATUS current**

**DESCRIPTION**

**"This is a control variable.**

**It is written by an external management entity or the SME.**

**Changes take effect as soon as practical in the implementation. This attribute indicates whether**

**or not CIP is enabled."**

**DEFVAL { false }**

(…existing texts…)

**Dot11RSNAStatsEntry ::=**

**SEQUENCE {**

**dot11RSNAStatsIndex Unsigned32,**

**dot11RSNAStatsSTAAddress MacAddress,**

**dot11RSNAStatsVersion Unsigned32,**

**dot11RSNAStatsSelectedPairwiseCipher OCTET STRING,**

**dot11RSNAStatsTKIPICVErrors Counter32,**

**dot11RSNAStatsTKIPLocalMICFailures Counter32,**

**dot11RSNAStatsTKIPRemoteMICFailures Counter32,**

**dot11RSNAStatsCCMPReplays Counter32,**

**dot11RSNAStatsCCMPDecryptErrors Counter32,**

**dot11RSNAStatsTKIPReplays Counter32,**

**dot11RSNAStatsCMACReplays Counter32,**

**dot11RSNAStatsRobustMgmtCCMPReplays Counter32,**

**dot11RSNAStatsBIPMICErrors Counter32,**

**(11ba)dot11RSNAStatsCMACWURReplays Counter32,**

**dot11RSNAStatsCIPReplays Counter32,**

**dot11RSNAStatsCIPMICErrors Counter32**

**}**

**dot11RSNAStatsCIPReplays**

 **SYNTAX Counter32**

 **MAX-ACCESS read-only**

 **STATUS current**

 **DESCRIPTION**

 **"This is a status variable.**

 **It is written by the MAC when the condition described below occurs.**

 **The number of received MPDUs discarded due to CIP replay errors."**

 **::= { dot11RSNAStatsEntry <ANA> }**

**dot11RSNAStatsCIPMICErrors OBJECT-TYPE**

 **SYNTAX Counter32**

 **MAX-ACCESS read-only**

 **STATUS current**

 **DESCRIPTION**

 **"This is a status variable.**

 **It is written by the MAC when the condition described below occurs.**

 **The number of received MPDUs discarded due to CIP MIC errors."**

 **::= { dot11RSNAStatsEntry <ANA> }**

Reference:

[1] 11-23-286 Trigger frame protection

[2] 11-23-0312 Thoughts on Secure Control frames

[3] 11-23-352 enhanced security discussion

[4] 11-23-1102 security enhancement follow up

[5] 11-23-1914 Enhanced Security Considerations in UHR

[6] 11-23-1915 Enhanced Security for Control frame in 11bn

[7] 11-23-1933 security enhancement follow up

[8] 11-23-1995 Trigger, BA, and BAR Protection

[9] 11-23-2001 Secure Control frames - Follow up

[10] 11-24-0151 Establishment of Security Key for Control Frame

[11] 11-24-497 security enhancement (control frame protection) follow up

[12] 11-24-535 Trigger, BA, and BAR Protection follow up

[13] 11-24-547 Secure Control frames - Follow Up

[14] 11-24-1226 ICF-ICR design

[15] 11-24-1661 Control frame protection in multiple BSSID

[16] 11-24-1897 Control frame protection keys

[17] 11-24-1990 On Protected Trigger Frame

[18] 11-24 2055 Detailed text proposal on Trigger, BA, BAR Protection

[19] 11-25-144 IFCS, PN and MIC inclusion in a Trigger Frame