### 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.

**Discussion:**

Trigger, BA, BAR protection has been discussed in 802.11 [1-18]. 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
* BA frame to move the transmission window
* BAR frame to move the received reordering buffer

Revme D7.0 has PBAC to address BAR attacks, but the capability disallows BAR frame and needs to rely on additional management frame to move the window, which loses the original benefits of BAR frame.

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 BA frames
* Individually addressed Compressed BAR frames
* Individually addressed Multi-TID BAR frames

GMAC-256 will be used 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 integriy group temporal key

CIP control frame integrity protocol

***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 protection for control frame is negotiated, the Protected Control subfield is set to 1 if the BlockAckReq frame contains information that has been processed with a message integrity check algorithm and is set to 0 if the BlockAckReq frame does not contain information that has been processed with a message integrity check algorithm. Otherwise, the Protected Control subfield is reserved.

The Protected Control subfield is reserved in all BlockAckReq variants except for Compressed BlockAckReq and Multi-TID BlockAckReq. When the Protected Control subfield is equal to 1, the BlockAckReq is protected utilizing the message integrity check algorithm as defined in clause 12.5.X (Control frame integrity protocol (CIP).

The Key ID subfield contains the key ID when the Protected Control subfield is 1. Otherwise, the Key ID subfield 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 subfield 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 subfield contains the PN corresponding to the integrity key indicated by the Key ID subfield. The PN subfield format is defined in Figure 9-1029 (PN field format).

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

The Padding field is optionally present in the BlockAckReq frame to extend the frame length to give the recipient STA enough time to perform 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 protection for control frame is negotiated, the Protected Control subfield is set to 1 in a Multi-STA BlockAck frame to indicate that the frame is protected and is set to 0 in a Multi-STA BlockAck frame to indicate that the frame is not protected. Otherwise the Protected Control subfield is reserved.

The Key ID subfield in a Multi-STA BlockAck frame with Protected Control subfield equal to 1 indicates the Key being used to protect the Multi-STA BlockAck frame. Otherwise the Key ID subfield 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 neither 2045, 2009, nor 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 neither 2045(11ax), 2009, nor 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 neither 2045(11ax), 2009, nor 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 value in AID11 subfield equal to 2009 is after other Per AID TID Info fields in the Multi-STA BlockAck frame with AID11 not equal to 2047 and that 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) to indicate the length of the PN And MIC subfield.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | 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 subfield 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 subfield contains the PN corresponding to the integrity key indicated by the Key ID subfield. The PN subfield format is defined in Figure 9-1029 (PN field format).

The MIC subfield contains a message integrity check calculated over the BlockAck frame as defined in 12.5.x (Control frame 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 value in AID11 subfield equal to 2047 is after 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 is reserved and the Fragment Number subfield of the Block Ack Starting Sequence Control subfield is set as defined in Table 9-40 (Fragment Number subfield encoding for the Multi-STA BlockAck variant) to indicate the length of the Padding subfield.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | 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 subfields | Context of a Per AID TID Info subfield in a Multi-STA BlockAck frame |
| 0 | 0–7 | Present | Block acknowledgment context:Sent as an acknowledgment to QoS Data frames that solicit a BlockAck frame response or to a BlockAckReq frame. |
| 1 | 0–7 | 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 2009Padding context if AID11 subfield is equal to 2047 |
| 1 | 0 | Not present | 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-SIGA2 Reserved subfield to all 1s except when protection for control frame is negotiated, where B61 and B62 setting are set as follows.

The Protected Control subfield is in B61 of the Common Info field of the Trigger frame. If protection for control frame is negotiated, the Protected Control subfield is set to 1 if the Trigger frame contains information that has been processed with a message integrity check algorithm and is set to 0 if the Trigger frame does not contain information that has been processed with a message integrity check algorithm.

When the Protected Control subfield is equal to 1, the Trigger frame is protected utilizing the message integrty check algorithm as defined in clause 12.5.X (Control frame integrity protocol (CIP).

The Key ID subfield is in B62 of the Common Info field of the Trigger frame. The Key ID subfield contains the key ID when the Protected Control subfield is 1. Otherwise, the Key ID subfield is reserved.

(…existing texts….)

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

The Control MIC field contains the PN subfield and the MIC subfield as shown in Figure 9.XY (Control MIC field format).

The PN subfield contains the PN corresponding to the integrity key indicated by the Key ID subfield. The PN subfield format is defined in Figure 9-xxx (Formats of User Info fields with AID12 subfield equal to 2009) and is carried in two contiguous User Info fields, each with AID12 subfield equal to 2009. The format of the User Info fields with AID12 subfield equal to 2009 is shown in Figure 9-64d (Formats of User Info fields with AID12 subfield equal to 2009). The Trigger Dependent User Info field (if present) is set to 0.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | B0   B11 | B12    B15 | B16 B23 | B24   B31 | B32 B39  |  |
| First User Info field | AID12 (2009) | 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 | AID12 (2009) | 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 MIC subfield contains a message integrity check calculated over the Trigger frame as defined in 12.5.x (Control frame integrity protocol (CIP)). The MIC field is carried in five contiguous User Info fields, each with AID12 subfield equal to 2010. The format of the User Info fields with AID12 subfield equal to 2010 is shown in Figure 9-64d (Formats of User Info fields with AID12 subfield equal to 2010). The Trigger Dependent User Info field (if present) is set to 0.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | B0 B11 | B12  B15 |  B16 B39  |  |  |
| First User Info field | AID12 (2010) | Reserved | MIC[0:23] | TriggerDependentUser Info |
| Bits: | 12 | 4 | 24 | Variable |
|  | B0 B11 | B12 B15 |  B16 B39  |  |
| Second User Info field | AID12 (2010) | Reserved | MIC[24:47] | TriggerDependentUser Info |
| Bits: | 12 | 4 | 24 | Variable |
|  | B0 B11 | B12 B15 |  B16 B39  |  |
| Third User Info field | AID12 (2010) | Reserved | MIC[48:71] | TriggerDependentUser Info |
| Bits: | 12 | 4 | 24 | Variable |
|  | B0 B11 | B12 B15 |  B16 B39  |  |
| Fourth User Info field | AID12 (2010) | Reserved | MIC[72:95] | TriggerDependentUser Info |
| Bits: | 12 | 4 | 24 | Variable |
|  | B0 B11 | B12 B15 |  B16 B39  |  |
| Fifth User Info field | AID12 (2010) | Reserved | MIC[96:119] | TriggerDependentUser Info |
| Bits: | 12 | 4 | 24 | Variable |
|  | B0 B11 | B12 B15 |  B16 B39  |  |
| Sixth User Info field | AID12 (2010) | 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

The Control MIC field is after other User Info fields in the Triger frame that are addressed to STAs that have negotiated control frame protection.***TGmf editor: Add new clause at the end of 12.5 as follows:***

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

**12.5.x.1 Overview**

Control frame integrity protocol (CIP) provides integrity and replay protection for the following control frames:

* Individually and group addressed Trigger frames
* Individually and group addressed Multi-STA BA frames
* Individually addressed Compressed BAR frames
* Individually addressed Multi-TID BAR frames

GMAC-256 shall be used for CIP.

If CIP is used, then GCMP-256 shall also be used as the pairwise cipher for individually addressed data and management frames. If CIP is used then 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.

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

CIP uses CIGTK 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 Protected Control frame Setup and Operation**

CIP is an optional feature. A STA that supports CIP has *dot11CIPActivated* equal to true and it sets the CIP Supported field in the RSNXE. If both the associated non-AP STA and AP have set CIP Supported field in the RSNXE, then the control frame protection is negotiated and all Trigger, C-BAR, Multi-TID BAR and M-BA frames transmitted between the non-AP STA and AP shall be protected.

A non-AP indicates in the CIP Capability element of (Re)Association Request frame the padding durations of the protected control frames. An AP indicates in the CIP Capability element of (Re)Association Response frame the padding durations of the protected control frames.

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

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 M-BA frame, instead of a GCR BA frame, in the TB PPDU that is sent in response to a protected GCR MU BAR Trigger frame (see 9.3.1.22.7). An AP shall not send a GCR BAR 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 M-BA frame in the TB PPDU that is sent in response to a protected MU BAR Trigger frame (see 9.3.1.22.4).

**12.5.x.3 Encapsulation format**

To provide integrity and replay protection, CIP utilizes 1 bit Key ID field, 6 bytes PN field, and 16 bytes MIC field in the control frame 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.4 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 | 6 | 6 | 6 |

Figure 12-xx—CIP AAD construction

**12.5.x.5 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 PN 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 1.
* The non-AP STA and the AP shall maintain a 48-bit replay counter to check replay of individually addressed control frame that are defined to be protected
* The AP shall maintain a PN for each CIGTK. The PN shall be implemented as a 48-bit strictly increasing integer, initialized to 1 when the corresponding CIGTK is initialized. A single PN space is maintained for all APs in a multiple BSSID set.
* The non-AP STA shall maintain a 48-bit replay counter for each CIGTK.

**12.5.x.6 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
2. The Key ID field set to the corresponding key ID. The PN field set to the corresponding PN. The nonce, i.e., the initialization vector, shall be a concatenation of TA field and the non-negative integer inserted into the PN field.
3. Compute AAD as specified in 12.5.x.3 (CIP AAD construction).
4. For Trigger frame, compute an integrity value over the concatenation of AAD and contents after TA field up to and including the last User Info that precedes the first User Info field that carries the MIC (see Figure XYZ (Formats of User Info fields with AID12 subfield equal to 2010) but not including anything from any of the User Info fields that carry MIC or any subsequent User Info fields that follow the User Info fields that carry the MIC (if any). Otherwise, compute an integrity value over the concatenation of AAD and contents after TA field and before MIC field. Insert the output into the MIC field.
5. Include padding if needed by the intended recipient(s).
6. Transmit the frame.

**12.5.x.7 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 as a 48-bit unsigned integer. The receiver shall compare the PN to the value of the corresponding replay counter identified by the Key ID field. If the value from the received PN field is less than or equal to the replay counter value, the receiver shall discard the frame and increment the dot11RSNACIPStatsReplays counter by 1.
3. The nonce, i.e., the initialization vector, shall be a concatenation of TA field and the non-negative integer inserted into the PN field.
4. Compute AAD as specified in 12.5.x.3 (CIP AAD construction).
5. Extract and save the received MIC value, and compute a verifier over the concatenation of AAD and contents after TA field and before MIC field. If the computed verifier does not match 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 with the value of the PN field.

**12.5.x.8 Padding**

A STA transmitting a PPDU that contains a BCC-encoded protected control frame shall ensure that for each target STA, the number of bits in the PSDU following the Vlast is at least VPAD,MAC, which is based on the MIC Verification 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 BAR frame
* Last bit of the Per AID TID Info field containing MIC (see Figure 9.xxx) if the frame is a Multi-STA BA frame
* Last bit of 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

*VPAD,MAC* = *NDBPSVPAD (12-x1)*

where

*NDBPS* is defined in Table 17-4 (Modulation-dependent parameters) for a non-HT 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).

*VPAD* is defined as follows:
—  For a non-HT PPDU, HT PPDU, and VHT PPDU, *VPAD* is

* 0 if MIC Verification Padding Delay is 0 us
* 1 if MIC Verification Padding Delay is 4 us
* 2 if MIC Verification Padding Delay is 8 us
* 3 if MIC Verification Padding Delay is 12 us
* 4 if MIC Verification Padding Delay is 16 us
* 5 if MIC Verification Padding Delay is 20 us
* 6 if MIC Verification Padding Delay is 24 us
* 7 if MIC Verification Padding Delay is 28 us
* 8 if MIC Verification Padding Delay is 32 us

 —  For an HE PPDU, *VPAD* is

* 0 if MIC Verification Padding Delay is 0 us
* 1 if MIC Verification Padding Delay is less than or equal to 16 us
* 2 if MIC Verification Padding Delay is less than or equal to 32 us

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

A STA transmitting a PPDU that contains a LDPC-encoded protected control frame shall ensure that for each target STA, VProc is greater than or equal to the MIC Verification 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 MIC Verification Padding Delay.

A STA transmitting a PPDU that contains the last BCC-encoded 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 CPAD,MAC, which is based on the MIC Calculation 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 BAR frame (see 9.3.1.7.1 (Overview))

*CPAD,MAC* = *NDBPSCPAD (12-x2)*

where

*NDBPS* is defined in Table 17-4 (Modulation-dependent parameters) for a non-HT 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-x2).

*CPAD* is defined as follows:
—  For a non-HT PPDU, HT PPDU, and VHT PPDU, *PPAD* is

* 0 if MIC Calculation Padding Delay is 0 us
* 1 if MIC CalculationPadding Delay is 4 us
* 2 if MIC CalculationPadding Delay is 8 us
* 3 if MIC CalculationPadding Delay is 12 us
* 4 if MIC CalculationPadding Delay is 16 us
* 5 if MIC CalculationPadding Delay is 20 us
* 6 if MIC CalculationPadding Delay is 24 us
* 7 if MIC CalculationPadding Delay is 28 us
* 8 if MIC CalculationPadding Delay is 32 us

 —  For an HE PPDU, *CPAD* is

* 0 if MIC CalculationPadding Delay is 0 us
* 1 if MIC CalculationPadding Delay is less than or equal to 16 us
* 2 if MIC CalculationPadding Delay is less than or equal to 32 us

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

A STA transmitting a PPDU that contains the last LDPC-encoded frame soliciting protected control frames shall ensure that for each target STA, CProc is greater than or equal to the MIC Calculation Padding Delay indicated by the target STA (see 9.4.2.xx (CIP Capabilities element)).

Except the exception mentioned in this clause, a STA may use any type of padding to satisfy the requirements such as using the Padding field in a Trigger frame, using the Padding field in a BAR frame, using one or more Per-AID TID Info subfields with the AID11 subfield equal to 2047 in a M-BA frame, using pre-EOF A-MPDU padding, using post-EOF A-MPDU padding, or aggregating other MPDUs in the A-MPDU.

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

**9.4.2 elements**

* General

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

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

* Element IDs

**12.7.2 EAPOL-Key frames**

***TGmf editor: 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 |
| … | … | … |

***TGmf editor: Insert the following:***

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

|  |  |  |
| --- | --- | --- |
| Key ID | CIPN | CIGTK |

bytes: 2 6 32

**Figure 12-50c—CIGTK KDE**

The Key ID field contains the CIGTK key ID.

The CIPN corresponds to the CIPN value used to protect the last protected group-addressed control frame. It is used by the receiver as the initial value for the CIP replay counter for the CIGTK.

The CIGTK field contains the CIGTK.

***TGmf editor: 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

***TGmf editor: 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 dot11RSNAOperatingC\hannelValidationActivated 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.

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

**9.4.2.xx CIP Capabilities element**

The CIP Capability element contains fields that are used to advertise padding delay of CIP.

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

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

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

The Padding Delay field indicates the padding delay of CIP.

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

|  |  |  |
| --- | --- | --- |
|  | MIC Calculation Padding Delay | MIC Verification Padding Delay |
|  Bits |  4 |  4 |

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

The MIC Calculation Padding Delay field indicates the minimum padding duration of the PPDU soliciting protected control frame from the STA that sends the CIP Capability element as defined in 12.5.x.7 (Padding).

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

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

|  |  |
| --- | --- |
| MIC Calculation Padding Delay field value | MIC Calculation padding delay |
| 0 | 0 |
| 1 | 4 us |
| 2 | 8 us |
| 3 | 12 us |
| 4 | 16 us |
| 5 | 20 us |
| 6 | 24 us |
| 7 | 28 us |
| 8 | 32 us |
| 9-15 | Reserved |

The MIC Verification Padding Delay field indicates the minimum padding duration of the protected control frame received by the STA that sends the CIP capability element as defined in 12.5.x.7 (Padding).

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

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

|  |  |
| --- | --- |
| MIC Verification Padding Delay field value | MIC Verification padding delay |
| 0 | 0 |
| 1 | 4 us |
| 2 | 8 us |
| 3 | 12 us |
| 4 | 16 us |
| 5 | 20 us |
| 6 | 24 us |
| 7 | 28 us |
| 8 | 32 us |
| 9-15 | 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.  |

**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 element | 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 element | 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 element | The CIP Capabilities element is present if dot11CIPActivated is true; otherwise, it is not present. |

**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 }**

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