IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LB 203 Comment Resolution for clause 11 | | | | |
| Date: 2014-09-01 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Alfred Asterjadhi | Qualcomm Inc. | 5775 Morehouse Dr, San Diego, CA 92109 | +1-858-658-5302 | aasterja@qti.qualcomm.com |
| Menzo Wentink | Qualcomm Inc. |  |  |  |

Abstract

This submission proposes resolutions for comments in clause 11 of TGah Draft 2.0 with the following CIDs (12 CIDs):

* 3202, 3416, 3420, 3421, 3422, 3857
* 3163, 3203, 3419, 3856, 4193, 4194

Revisions:

* Rev 0: Initial version of the document
* Rev 1: Added missing text for PV1 QMF frames (in green)

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 TGah Draft. This introduction is not part of the adopted material.

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

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **P.L** | **Clause** | **Comment** | **Proposed Change** | **Resolution** |
| 3163 | 341.1 | 11 | An S1G STA can use GCMP for secure PV1 frames | Make the appropriate changes to clarify the GCMP MPDU format for PV1 frames. | Rejected --  The comment does not identify a technical issue.  The commenter is working on a submission that clarifies the GCMP MPDU format for PV1 MPUs which will be discussed during the comment resolution phase of a future LB. |
| 3203 | 347.43 | 11.4.3.4.2 | Duplicate occurrence of Short Management frame in the sentence. Also PV1 terminology is used in other parts so would it be better to use the following terminology: PV1 Management frame? | Replace the second occurrence of Short Management frame with "PV1 Data frame" and "Short Management frame with PV1 Management frame" throughout the draft. | Accepted |
| 3419 | 341.31 | 11.4.3.2 | the size of the MIC field in CCMP is independent of the key length. There's another parameter you need to define too. You're defining a new encapsulation here so explain why you're doing what you're doing. | make separate sections defining the CCMP parameters and explain the situations in which they are different. Include both the L parameter and the M paramete as well as the key length. | Revised –  According to REVmc D3.0 the size of the MIC field in secure PV0 frames depends on the key length. Please refer to paragraph in [P1889L55@802.11REVmc](mailto:P1889L55@802.11REVmc) D3.0. PV1 frames are no different in this aspect with PV0 frames. Neither are they with respect to L, M, and key lengths parameters The main difference between secure PV1 and secure PV0 MPDUs is that in the PV1 case the CCMP header is not included in the frame but rather stored at the receiver.  Please refer to the following presentation for more info: <https://mentor.ieee.org/802.11/dcn/14/11-14-0157-00-00ah-ccmp-header-compression.pptx>  Proposed resolution is to use the same wording that is used in D3.0.  TGah editor to make the changes shown in 11-14/1118r1 under all headings that include CID 3419. |
| 3856 | 341.1 | 11 | Relay security should be optimized. | Define the optimized relay security protocol | Rejected –  The comment fails to identify a specific issue to be addressed. It fails to identify changes in sufficient detail so that the specific wording of the changes that will satisfy the commenter can be determined. |
| 4193 | 341 | 11.4 | It appears that 11h draft 2.0 is based on 11ac D5.0 (based on change description in). That draft (as well as recently approved 11mcd3.0) contains support for GCM and GCM/CCM 256 bit ciphers. The changes for supporting GCM seem missing from this section | Rework the section to include GCM | Rejected --  The comment does not identify a technical issue.  The commenter of CID 3163 is working on a submission that clarifies the GCMP MPDU format for PV1 MPUs which will be discussed during the comment resolution phase of a future LB. |
| 4194 | 348.15 | 11.4.3.4.4 | Maintaining separate replay counters for each PTKSA and protocol version values seems too onerous. If I understand correctly, this is doubling the number of replay counters maintained per association (PTKSA). Why should reordering be allowed across protocol versions from the same sender - packets are already allowed to be reordered across TIDs | Drop this requirement | Rejected -  The PN for PV1 is derived from the sequence number, so having it as a separate replay counter avoids that the replay counter for PV0 also becomes dependent on the sequence counter. Reordering across protocol values is not allowed, because the sequence number space is the same independent of the protocol value. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **P.L** | **Clause** | **Comment** | **Proposed Change** | **Resolution** |
| 3202 | 342.5 | 11.4.3.2a | Short Management frames do not have a TID. Same observation for mechanism in 9.53. | clarify how the construction of the CCMP header is done for Short Management frames. Idem for QMF Management frames and Header Compression procedure. | Revised --  Agree in principle with the commenter. Proposed resolution accounts for the suggested change in this subclause.  TGah editor to make the changes shown in 11-14/1118r1 under all headings that include CID 3202. |
| 3416 | 342.14 | 11.4.3.2a | the multiple conditionals "when not...or when...when..." is confusing. | split it up into multiple sentences and explain when happens when each conditional is true and not true, for each sentence. | Revised --  Agree in principle with the commenter. Proposed resolution accounts for the suggested change.  TGah editor to make the changes shown in 11-14/1118r1 under all headings that include CID 3416. |
| 3420 | 342.25 | 11.4.3.2a | This algorithm doesn't look like it will work right. "w" never changes and "b" can take on the value of "SN", which I believe grows. So eventually "b" will always be greater than "w". | fix the algorithm, does "b" need to be modularly reduced? | Rejected –  *b* is already modular because the SN rolls over at 2^12, which implies that *b* will never be larger than 2^12 either. The window size w is fixed, but the first test (if b >= w) is to see whether the window sits somewhere between 0 and 2^12 (in which case b > a), or whether it is split into two parts, around 0 (in which case b < a). |
| 3421 | 341.63 | 11.4.3.2a | 11ah is basically a new PHY. Why are the MAC changes to how CCMP is handled? | Use CCMP the same way as it's used in the base standard | Rejected –  The comment fails to identify an issue and is asking a question.  As a response to the comment: The changes are introduced so that PV1 frames do not need to include the CCMP header, reducing the overhead to a minimum. This topic has been discussed in TGah and for more information the commenter is referred to <https://mentor.ieee.org/802.11/dcn/14/11-14-0157-00-00ah-ccmp-header-compression.pptx> |
| 3422 | 341.63 | 11.4.3.2a | there is special encapsulation for CCMP for PV1 frames but apparently PV1 can use the same encapsulation for GCMP. | Either get rid of the special CCMP encapsulation for PV1 frames or do the same thing for GCMP frames, and explain why you're doing this new encapsulation. | Rejected –  The comment does not identify a technical issue.  These changes allow a PV1 frame to not include the CCMP header and more information regarding this can be found in <https://mentor.ieee.org/802.11/dcn/14/11-14-0157-00-00ah-ccmp-header-compression.pptx>  The commenter of CID 3163 is working on a submission that clarifies the GCMP MPDU format for PV1 MPUs which will be discussed during the comment resolution phase of a future LB. |
| 3857 | 342.18 | 11.4.3.2a | The proposed update algorithm is too complicate. | Simplify it. | Rejected –  The comment fails to identify a specific issue to be addressed. It fails to identify changes in sufficient detail so that the specific wording of the changes that will satisfy the commenter can be determined.  There are two update algorithms specified. One is the implicit update which simply indicates to the intended receiver to increase its BPN if the SN of the received frame falls within the sequence number window W. The other mechanism is the explicit update where the transmitter sends a Header Compression request to explicitly update the BPN. |

**Discussion:** *None.*

* **Security**
* **Framework**
* **RSNA assumptions and constraints**
* **RSNA confidentiality and integrity protocols**
* **CTR with CBC-MAC Protocol (CCMP)**
* **CCMP MPDU format**

***Change the length of the CCMP header field in Figure 11-16 from "8 octets" to "0 or 8 octets".***

***Change the 2nd paragraph in 802.11ac D5.0 and insert a new figure as follows:***

For secure PV0 MPDUs, CCMP-128 processing expands the original MPDU size by 16 octets, 8 octets for the CCMP Header field and 8 octets for the MIC field. CCMP-256 processing expands the original MPDU size by 24 octets, 8 octets for the CCMP Header field, and 16 octets for the MIC field. The CCMP Header field is constructed from the PN, ExtIV, and Key ID subfields. PN is a 48-bit PN represented as an array of 6 octets. PN5 is the most significant octet of the PN, and PN0 is the least significant.

**TGah Editor: *Change the paragraph below as follows (#3419):***

The CCMP header is not included in secure PV1 MPDUs, but constructed locally at the STA as defined in 11.4.3.2a (Construction of the CCMP Header for PV1 MPDUs). For secure PV1 MPDUs, CCMP-128 processing expands the original MPDU size by 8 octets for the MIC field. CCMP-256 processing expands the original MPDU size by 16 octets for the MIC field. Figure 11-16a (Expanded PV1 CCMP MPDU) depicts the PV1 MPDU when using CCMP.

Note that CCMP does not use the WEP ICV.

|  |
| --- |
|  |
| * **Expanded PV1 CCMP MPDU** |

***Insert a new subclause after subclause 11.4.3.2 as follows:***

* **Construction of the CCMP Header for PV1 MPDUs**

**TGah Editor: *Change the paragraph below as follows (#3202):***

The CCMP Header is not present in secure PV1 MPDUs, but constructed locally at the STA as follows (where || denotes concatenation):

* The PN is composed of the Sequence Control (SC) field and a Base PN (BPN), as SC||BPN, where
* the Sequence Control field is present in the MPDU header
* PN0||PN1 = SC
* the Base PN is retrieved from the local storage at the receiver, per TID/ACI
* PN2||PN3||PN4||PN5 = BPN
* PN = PN0||PN1|| PN2||PN3||PN4||PN5 (= SC||BPN)
* The Key ID is retrieved from the local storage at the receiver

The locally stored BPN and Key ID are initialized at 0 when a secure link is established.

**TGah Editor: *Change the paragraph below as follows (#3202, 3416):***

The locally stored BPN shall be incremented by 1 when the Sequence Number of the MPDU is less than the previous Sequence Number for that TID/ACI if any of the following two conditions is satisfied:

* Block Ack is not used
* Block Ack is used but decryption occurs after Block Ack reordering

When Block Ack is used and decryption occurs before Block Ack reordering, the BPN may be updated as follows. The receiver maintains a sequence number window of size w, which is equal to twice the Block Ack reorder window. The sequence number window has a lower edge *a* and an upper edge *b*. For a received sequence number SN (as part of the received sequence control field SC), the associated packet number (PN) is determined as follows (where b is initialized as *b* = 0):

* if (*b* ≥ w) then
* *a* = *b* − w
* if (SN < *a*) then BPN = BPN + 1
* PN = SC||BPN
* if not (*a* < SN < *b*) then *b* = SN
* else (i.e., *b* < w) then
* *a* = *b* − w + 212
* if (SN < *a*) then PN = SC||BPN
* if (SN ≥ *a*) then PN = SC||(BPN − 1)
* if (b < SN < *a*) then *b* = SN

The BPN can also be updated explicitly through a Header Compression Request/Response exchange, as defined in 9.53 (Header Compression procedure).

* **CCMP cryptographic encapsulation**
* **General**

***Change the 2nd paragraph as follows:***

For secure PV0 MPDUs, CCMP encrypts the payload of a plaintext MPDU and encapsulates the resulting cipher text using the following steps:

* Increment the PN, to obtain a fresh PN for each MPDU, so that the PN never repeats for the same temporal key. Note that retransmitted MPDUs are not modified on retransmission.
* Use the fields in the MPDU header to construct the additional authentication data (AAD) for CCM. The CCM algorithm provides integrity protection for the fields included in the AAD. MPDU header fields that may change when retransmitted are muted by being masked to 0 when calculating the AAD.
* Construct the CCM Nonce block from the PN, A2, and the priority value of the MPDU where A2 is MPDU Address 2. If the Type field of the Frame Control field is 10 (Data frame) and there is a QoS Control field present in the MPDU header, the priority value of the MPDU is equal to the value of the QC field TID (bits 0 to 3 of the QC field). If the Type field of the Frame Control field is 00 (Management frame), and the frame is a QMF, the priority value of the MPDU is equal to the value in the ACI subfield of the Sequence Number field. Otherwise, the priority value of the MPDU is equal to the fixed value 0.
* Place the new PN and the key identifier into the 8-octet CCMP header.
* Use the temporal key, AAD, nonce, and MPDU data to form the cipher text and MIC. This step is known as CCM originator processing.
* Form the encrypted MPDU by combining the original MPDU header, the CCMP header, the encrypted data and MIC, as described in 11.4.3.2 (CCMP MPDU format)

**TGah Editor: *Change the paragraph below as follows (#3202):***

For secure PV1 MPDUs, CCMP encrypts the payload of a plaintext MPDU and encapsulates the resulting cipher text using the following steps:

* When the Sequence Number of the MPDU is less than the previous Sequence Number and satisfies the BPN update conditions in 11.4.3.2a (Construction of the CCMP Header for PV1 MPDUs) for that TID/ACI, increment the Base PN, to obtain a fresh PN for each MPDU, so that the PN never repeats for the same temporal key and TID/ACI. Note that retransmitted MPDUs are not modified on retransmission.
* Use the fields in the MPDU header to construct the additional authentication data (AAD) for CCM. The CCM algorithm provides integrity protection for the fields included in the AAD. MPDU header fields that might change when retransmitted are muted by being masked to 0 when calculating the AAD.
* Construct the CCMP header as defined in 11.4.3.2a (Construction of the CCMP Header for PV1 MPDUs). If the Type field of the Frame Control field is 001 (Management frame), and the frame is a QMF, the priority value of the MPDU is equal to the value in the ACI subfield of the Sequence Number field. Otherwise, the priority value of the MPDU is equal to the fixed value 0.
* Construct the CCM Nonce block from the PN, A2, and the Priority field of the MPDU where A2 is the STA MAC Address identified by MPDU Address 2.
* Use the temporal key, AAD, nonce, PN and MPDU data to form the cipher text and MIC. This step is known as CCM originator processing.
* Form the encrypted MPDU by combining the original MPDU header, the encrypted data and MIC, as described in 11.4.3.2 (CCMP MPDU format).
* **PN processing**

***Change the 1st paragraph as follows:***

The PN is incremented by a positive number for each MPDU. For PV0 MPDUs, t~~T~~he PN shall never repeat for a series of encrypted MPDUs using the same temporal key.

**TGah Editor: *Change the paragraph below as follows (#3202):***

For PV1 MPDUs, the PN shall never repeat for a series of encrypted MPDUs using the same temporal key and TID/ACI.

* **Construct AAD**

***Change the 1st, 2nd and 3rd paragraphs as follows:***

For PV0 MPDUs, t~~T~~he format of the AAD is shown in Figure 11-18 (AAD construction for PV0 MPDUs

).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | FC | A1 | A2 | A3 | SC | A4 | QC |
| Octets: | 2 | 6 | 6 | 6 | 2 | 6 | 2 |

* **AAD construction for PV0 MPDUs**

The length of the AAD for PV0 MPDUs varies depending on the presence or absence of the QC and A4 fields and is shown in Table 11-1 (AAD length for PV0 MPDUs).

|  |  |  |
| --- | --- | --- |
| **QC field** | **A4 field** | **AAD length (octets)** |
| Absent | Absent | 22 |
| Present | Absent | 24 |
| Absent | Present | 28 |
| Present | Present | 30 |

* **AAD length for PV0 MPDUs**

The AAD is constructed from the MPDU header. The AAD does not include the header Duration field, because the Duration field value might change due to normal IEEE Std 802.11 operation (e.g., a rate change during retransmission). The AAD includes neither the Duration/ID field nor the HT Control field because the contents of these fields might change during normal operation (e.g., due to a rate change preceding retransmission). The HT Control field might also be inserted or removed during normal operation (e.g., retransmission of an A MPDU where the original A MPDU included an MRQ that has already generated a response). For similar reasons, several subfields in the Frame Control field are masked to 0. For PV0 MPDUs, the AAD construction is performed as follows:

***Insert the following at the end of the sub-clause 11.4.3.3.3 as the following:***

For PV1 MPDUs, the format of the AAD is shown in Figure 11-18a (AAD construction for PV1 MPDUs).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | FC | A1 | A2 | SC | A3 | A4 |
| Octets: | 2 | 6 or 2 | 6 or 2 | 2 | 0 or 6 | 0 or 6 |
| * **AAD construction for PV1 MPDUs** | | | | | | |

For PV1 MPDUs, the length of the AAD varies depending on the presence or absence of the A3 and A4 fields and is shown in Table 11-1a (AAD length for PV1 MPDUs).

|  |  |  |  |
| --- | --- | --- | --- |
| * **AAD length for PV1 MPDUs** | | | |
| Type field value in the Frame Control | A3 field | A4 field | AAD length (octets) |
| 0 or 1 | Absent | Absent | 12 |
| 0 or 1 | Present | Absent | 18 |
| 0 | Absent | Present | 18 |
| 0 | Present | Present | 24 |
| 3 | Absent | Absent | 16 |

For PV1 MPDUs, AAD construction is performed as follows:

* FC – MPDU Frame Control field, with
* Type bits (bits 3 4) in a Data MPDU masked to 0
* Power Management bit (bit 10) masked to 0
* More Data bit (bit 11) masked to 0
* Protected Frame bit (bit 12) always set to 1
* EOSP bit (bit 13) masked to 0
* Relayed Frame bit (bit 14) masked to 0
* Ack Policy bit (bit 15) masked to 0
* A1 –MPDU Address 1 field.
* When the SID field is present as the A1 field
* A3 Present bit (bit 13) of SID field masked to 0
* A4 Present bit (bit 14) of SID field masked to 0
* A-MSDU bit (bit 15) of SID field is masked to 0 if either the STA or its peer has the SPP A-MSDU Capable field equal to 0
* A2 –MPDU Address 2 field.
* When the SID field is present as the A2 field
* A3 Present bit (bit 13) of SID field masked to 0
* A4 Present bit (bit 14) of SID field masked to 0
* A-MSDU bit (bit 15) of SID field is masked to 0 if either the STA or its peer has the SPP A-MSDU Capable field equal to 0
* A3 –MPDU Address 3 field, if present.
* A4 –MPDU Address 4 field, if present.
* SC – MPDU Sequence Control field, with the Sequence Number subfield (bits 4–15 of the Sequence Control field) masked to 0. The Fragment Number subfield is not modified.
* **Construct CCM nonce**

***Change Figures 11-19, 11-20, and the 2nd paragraph in the sub-clause 11.4.3.3.4 as the following:***

|  |  |  |  |
| --- | --- | --- | --- |
|  | Nonce Flags | STA MAC Address identified by A2 | PN |
| Octets: | 1 | 6 | 6 |
| * **Nonce construction** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| B0 B3 | B4 | B5 | B6~~5~~ B7 |
| Priority | Management | PV1 | Reserved |
| 4 | 1 | 1 | 2 |

* **Nonce Flags subfields**

The Nonce field has an internal structure of Nonce Flags || STA MAC Address identified by A2 || PN (“||” is concatenation), where

* The Priority subfield of the Nonce Flags field shall be set to the priority value of the MPDU
* When management frame protection is negotiated, the Management field of the Nonce Flags field shall be set to 1 if the PV0 MPDU’s Type field of the Frame Control field is 00 (Management frame) or the PV1 MPDU’s Type field of the Frame Control field is 001 (Management frame); otherwise it is set to 0.
* The PV1 subfield of the Nonce Flags field shall be set to 1 when the value of the Protocol Version field of the Frame Control field of the MPDU header is equal to 1. The PV1 subfield of the Nonce Flags field shall be set to 0 otherwise.
* Bits ~~5~~6 to 7 of the Nonce Flags field are reserved and shall be set to 0 on transmission.
* STA MAC Address identified by ~~MPDU address~~ A2 field occupies octets 1–6. This shall be encoded with the octets ordered with STA MAC Address identified by A2 octet 0 at octet index 1 and STA MAC Address identified by A2 octet 5 at octet index 6.
* The PN field occupies octets 7–12. The octets of PN shall be ordered so that PN0 is at octet index 12 and PN5 is at octet index 7.
* **CCM originator processing**

***Change the 2nd paragraph as follows:***

There are four inputs to CCM originator processing:

* Key: the temporal key (16 octets).
* Nonce: the nonce (13 octets) constructed as described in 11.4.3.3.4 (Construct CCM nonce).
* Frame body: the frame body of the MPDU.
* AAD: the AAD (~~2~~12–30 octets) constructed from the MPDU header as described in 11.4.3.3.3 (Construct AAD).
* **CCMP decapsulation**
* **General**

***Change the 2nd paragraph as follows:***

For secure PV0 MPDUs, CCMP decrypts the payload of a cipher text MPDU and decapsulates a plaintext MPDU using the following steps:

* The encrypted MPDU is parsed to construct the AAD and nonce values.
* The AAD is formed from the MPDU header of the encrypted MPDU.
* The Nonce value is constructed from the A2, PN, and Nonce Flags fields.
* The MIC is extracted for use in the CCM integrity checking.
* The CCM recipient processing uses the temporal key, AAD, nonce, MIC, and MPDU cipher text data to recover the MPDU plaintext data as well as to check the integrity of the AAD and MPDU plaintext data.
* The received MPDU header and the MPDU plaintext data from the CCM recipient processing are concatenated to form a plaintext MPDU.
* The decryption processing prevents replay of MPDUs by validating that the PN in the MPDU is greater than the replay counter maintained for the session.

**TGah Editor: *Change the paragraph below as follows (#3202):***

For secure PV1 MPDUs, CCMP decrypts the payload of a cipher text MPDU and decapsulates a plaintext MPDU using the following steps:

* The encrypted MPDU is parsed to construct the AAD and nonce values.
* The CCMP header is constructed as defined in 11.4.3.2a (Construction of the CCMP Header for PV1 MPDUs).
* The AAD is formed from the MPDU header of the encrypted MPDU.
* The Nonce value is constructed from the STA MAC Address identified by A2, PN, and Nonce Flags fields.
* The MIC is extracted for use in the CCM integrity checking.
* The CCM recipient processing uses the temporal key, AAD, nonce, MIC, and MPDU cipher text data to recover the MPDU plaintext data as well as to check the integrity of the AAD and MPDU plaintext data.
* The received MPDU header and the MPDU plaintext data from the CCM recipient processing are concatenated to form a plaintext MPDU.
* The decryption processing prevents replay of MPDUs by validating that the PN in the CCMP header is greater than the replay counter maintained for the session and TID/ACI.

***Change the last paragraph of this subclause as follows:***

When the received frame is a CCMP protected individually addressed robust Management frame or Short Management frame, contents of the MMPDU body after protection is removed shall be delivered to the SME via the MLME primitive designated for that Management frame or Short Management frame rather than through the MA-UNITDATA.indication primitive.

* **CCM recipient processing**

***Change the 1st and 2nd paragraph of this subclause as follows:***

CCM recipient processing uses the same parameters as CCM originator processing. A CCMP protected individually addressed robust Management frame or Short Management frame shall use the same TK as a Data frame or Short Management frame.

There are four inputs to CCM recipient processing:

* *Key:* the temporal key (16 octets).
* *Nonce:* the nonce (13 octets) constructed as described in 11.4.3.3.4 (Construct CCM nonce).
* *Encrypted frame body:* the encrypted frame body from the received MPDU. The encrypted frame body includes the MIC.
* *AAD:* the AAD (~~2~~12–30 octets) that is the canonical MPDU header as described in 11.4.3.3.3 (Construct AAD).
* **PN and replay detection**

**TGah Editor: *Change the paragraph below as follows (#3202):***

To effect replay detection, the receiver extracts the PN from the CCMP header. NOTE: The CCMP header is not present in secure PV1 MPDUs, but constructed locally at the STA as defined in 11.4.3.2a (Construction of the CCMP Header for PV1 MPDUs). See 11.4.3.2 (CCMP MPDU format) for a description of how the PN is encoded in the CCMP header. The following processing rules are used to detect replay:

* The PN values sequentially number each MPDU.
* Each transmitter shall maintain a single PN (48-bit counter) for each PTKSA, GTKSA, and STKSA, and for PV1 transmissions for each TID/ACI. NOTE: The PN for secure PV1 MPDUs is based on the sequence number of the MPDU.
* The PN shall be implemented as a 48-bit monotonically incrementing non-negative integer, initialized to 1 when the corresponding temporal key is initialized or refreshed.
* A receiver shall maintain a separate set of PN replay counters for each PTKSA, GTKSA, ~~and~~ STKSA, and Protocol Version value. The receiver initializes these replay counters to 0 when it resets the temporal key for a peer. The replay counter is set to the PN value of accepted CCMP MPDUs.
* For each PTKSA, GTKSA, ~~and~~ STKSA, and Protocol Version value, the recipient shall maintain a separate replay counter for each IEEE Std 802.11 MSDU or A MSDU priority and shall use the PN recovered from a received frame to detect replayed frames, subject to the limitation of the number of supported replay counters indicated in the RSN Capabilities field (see 8.4.2.24 (RSNE)). A replayed frame occurs when the PN extracted from a received frame is less than or equal to the current replay counter value for the frame’s MSDU or A MSDU priority and frame type. A transmitter shall not use IEEE Std 802.11 MSDU or A MSDU priorities without ensuring that the receiver supports the required number of replay counters. The transmitter shall not reorder CCMP protected frames that are transmitted to the same DA within a replay counter, but may reorder frames across replay counters. One possible reason for reordering frames is the IEEE Std 802.11 MSDU or A MSDU priority.
* If dot11RSNAProtectedManagementFramesActivated is true, the recipient shall maintain a single replay counter for received individually addressed robust Management frames that are received with the To DS field equal to 0, and a single replay counter for received individually addressed robust Short Management frames and shall use the PN from the received frame to detect replays. If dot11QMFActivated is also true, the recipient shall maintain an additional replay counter for each ACI for received individually addressed Robust Management frames and Robust Short Management frames that are received with the To DS field equal to 1. The QMF receiver shall use the ACI encoded in the Sequence Number field of the received frame to select the replay counter to use for the received frame, and shall use the PN from the received frame to detect replays. A replayed frame occurs when the PN from the frame is less than or equal to the current value of the(11ae) management frame replay counter that corresponds to the ACI of the frame. The transmitter shall preserve the order of protected robust Management frames and Short Management frames that are transmitted to the same DA without the QMF service. When the QMF service is used, the transmitter shall not reorder robust IQMFs within an AC when the frames are transmitted to the same RA.
* If dot11RSNAProtectedManagementFramesActivated is true and dot11MeshSecurityActivated is true, the recipient shall maintain a single replay counter for received group addressed robust Management frames and for received group addressed robust Short Management frames that do not use the QMF service and shall use the PN from the received frame to detect replays. If dot11QMFActivated is also true, the recipient shall maintain an additional replay counter for each ACI for received group addressed Robust Management frames and for received group addressed robust Short Management frames that use the QMF service. The QMF receiver shall use the ACI encoded in the Sequence Number field of the received frame to select the replay counter to use for the received frame, and shall use the PN from the received frame to detect replays. A replayed frame occurs when the PN from the frame is less than or equal to the value of the management frame replay counter that corresponds to the ACI of the frame. The transmitter shall preserve the order of protected robust Management frames and Short Management frames transmitted to the same DA without the QMF service. When the QMF service is used, the transmitter shall not reorder robust GQMFs within an AC when the frames are transmitted to the same RA.
* The receiver shall discard MSDUs and MMPDUs whose constituent MPDU PN values are not sequential. A receiver shall discard any MPDU that is received with its PN less than or equal to the replay counter. When discarding a frame, the receiver shall increment by 1 the value of dot11RSNAStatsCCMPReplays for Data frames or dot11RSNAStatsRobustMgmtCCMPReplays for robust Management frames.
* For MSDUs or A-MSDUs sent using the block ack feature, reordering of received MSDUs or A-MSDUs according to the block ack receiver operation (described in 9.23.4 (Receive buffer operation)) is performed prior to replay detection.