IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LB 200 Comment Resolution for Clause 11 | | | | |
| Date: 2014-01-01 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Menzo Wentink | Qualcomm Inc. |  |  |  |
| Alfred Asterjadhi | Qualcomm Inc. | 5775 Morehouse Dr, San Diego, CA 92109 | +1-858-658-5302 | aasterja@qti.qualcomm.com |
| Simone Merlin | Qualcomm Inc. |  |  |  |
| Amin Jafarian | Qualcomm Inc. |  |  |  |

Abstract

This submission proposes resolutions for comments in clause 11 of TGah Draft 1.0 with the following CIDs:

1551, 1988, 2521, 1641, 1642, 1991, 1992, 1993, 1994, 1995, 1996, 2644, 2815, 2816, 1643, 1644, 1997, 2487

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** |
| 1551 | 233.01 | 11 | Define a short CCMP header for short frames that has a reduced overhead. Also there are some inconsistencies throughout clause 11 due to modifications to Short frames format. | Will submit a document with the resolution. | Agree with the commenter. See discussion.  Revised –  TGah editor to make changes shown in 14/0080r0 under the heading for CIDs from 1551 to 2487. |
| 1988 | 233.14 | 11.1.6 | Wording inconsistent with base standard and in wrong clause | Move text to the end of clause 5.1.2 and change to 'An S1G STA shall not use the pairwise cipher suite selectors WEP-40 WEP-104 TKIP or Use group cipher suite.' | Agree with the commenter.  Revised –  TGah editor to make the changes suggested by the commenter to the sentence in P233.14 and successively move it at the end of clause 5.1.2. |
| 2521 | 233.14 | 11.1.6 | WEP and TKIP are deprecated in IEEE 802.11-2012 so this statement is not needed. | Remove the statement since it is not required. | Revised –  Same resolution as CID 1988. |
| 1641 | 233.25 | 11.4.3.3.3 | If new frames such as the short Beacon are to be robust then the AAD needs to be specified for short management frames as well as for short data frames. | Review security requirements for the new short management frames and add details for AAD construction if necessary. | Agree in principle with the commenter regarding the short management frames (but short beacon is a PV0 Extensione frame so no changes to this frame are required). Hence, the proposed resolution is to review security requirements for short management frames.  Revised –  TGah editor to make changes shown in 14/0080r0 under the heading for CIDs from 1551 to 2487. |
| 1642 | 234.05 | 11.4.3.3.3 | AAD construction for short data frames does not match the frame control field in Figure 8-532b | Review short MAC header for data frames and make AAD construction consistent with that. | Agree with the commenter. Proposed change solves the inconsistency.  Revised –  TGah editor to make changes shown in 14/0080r0 under the heading for CIDs from 1551 to 2487. |
| 1991 | 233.29 | 11.4.3.3.3 | What is the format of the AAD fo the Short MAC Header if dot11ShortMACHeaderImplemented is false? | Delete the phrase 'When dot11ShortMACHeaderImplemented is set to true' | Agree with the commenter. Proposed resolution accounts for the change.  Revised –  TGah editor to make changes shown in 14/0080r0 under the heading for CIDs from 1551 to 2487. |
| 1992 | 233.43 | 11.4.3.3.3 | What is the length of the AAD fo the Short MAC Header if dot11ShortMACHeaderImplemented is false? | Delete the phrase 'When dot11ShortMACHeaderImplemented is set to true' | Agree with the commenter. Proposed resolution accounts for the change.  Revised –  TGah editor to make changes shown in 14/0080r0 under the heading for CIDs from 1551 to 2487. |
| 1993 | 234.16 | 11.4.3.3.3 | How is A1 included in the AAD calculation if the SID is not present in the A1 field? | Add text describing the inclusion of A1 in the AAD when A1 is not and SID | Rejected –  The A1 included in the AAD calculation contains always the MAC Address of the STA identified by the A1 field which can be either a 48 bit MAC address or the AID of the STA (refer to11.4.3.3.4 Construct CCM nonce). |
| 1994 | 234.20 | 11.4.3.3.3 | How is A2 included in the AAD calculation if the SID is not present in the A2 field? | Add text describing the inclusion of A2 in the AAD when A1 is not and SID | Rejected –  The A2 included in the AAD calculation contains always the MAC Address of the STA identified by the A2 field which can be either a 48 bit MAC address or the AID of the STA (refer to11.4.3.3.4 Construct CCM nonce). |
| 1995 | 234.16 | 11.4.3.3.3 | its peer has the SPP A-MSDU Capable field equal to 0' is not something observable at the STA | Use some information that is currently available at the encrypting STA and has been authenticated i.e. some information that was obtained during security negotiation with the peer | Rejected –  SPP A-MSDU capability is observable at the STA. According to subclause10.19@REVmc D2.0: “On either association or reassociation, the associating STA and its peer STA both determine and maintain a record of whether an encrypted A-MSDU sent to its peer is to be a PP A-MSDU or an SPP A-MSDU based on the value of the SPP A-MSDU Capable and SPP A-MSDU Required subfields of the RSN Capabilities field of the RSNE (see 8.4.2.24.4 (RSN capabilities)).” |
| 1996 | 234.20 | 11.4.3.3.3 | its peer has the SPP A-MSDU Capable field equal to 0' is not something observable at the STA | Use some information that is currently available at the encrypting STA and has been authenticated i.e. some information that was obtained during security negotiation with the peer | Rejected –  SPP A-MSDU capability is observable at the STA. According to subclause10.19@REVmc D2.0: “On either association or reassociation, the associating STA and its peer STA both determine and maintain a record of whether an encrypted A-MSDU sent to its peer is to be a PP A-MSDU or an SPP A-MSDU based on the value of the SPP A-MSDU Capable and SPP A-MSDU Required subfields of the RSN Capabilities field of the RSNE (see 8.4.2.24.4 (RSN capabilities)).” |
| 2644 | 233.25 | 11.4.3.3.3 | What are the security requirements for the new short management frames? The AAD needs to be specified for short management frames as well as for short data frames. | Clarify the security requirements for the new short management frames and specifiy the AAD construction details if necessary. | Agree in principle with the commenter regarding the short management frames. The proposed resolution is to clarify security requirements for short management frames.  Revised –  TGah editor to make changes shown in 14/0080r0 under the heading for CIDs from 1551 to 2487. |
| 2815 | 233.25 | 11.4.3.3.3 | If new frames such as the short Beacon are to be robust then the AAD needs to be specified for short management frames as well as for short data frames. | Review security requirements for the new short management frames and add details for AAD construction if necessary. | Agree in principle with the commenter regarding the short management frames. The proposed resolution is to clarify security requirements for short management frames.  Revised –  TGah editor to make changes shown in 14/0080r0 under the heading for CIDs from 1551 to 2487. |
| 2816 | 234.05 | 11.4.3.3.3 | AAD construction for short data frames does not match the frame control field in Figure 8-532b | Review short MAC header for data frames and make AAD construction consistent with that. | Agree with the commenter. Proposed resolution is to fix the inconsistency.  Revised –  TGah editor to make changes shown in 14/0080r0 under the heading for CIDs from 1551 to 2487. |
| 1643 | 234.37 | 11.4.3.3.4 | Does the nonce construction use the SID or somehow fabricate the entire A2 address, if A2 is not supplied by the higher layer? | Specify whether SID or complete A2 is used in construction of AAD. | Rejected –  No changes are needed as it is already specified that the nonce construction uses the 48-bit “STA MAC address identified by A2” as shown in Figure 11-19. |
| 1644 | 234.55 | 11.4.3.3.4 | FC TID bits are not bits 13-15. | Review which bits are included in nonce construction. | Agree with the commenter. Proposed resolution is to fix the inconsistency.  Revised –  TGah editor to make changes shown in 14/0080r0 under the heading for CIDs from 1551 to 2487. |
| 1997 | 234.47 | 11.4.3.3.4 | Old text needs to be properly conditionalized | Modify old text to be conditional on protocol version 0 | Agree with the commenter. Proposed resolution is to fix the inconsistencies.  Revised –  TGah editor to make changes shown in 14/0080r0 under the heading for CIDs from 1551 to 2487. |
| 2487 | 234.47 | 11.4.3.3.4 | "Type field of the Frame Control field is 10 (Data frame)" -- this is meaningless if the Protocol Version field is not 0 | Find all places in the baseline which refer to Type/Subtype and additionally qualify them with the PV being 0 | Agree with the commenter.  Revised –  TGah editor to make changes shown in 14/0080r0 under the heading for CIDs from 1551 to 2487. |

**Discussion:** *For further details refer to 11-13-xxxx-00-00ah-ccmp-header-compression.*

**Proposed Changes:**

**Instruction to TGah Editor: *Add the following definitions in subclause 3.2 (@REVmc D2.0):***

**3.2 Definitions specific to IEEE Std 802.11**

**Protocol Version 0 (PV0) MPDU:** An MPDU with the Protocol Version field of the Frame Control field of the MPDU header equal to 0.

**Protocol Version 1 (PV1) MPDU:** An MPDU with the Protocol Version field of the Frame Control field of the MPDU header equal to 1.

**Instruction to TGah Editor: *Add the following acronyms in subclause 3.3 (@REVmc D2.0):***

**3.3 Abbreviations and acronyms**

BPN base packet number

PV0 Protocol Version 0

PV1 Protocol Version 1

SC Sequence Counter

* **CTR with CBC-MAC Protocol (CCMP)**
* **CCMP MPDU format**

Instruction to TGah Editor: Change the length of the CCMP header field in Figure 11-16 from "8 octets" to "0 or 8 octets" (@REVmc D2.0).

**Instruction to TGah Editor: *Change the 2nd paragraph as follows (@802.11ac D5.0):***

For secure PV0 MPDUs, when used with a 128-bit key, CCMP processing expands the original MPDU size by 16 octets, 8 octets for the CCMP Header field and 8 octets for the MIC field. When used with a 256-bit key, CCMP 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.

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 processing expands the original MPDU size by 8 octets for the MIC field when used with a 128-bit key, or by 16 octets for the MIC field when used with a 256-bit key. Figure 11-16a (Expanded PV1 CCMP MPDU) depicts the PV1 MPDU when using CCMP.



**Figure 11-16a—Expanded PV1 CCMP MPDU**

Note that CCMP does not use the WEP ICV.

**Instruction to TGah Editor: *Add a new subclause after subclause 11.4.3.2(@REVmc D2.0):***

**11.4.3.2a Construction of the CCMP Header for PV1 MPDUs**

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
    - 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.

When no Block Ack is used or when decryption occurs after Block Ack reordering, 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.

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 + 2^12

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 9.40a (Header Compression procedure).

* **CCMP cryptographic encapsulation**

**11.4.3.3.1 General**

**Instruction to TGah Editor: *Modify this subclause as follows (@REVmcD2.0):***

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 field of the MPDU where A2 is MPDU Address 2.
* 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).

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

1. 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, 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. Note that retransmitted MPDUs are not modified on retransmission.
2. 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.
3. Construct the CCMP header as defined in 11.4.3.2a (Construction of the CCMP header for PV1 MPDUs).
4. 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.
5. 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.

f) 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**

**Instruction to TGah Editor: *Change the 1st paragraph as follows (@REVmcD2.0):***

The PN is incremented by a positive number for each MPDU. For PV0 MPDUs, the PN shall never repeat for a series of encrypted MPDUs using the same temporal key. For PV1 MPDUs, the PN shall never repeat for a series of encrypted MPDUs using the same temporal key and TID.

* **Construct AAD**

**Instruction to TGah Editor: *Change these paragraphs as follows (@REVmcD2.0):***

For PV0 MPDUs, the 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).

|  |  |  |
| --- | --- | --- |
| * **AAD length for PV0 MPDUs** | | |
| **QC field** | **A4 field** | **AAD length (octets)** |
| Absent | Absent | 22 |
| Present | Absent | 24 |
| Absent | Present | 28 |
| Present | Present | 30 |

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(#130) 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:

* FC – MPDU Frame Control field, with
* Subtype bits (bits 4 5 6) in a Data MPDU masked to 0
* Retry bit (bit 11) masked to 0
* Power Management bit (bit 12) masked to 0
* More Data bit (bit 13) masked to 0
* Protected Frame bit (bit 14) always set to 1
* Order bit (bit 15) as follows:
* Masked to 0 in all (#100)Data MPDUs containing a QoS Control field
* Unmasked otherwise
* A1 – MPDU Address 1 field.
* A2 – MPDU Address 2 field.
* A3 – MPDU Address 3 field.
* 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.
* A4 – MPDU Address field, if present.

QC – QoS Control field, if present, a 2-octet field that includes the MSDU priority. The QC TID is used in the construction of the AAD. When in a non-DMG BSS and(11ad) both the STA and its peer have their SPP A‑MSDU Capable fields equal to 1, bit 7 (the A‑MSDU Present field) is used in the construction of the AAD. The remaining QC fields are masked to 0 for the AAD calculation (bits 4 to 6, bits 8 to 15, and bit 7 when either the STA or its peer has the SPP A‑MSDU Capable field equal to 0). When in a DMG BSS, the A‑MSDU Present bit 7 and A‑MSDU Type bit 8 are used in the construction of the AAD, and the remaining QC fields are masked to 0 for the AAD calculation (bits 4 to 6, bits 9 to 15).

**Instruction to TGah Editor: *Change the inserted paragraphs as follows:***

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** | | | | | | |

(#870)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, the AAD construction is performed as follows:

* FC – MPDU Frame Control field, with
* Type bits (bits3 ) 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 (bit13) masked to 0
* Relayed Frame bit (bit 14) masked to 0

7) Ack Policy bit (bit 15) masked to 0

* A1 –MPDU Address 1 field.
* When the SID field is present as the A1 field

a) A3 Present bit (bit 13) of SID field masked to 0

b) 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

a) A3 Present bit (bit 13) of SID field masked to 0

b) 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 Figure 11-19 and 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** | | | |

**Instruction to TGah Editor: *Modify Figure 11.20 as shown below (@REVmcD2.0):***



**Figure 11-20—Nonce Flags subfield**

**Instruction to TGah Editor: *Change the 2nd paragraph as follows:***

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

* For PV0 MPDUs, if the Type field of the Frame Control field is 10 (Data frame) and there is a QC field present in the MPDU header, bits 0 to 3 of the Priority subfield of the Nonce Flags field shall be set to the value of the QC 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 subfield of the Nonce Flags field shall be set to the value in the ACI subfield of the Sequence Number field. Otherwise, the Priority subfield of the Nonce Flags field shall be set to the fixed value 0.(11ae) For PV1 MPDUs, if the Type field of the Frame Control field is 000 (Data frame) or 011 (Data frame), bits 1 to 3 of the Priority subfield shall be set to the value of the PTID/Subtype field (bits 5-7 of the Frame Control field). If the Type field of the Frame Control field is 001 (Management frame) and the frame is a QMF, the Priority subfield of the Nonce Flags shall be set to the value in the ACI subfield of the Sequence Number field. Otherwise the Priority subfield of the Nonce Flags field shall be set to the fixed value 0.
* 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 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**

**Instruction to TGah Editor: *Change the 2nd paragraph as follows (@REVmcD2.0):***

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 (12–30 octets) constructed from the MPDU header as described in 11.4.3.3.3 (Construct AAD).
* **General**

**Instruction to TGah Editor: *Modify this clause as shown below (@REVmcD2.0):***

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.

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

1. The encrypted MPDU is parsed to construct the AAD and nonce values.
2. The CCMP header is constructed as defined in 11.4.3.2a (Construction of the CCMP header for PV1 MPDUs).
3. The AAD is formed from the MPDU header of the encrypted MPDU.
4. The Nonce value is constructed from the STA MAC Address identified by A2, PN, and Nonce Flags fields.
5. The MIC is extracted for use in the CCM integrity checking.
6. 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.
7. The received MPDU header and the MPDU plaintext data from the CCM recipient processing are concatenated to form a plaintext MPDU.
8. 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.

See 11.4.3.4.2 (CCM recipient processing) to 11.4.3.4.4 (PN and replay detection) for details of this processing.

When the received frame is a CCMP protected individually addressed robust (#100)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 (#100)Management frame or Short Management rather than through the MA-UNITDATA.indication primitive.

* **CCM recipient processing**

**Instruction to TGah Editor: *Modify this clause as shown below (@REVmcD2.0):***

CCM recipient processing uses the same parameters as CCM originator processing. A CCMP protected individually addressed robust (#100)Management frame or Short Management frame shall use the same TK as a Data MPDU or Short Data MPDU.

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 an 8-octet MIC.
* *AAD:* the AAD (12–30 octets) that is the canonical MPDU header as described in 11.4.3.3.3 (Construct AAD).

The CCM recipient processing checks the authentication and integrity of the frame body and the AAD as well as decrypting the frame body. The plaintext is returned only if the MIC check is successful.

There is one output from error-free CCM recipient processing:

* *Frame body:* the plaintext frame body, which is 8 octets smaller than the encrypted frame body.
* **PN and replay detection**

**Instruction to TGah Editor: *Modify this clause as shown below (@REVmcD2.0):***

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. 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, 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, STKSA, and Protocol Version value, the recipient shall maintain a separate replay counter for each IEEE Std(#130) 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(#130) 802.11 MSDU or A‑MSDU priorities without ensuring that the receiver supports the required number of replay counters. The transmitter shall not reorder frames within a replay counter, but may reorder frames across replay counters. One possible reason for reordering frames is the IEEE Std(#130) 802.11 MSDU or A‑MSDU priority.
* If dot11RSNAProtectedManagementFramesActivated is true, the recipient shall maintain a single replay counter for received individually addressed robust (#100)Management frames that are received with the To DS field equal to 0(#1071)(11ae), 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(#1071). 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.(11ae) 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.(11ae) The transmitter shall preserve the order of protected robust (#100)Management frames and Short Management frames that are transmitted(11ae) 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(11ae).
* If dot11RSNAProtectedManagementFramesActivated is true and dot11MeshSecurityActivated is true, the recipient shall maintain a single replay counter for received group addressed robust (#100)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 (#100)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.(11ae)
* The receiver shall discard MSDUs(#211) 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 (#100)Data frames or dot11RSNAStatsRobustMgmtCCMPReplays for robust (#100)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 9.22.4 (Receive buffer operation)) is performed prior to replay detection.
* **Format of Short Management frames**

**Instruction to TGah Editor: *Insert the following sentence at the end of the forth paragraph as shown below:***

The Sequence Control field (see 8.2.4.4 (Sequence Control field)) is present in all types of management frames unless stated otherwise.