IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GCMP decapsulation | | | | |
| Date: 2015-09-16 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Jouni Malinen | Qualcomm |  |  | jouni@qca.qualcomm.com |
| Mark RISON | Samsung Cambridge Solution Centre |  |  |  |

Abstract

Number of comments in REVmc/D4.0 SB comments pointed out differences between the descriptions of CCMP and GCMP decapsulation description. While some differences may be appropriate, most of these are neither necessary nor desired. This contribution proposes changes to align description of CCMP and GCMP decapsulation steps and address the identified issues in the SB comments CIDs 6024, 6239, 6240, 6393, 6564. In addition to the issues identified in those comments, the proposed cleanup moves transmitter requirements into the appropriate subclauses and requirements related to BIP into the BIP subclauses.

r1: Comments from Mark Rison: additional cleanup to CCMP, BIP, GCMP

r2: Fix ambiguous BIP-CMAC-256 MIC length description and further cleanup based on comments from Mark Rison

**Comments**

CID 6024

11.4.5.4.4 1925 1 1925,01 1

Comment:

The PN and replay detection rules for GCMP have been specified in a way that is significantly different from the style used for CCMP. This is unfortunate since number of the rules are actually supposed to be identical due to the shared AAD design. It looks like one of the important steps for the security of the design has been lost, i.e., GCMP does not protect against an attack related to fragmented frames while CCMP does. The key missing requirement for GCMP is this step from 11.4.3.4.4 (same subclause for CCMP): "h) The receiver shall discard MSDUs and MMPDUs whose constituent MPDU PN values are not sequential."

It should be noted that the specific use of "sequential" here requires the PN to be incremented exactly by one (i.e., not skipping any PN). Without that interpretation, this would not protect against the attack where an attacker picks MPDUs from two different fragmented MSDU/MMPDU and replaces the sequence number (not protected by AAD) to get the recipient accept these as an MSDU/MMPDU even though the data is from two different MSDU/MMPDU. Unfortunately, the other occurrences of "sequential" in both of these "PN and replay detection" subclauses need to have different interpretation.. It would be good to reword these subclauses to address this small, but important, difference.

For background history on this special requirement for fragmented frames:

- text added for CCMP: https://mentor.ieee.org/802.11/dcn/03/11-03-0118-02-000i-alternate-text-for-tgi-8-3-4.doc

- motion on Slide 13 of https://mentor.ieee.org/802.11/dcn/03/11-03-0092-04-000i-ccmp-reorganization.ppt

- minutes: pages 30-32 of http://grouper.ieee.org/groups/802/11/Minutes/Cons\_Minutes\_Jan-2003.pdf

The specific attack:

Say, there are two MSDUs, each with two MPDUs:

MSDU\_a(MPDU\_a1, MPDU\_a2), MSDU\_b(MPDU\_b1,MPDU\_b2). An attacker prevents a STA from seeing MPDU\_a2 and MPDU\_b1 and MPDU\_b2… and replays MPDU\_b2 with SeqNum changed to be that of MPDU\_a1. Without this "no-skipping-PNs-within-MSDU" rule, the recipient would have accepted the invalid MSDU.

Since GCMP has the same AAD design (which does not protect seq#) as CCMP, it needs the same protection step for this case.

Proposed Change:

Add "The receiver shall discard MSDUs and MMPDUs whose constituent MPDU PN values are not sequential." to 11.4.5.4.4.

CID 6239

11.4.5.4.4 1924 60 1924,60 60

Comment:

11.4.3.4.4 re CCMP PN and replay detection says that "The receiver shall discard MSDUs and MMPDUs whose constituent MPDU PN values are not sequential.", but no such statement appears here for GCMP

Proposed Change:

Add such a statement here

CID 6240

11.4.5.4.4 1924 60 1924,60 60

Comment:

The CCMP version of this subclause seems markedly different

Proposed Change:

Align the two subclauses, since they should be essentially the same

CID 6393

11.4.5.4.4 1924 60 1924,60 60

Comment:

"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." appears for CCMP but not for GCMP. More generally, CCMP and GCMP do not seem quite aligned

Proposed Change:

Align GCMP and CCMP

CID 6564

(no page/line or clause identified; 11.4.3.4.4 g is at P1917 L25; CCMP PN and replay detection)

Comment:

What exactly does "sequential" mean? It seems to be intended to mean "increasing in steps of 1" but might be interpreted as just "increasing"

Proposed Change:

Clarify this, especially for 11.4.3.4.4.g

**Discussion**

CID 6024 provides references to understand the history of how the CCMP and GCMP descriptions ended up in their current form. The split history of the descriptions has then continued through number of amendments and some of the required changes have been made only to one of the sections. Furthermore, transmitter rules have been added in the decapsulation subclauses which are not the correct places for those; such sentences are proposed to be moved into the frame originator subclause.

Moving the BIP related items into the BIP clauses ended up including additional cleanup and alignment of BIP definition and that effort discovered an issue introduced in P802.11ac/D7.0 for BIP-CMAC-256. That cipher was initially added in P802.11ac/D6.0 based on submission 13/561r0 which is clearly defining the MME MIC to be 16 octets. However, that document forgot to update the BIP overview clause and the later change in D7.0 updated that clause incorrectly (truncate MIC to 8 octets). The other locations in the standard indicate this to be the originally intended length (16 octets). The proposed changes in this document make that consistent by fixing the BIP overview clause to match the other places.

Proposed Resolution:

Revised. Make the changes proposed in <this document> (Proposed Changed to REVmc/D4.0 section). This aligns the description of GCMP decapsulation with CCMP decapsulation and by doing so, adds the previously missing rules for fragmented frames. In addition, the rules related to transmitter rules are moved to the originator subclauses and the rules related to group-addressed MMPDUs to the BIP subclauses.

**Proposed Changes to REVmc/D4.0**

*Note to editor: These changes are on top of the Draft P802.11REVmc\_D4.2\_rtfs.zip*

*Change 11.4.3.3.6 as indicated*

**11.4.3.3.6 CCM originator processing**

CCM is a generic authenticate-and-encrypt block cipher mode, and in this standard, CCM is used with the AES block cipher.

There are four inputs to CCM originator processing:

a) *Key:* the temporal key (16 octets).

b) *Nonce:* the nonce (13 octets) constructed as described in 11.4.3.3.4 (Construct CCM nonce).

c) *Frame body:* the plaintext frame body of the MPDU.

d) *AAD:* the AAD (22–30 octets) constructed from the MPDU header as described in 11.4.3.3.3 (Construct AAD).

The CCM originator processing provides authentication and integrity of the frame body and the AAD as well as data confidentiality of the frame body. The output from the CCM originator processing consists of the encrypted data and an encrypted MIC (see Figure 11-16 (Expanded CCMP MPDU)).

The PN values sequentially number each MPDU. Each transmitter shall maintain a single PN (48-bit counter) for each PTKSA, GTKSA, and STKSA. The PN shall be implemented as a 48-bit strictly increasing integer, initialized to 1 when the corresponding temporal key is initialized or refreshed.

-RA

A CCMP protected individually addressed robust Management frame shall be protected with the TK.

*Change 11.4.3.4 as indicated*

**11.4.3.4 CCMP decapsulation**

**11.4.3.4.1 General**

Figure 11-21 (CCMP decapsulation block diagram) depicts the CCMP decapsulation process.

CCMP decrypts the payload of a cipher text MPDU and decapsulates a plaintext MPDU using the following steps:

a) The encrypted MPDU is parsed to construct the AAD and nonce values.

b) The AAD is formed from the MPDU header of the encrypted MPDU.

c) The Nonce value is constructed from the A2, PN, and Nonce Flags fields.

d) The MIC is extracted for use in the CCM integrity checking.

e) 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.

f) The received MPDU header and the MPDU plaintext data from the CCM recipient processing are concatenated to form a plaintext MPDU.

g) 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.

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 Management frame, contents of the MMPDU body after protection is removed shall be delivered to the SME via the MLME primitive designated for that MMPDU rather than through the MA-UNITDATA.indication primitive.

**11.4.3.4.2 CCM recipient processing**

CCM recipient processing uses the same parameters as CCM originator processing. A CCMP protected individually addressed robust Management frame shall use the same TK as a Data 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 (22–30 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 (CCMP-128) or 16 octets (CCMP-256) smaller than the encrypted frame body.

**11.4.3.4.3 Decrypted CCMP MPDU**

The decapsulation process succeeds when the calculated MIC matches the MIC value obtained from decrypting the received encrypted MPDU. The original MPDU header is concatenated with the plaintext data resulting from the successful CCM recipient processing to create the plaintext MPDU.

**11.4.3.4.4 PN and replay detection**

To effect replay detection, the receiver extracts the PN from the CCMP header. 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:

a) The receiver shall maintain a separate set of replay counters for each PTKSA, GTKSA, and STKSA. 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.

b) For each PTKSA, GTKSA, and STKSA, the recipient shall maintain a separate replay counter for each TID, subject to the limitation of the number of supported replay counters indicated in the RSN Capabilities field (see 8.4.2.24 (RSNE)), and shall use the PN in a received frame to detect replayed frames. 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.

c) 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 subfield equal to 0 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 that are received with the To DS subfield 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 management frame replay counter that corresponds to the ACI of the frame.

d) The receiver shall discard any Data frame that is received with its PN less than or equal to the value of the replay counter that is associated with the TA and priority value of the received MPDU. The receiver shall discard MSDUs and MMPDUs whose constituent MPDU PN values are not incrementing in steps of 1. If dot11RSNAProtectedManagementFramesActivated is true, the receiver shall discard any individually addressed robust Management frame that is received with its PN less than or equal to the value of the replay counter associated with the TA of that individually addressed Management frame.

e) When discarding a frame, the receiver shall increment by 1 the value of dot11RSNAStatsCCMPReplays for Data frames or dot11RSNAStatsRobustMgmtCCMPReplays for robust Management frames.

f) 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 9.24.4 (Receive buffer operation)) is performed prior to replay detection.

*Change 11.4.4 as indicated*

**11.4.4.1 BIP overview**

BIP provides data integrity and replay protection for group addressed robust Management frames after successful establishment of an IGTKSA (see 11.5.1.1.9 (IGTKSA)).

BIP-CMAC-128 provides data integrity and replay protection, using AES-128 in CMAC Mode with a 128-bit integrity key and a CMAC TLen value of 128 (16 octets). BIP-CMAC-256 provides data integrity and replay protection, using AES-256 in CMAC Mode with a 256-bit integrity key and a CMAC TLen value of 128 (16 octets). NIST Special Publication 800-38B defines the CMAC algorithm, and NIST SP 800-38D defines the GMAC algorithm. BIP processing uses AES with a 128-bit or 256-bit integrity key and a CMAC TLen value of 128 (16 octets). The CMAC output for BIP-CMAC-256 is not truncated and shall be 128 bits (16 octets). The CMAC output for BIP-CMAC-128 is truncated to 64 bits:

MIC = L(CMAC Output, 0, 64).

BIP-GCMP-128 uses AES with a 128-bit integrity key, and BIP-GCMP-256 uses AES with a 256-bit integrity key. The authentication tag for both BIP-GCMP-128 and BIP-GCMP-256 is not truncated and shall be 128 bits (16 octets).

BIP uses the IGTK to compute the MMPDU MIC. The authenticator shall distribute one new IGTK and IGTK PN (IPN) whenever it distributes a new GTK. The IGTK is identified by the MAC address of the transmitting STA plus an IGTK identifier that is encoded in the MME Key ID field.

**11.4.4.4 BIP replay protection**

The MME Sequence Number field represents a sequence number whose length is 6 octets.

When management frame protection is negotiated, the receiver shall maintain a 48-bit replay counter for each IGTK. The receiver shall set the receive replay counter to the value of the IPN in the IGTK key data encapsulation (KDE) (see 11.6.2 (EAPOL-Key frames)) provided by the Authenticator in either the 4-way handshake, FT 4-way handshake, FT handshake, or group key handshake. The transmitter shall maintain a single IPN for each IGTK. The IPN shall be implemented as a 48-bit strictly increasing integer, initialized to 1 when the corresponding IGTK is initialized. The transmitter may reinitialize the sequence counter when the IGTK is refreshed. See 11.4.4.5 (BIP transmission) and 11.4.4.6 (BIP reception) for per packet BIP processing.

NOTE—When the IPN space is exhausted, the choices available to an implementation are to replace the IGTK or to end communications.

When dot11QMFActivated is true, the receiver shall maintain an additional replay counter for each ACI for received group addressed robust Management frames that use QMF. The receiver shall use the ACI encoded in the Sequence Number field of received GQMFs protected by BIP to select the replay counter to use for the received frame, and shall use the IPN from the received frame to detect replays.

**11.4.4.5 BIP transmission**

When a STA transmits a protected group addressed robust Management frame, it shall

a) Select the IGTK currently active for transmission of frames to the intended group of recipients and construct the MME (see 8.4.2.54 (Management MIC element)) with the MIC field masked to 0 and the Key ID field set to the corresponding IGTK Key ID value. If the frame is not a GQMF, the transmitting STA shall insert a strictly increasing integer into the MME IPN field. If the frame is a GQMF, then the transmitting STA shall maintain a 48-bit counter for use as the IPN, the counter shall be incremented for each GQMF until the two least significant bits of the counter match the ACI of the AC that is used to transmit the frame, and the counter value shall be inserted into the MME IPN field of the frame. For BIP-GMAC-128 and BIP-GMAC-256, the initialization vector passed to GMAC shall be a concatenation of Address 2 from the MAC header of the MPDU and the non-negative integer inserted into the MMP IPN field.

b) Compute AAD as specified in 11.4.4.3 (BIP AAD construction).

c) Compute an integrity value over the concatenation of AAD and the management frame body including MME, and insert the output into the MME MIC field. For BIP-CMAC-128, the integrity value is 64 bits and is computed using AES-128-CMAC; for BIP-CMAC-256, the integrity value is 128 bits and is computed using AES-256-CMAC; for BIP-GMAC-128, the integrity value is 128 bits and is computed using AES-128-GMAC; and, for BIP-GMAC-256, the integrity value is 128 bits and is computed using AES-256-GMAC.

d) Compose the frame as the IEEE Std 802.11 header, management frame body, including MME, and FCS. The MME shall appear last in the frame body.

e) Transmit the frame.

**11.4.4.6 BIP reception**

When a STA with management frame protection negotiated receives a group addressed robust Management frame protected by BIP-CMAC-128, BIP-CMAC-256, BIP-GMAC-128, or BIP-GMAC-256, it shall

a) Identify the appropriate IGTK key and associated state based on the MME Key ID field. If no such IGTK exists, silently drop the frame and terminate BIP processing for this reception.

b) Perform replay protection on the received frame. The receiver shall interpret the MME IPN field as a 48-bit unsigned integer.

1) If the frame is not a GQMF, the receiver shall compare this MME IPN integer value to the value of the receive replay counter for the IGTK identified by the MME Key ID field. If the integer value from the received MME IPN field is less than or equal to the replay counter value for this IGTK, the receiver shall discard the frame and increment the dot11RSNAStatsCMACReplays counter by 1.

2) If the frame is a GQMF, the receiver shall compare this MME IPN integer value to the value of the receive replay counter for the IGTK identified by the MME Key ID field and the AC represented by the value of the ACI subfield of the received frame. If the integer value from the received MME IPN field is less than or equal to the replay counter value for this IGTK and AC, the receiver shall discard the frame and increment the dot11RSNAStatsCMACReplays counter by 1.

c) Compute AAD for this Management frame, as specified in 11.4.4.3 (BIP AAD construction). For BIP-GMAC-128 and BIP-GMAC-256, an initialization vector for GMAC is constructed as the concatenation of Address 2 from the MAC header of the MPDU and the 48-bit unsigned integer from the MME IPN field.

d) Extract and save the received MIC value, and compute the a verifier over the concatenation of AAD, the management frame body and MME, with the MIC field masked to 0 in the MME. For BIP-CMAC-128, the integrity value is 64 bits and is computed using AES-128-CMAC; for BIP-CMAC-256, the integrity value is 128 bits and is computed using AES-256-CMAC; for BIP-GMAC-128, the integrity value is 128 bits and is computed using AES-128-GMAC; and, for BIP-GMAC-256, the integrity value is 128 bits and is computed using AES-256-GMAC. If the result does not match the received MIC value, then the receiver shall discard the frame and increment the dot11RSNAStatsCMACICVErrors counter by 1, and terminate BIP processing for this reception.

e) If the frame is not a GQMF, update the replay counter for the IGTK identified by the MME Key ID field with the integer value of the MME IPN field.

f) If the frame is a GQMF, update the replay counter for the IGTK identified by the MME Key ID field and the AC represented by the value of the ACI subfield of the received frame with the integer value of the MME IPN field if the frame is a GQMF.

If management frame protection is negotiated, group addressed robust Management frames that are received without BIP protection shall be discarded.

*Change 11.4.5.3.6 as indicated*

**11.4.5.3.6 GCM originator processing**

GCM is a generic authenticate-and-encrypt block cipher mode, and in this standard, GCM is used with the AES block cipher.

There are four inputs to GCM originator processing:

*Key:* the temporal key (16 octets).

*Nonce:* the nonce (12 octets) constructed as described in 11.4.5.3.4 (Construct GCM nonce).

*Frame body:* the plaintext frame body of the MPDU.

*AAD:* the AAD (22-30 octets) constructed from the MPDU header as described in 11.4.5.3.3 (Construct AAD).

The GCM originator processing provides authentication and integrity of the frame body and the AAD as well as data confidentiality of the frame body. The output from the GCM originator processing consists of the encrypted data and 16 additional octets of encrypted MIC (see Figure 11-24).

The PN values sequentially number each MPDU. Each transmitter shall maintain a single PN (48-bit counter) for each PTKSA, GTKSA, and STKSA. The PN shall be implemented as a 48-bit strictly increasing integer, initialized to 1 when the corresponding temporal key is initialized or refreshed.

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 RA 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.

The transmitter shall preserve the order of protected robust 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.

A GCMP protected individually addressed robust Management frame shall be protected with the TK.

*Change 11.4.5.4 as indicated*

**11.4.5.4 GCMP decapsulation**

**11.4.5.4.1 General**

Figure 11-27 (GCMP decapsulation block diagram) shows the GCMP decapsulation process.

GCMP decrypts the payload of a cipher text MPDU and decapsulates a plaintext MPDU using the following steps:

a) The encrypted MPDU is parsed to construct the AAD and nonce values.

b) The AAD is formed from the MPDU header of the encrypted MPDU.

c) The Nonce value is constructed from the A2 and PN fields.

d) The MIC is extracted for use in the GCM integrity checking.

e) The GCM 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.

f) The received MPDU header and the MPDU plaintext data from the GCM recipient processing are concatenated to form a plaintext MPDU.

g) 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.

See 11.4.5.4.2 (GCM recipient processing) to 11.4.5.4.4 (PN and replay detection) for details of this processing.

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

**11.4.5.4.2 GCM recipient processing**

GCM recipient processing shall use the same parameters as GCM originator processing. A GCMP protected individually addressed robust Management frame shall use the same TK as a Data frame.

There are four inputs to GCM recipient processing:

*— Key:* the temporal key (16 octets).

*— Nonce:* the nonce (12 octets) constructed as described in 11.4.5.3.4 (Construct GCM nonce).

*— Encrypted frame body:* the encrypted frame body from the received MPDU. The encrypted frame body includes a 16-octet MIC.

*— AAD*: the AAD (22-30 octets) that is the canonical MPDU header as described in 11.4.5.3.3 (Construct AAD).

The GCM 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 GCM recipient processing:

*— Frame body:* the plaintext frame body, which is 16 octets smaller than the encrypted frame body.

**11.4.5.4.3 Decrypted GCMP MPDU**

The decapsulation process succeeds when the calculated MIC matches the MIC value obtained from decrypting the received encrypted MPDU. The original MPDU header is concatenated with the plaintext data resulting from the successful GCM recipient processing to create the plaintext MPDU.

**11.4.5.4.4 PN and replay detection**

To effect replay detection, the receiver extracts the PN from the GCMP header. See 11.4.5.2 (GCMP MPDU format) for a description of how the PN is encoded in the GCMP header. The following processing rules are used to detect replay:

a) The receiver shall maintain a separate set of replay counters for each PTKSA, GTKSA, and STKSA. 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 GCMP MPDUs.

b) For each PTKSA, GTKSA, and STKSA, the recipient shall maintain a separate replay counter for each TID, subject to the limitation of the number of supported replay counters indicated in the RSN Capabilities field (see 8.4.2.24 (RSNE)), and shall use the PN in a received frame to detect replayed frames. 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.

c) 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 subfield equal to 0 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 that are received with the To DS subfield 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 management frame replay counter that corresponds to the ACI of the frame.

d) The receiver shall discard any Data frame that is received with its PN less than or equal to the value of the replay counter that is associated with the TA and priority value of the received MPDU. The receiver shall discard MSDUs and MMPDUs whose constituent MPDU PN values are not incrementing in steps of 1. If dot11RSNAProtectedManagementFramesActivated is true, the receiver shall discard any individually addressed robust Management frame that is received with its PN less than or equal to the value of the replay counter associated with the TA of that individually addressed Management frame.

e) When discarding a frame, the receiver shall increment by 1 the value of dot11RSNAStatsGCMPReplays for Data frames or dot11RSNAStatsRobustMgmtGCMPReplays for robust Management frames.

f) 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 9.24.4 (Receive buffer operation)) is performed prior to replay detection.