IEEE P802.11
Wireless LANs

|  |
| --- |
| Various REVme SA recirc comments |
| Date: 2024-04-16 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Jouni Malinen | Qualcomm Technologies, Inc. |  |  | jouni@qca.qualcomm.com |
|  |  |  |  |  |

Abstract

This document proposes comment resolutions to the following REVme/D5.0 recirculation SA ballot comments:

CID 7005, 7006, 7007, 7025, 7027, 7029, 7032, 7033

CID 7028: See 342r1 instead.

Rev 2: Address TODO items for CID 7007.

# CID 7032 (SEC)

Clause Number: 12.13.5 Page: 3163 Line: 26

Comment:

PASN is defined to be restricted to using the original SAE AKMP 00-0F-AC:8. There does not seem to be any good reason for this other than the new AKMP having been defined in parallel to the P802.11az work. REVme should extend PASN to cover the AKMPs that have been recently added.

Proposed Change:

At P3163 L26, replace "SAE AKMP 00-0F-AC:8" with "SAE AKMP 00-0F-AC:8 or 00-0F-AC:24".

Proposed Resolution:

REVISED - Incorporate changes under the “Proposed changes for CID 7032” section of <this doc>.

### Discussion

Context and proposed change are shown below with change tracking.

Table 9-190—AKM suite selectors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 00-0F-AC  | 8  | SAE authentication  | RSNA key management as defined in 12.7 (Keys and key distribution), or authenticated mesh peering exchange as defined in 14.6 (Authenticate d mesh peering exchange (AMPE))  | Defined in 12.7.1.6.2 (Key derivation function (KDF))using the hash algorithm specified in 12.4.2 (Assumptions on SAE)  | 3 (SAE) for SAE Authentication 0 (open) for PMKSA caching  | None  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 00-0F- AC  | 24  | SAE authentication  | RSNA key management as defined in 12.7 (Keys and key distribution), or authenticated mesh peering exchange as defined in 14.6 (Authenticate d mesh peering exchange (AMPE))  | Defined in 12.7.1.6.2 (Key derivation function (KDF))using the hash algorithm specified in 12.4.2 (Assumptions on SAE)  | 3 (SAE) for SAE Authentication0 (open) for PMKSA caching  | None  |
|  |  |  |  |  |  |  |

### Proposed changes for CID 7032

**12.13.5 PASN authentication with SAE**

*Modify 12.13.5 as shown (REVme-D5.0 P3163 L26):*

This subclause specifies aspects of PASN authentication when AKM 00-0F-AC:8 or 00-0F-AC:24 is used as the Base AKMP when PMK caching is not used. When PMK caching is used PASN authentication relies on the PMKSA already established by SAE protocol.

# CID 7033 (SEC)

Clause Number: 12.13.6 Page: 3164 Line: 39

Comment:

PASN is defined to be restricted to using a subset of FT AKMPs. There does not seem to be any good reason for this other than a new FT AKMP having been defined in parallel to the P802.11az work. REVme should extend PASN to cover the AKMPs that have been recently added.

Proposed Change:

At P3164 L39, replace "FT AKMPs 00-0F-AC: [3, 4, 13, 19]" with "FT AKMPs 00-0F-AC: [3, 4, 13, 19, 22]".

Proposed Resolution:

REVISED - Incorporate changes under the “Proposed changes for CID 7033” section of <this doc>.

### Discussion

Context and proposed change are shown below with change tracking.

Table 9-190—AKM suite selectors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 00-0F-AC  | 3  | FT authentication negotiated over IEEE Std 802.1X  | FT key management as defined in 12.7.1.6 (FT key hierarchy)  | Defined in 12.7.1.6.2 (Key derivation function (KDF)) using SHA- 256  | 2 (FT) for FT protocol reassociation as defined in 13.5 (FT protocol) 0 (open) for FT Initial Mobility Domain Association over IEEE Std 802.1X or PMKSA caching  | None  |
| 00-0F-AC  | 4  | FT authentication using PSK  | FT key management as defined in 12.7.1.6 (FT key hierarchy)  | Defined in 12.7.1.6.2 (Key derivation function (KDF)) using SHA- 256  | 2 (FT) for FT protocol reassociation as defined in 13.5 (FT protocol) 0 (open) for FT Initial Mobility Domain Association using PSK  | None  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 00-0F-AC  | 13  | FT authentication negotiated over IEEE Std 802.1X  | FT key management as defined in 12.7.1.6 (FT key hierarchy)  | Defined in 12.7.1.6.2 (Key derivation function (KDF)) using SHA- 384  | 2 (FT) for FT protocol reassociation as defined in 13.5 (FT protocol) 0 (open) for FT Initial Mobility Domain Association over IEEE Std 802.1X or PMKSA caching  | Used only with cipher suite selector values 00-0F- AC:9 (GCMP- 256), 00-0F- AC:10 (CCMP-256), 00-0F-AC:13 (BIP-CMAC- 256), and 00- 0F-AC:12 (BIP-GMAC- 256)  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |  |  |
| 00-0F-AC  | 19  | FT authentication using PSK  | FT key management as defined in 12.7.1.6 (FT key hierarchy)  | Defined in 12.7.1.6.2 (Key derivation function (KDF)) using SHA- 384  | 2 (FT) for FT protocol reassociation as defined in 13.5 (FT protocol) 0 (open) for FT Initial Mobility Domain Association using PSK  | None  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |
| 00-0F- AC(M20)  | 22  | FT authentication negotiated over IEEE Std 802.1X  | FT key management as defined in 12.7.1.6 (FT key hierarchy)  | Defined in 12.7.1.6.2 (Key derivation function (KDF)) using SHA- 384  | 2 (FT) for FT protocol reassociation as defined in 13.5 (FT protocol) 0 (open) for FT Initial Mobility Domain Association over IEEE Std 802.1X or PMKSA caching  | None  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

### Proposed changes for CID 7033

**12.13.6 PASN authentication with FT**

*Modify 12.13.6 as shown (REVme-D5.0 P3164 L39):*

This subclause specifies aspects of PASN authentication when one of AKM 00-0F-AC: [3, 4, 13, 19, 22] is used as the Base AKMP.

# CID 7027 (SEC)

Clause Number: 12.7.2 Page: 3090 Line: 61

Comment:

EAPOL-Key request frames are claimed to have the Encrypted Key Data bit set to 0. This is not correct. The Encrypted Key Data bit is set to 1 when an AEAD cipher is used and the EAPOL-Key frame is protected even if there is no plaintext Key Data contents (e.g., see EAPOL-Key message 4 or group message 2). This applies to EAPOL-Key request frames as well. It should also be noted that the standard requires encrypted Key Data fields even if they do not contain any key material (see P3094 L8) and as such, mandating the Encrypted Key Data bit to be set 0 for any case where a PTKSA is available would be questionable. For the EAPOL-Key request frame case, either 0 or 1 can be used for the Encrypted Key Data bit and as such, there is no point in trying to imply there is some constrain on this.

Proposed Change:

Replace "In an EAPOL-Key request frame, the Secure bit is set to 1, the Key MIC Present bit is set to 1 if not using an AEAD cipher and is set to 0 otherwise, and the Install and Encrypted Key Data bits are set to 0." with "In an EAPOL-Key request frame, the Secure bit is set to 1, the Key MIC Present bit is set to 1 if not using an AEAD cipher and is set to 0 otherwise, and the Install bit are set to 0."

Proposed Resolution:

REVISED - Incorporate changes under the “Proposed changes for CID 7027” section of <this doc>.

### Discussion

This was changed in REVme/D2.0 based on CID 1840:

Comment:

The setting of the Secure, Key MIC, Install, Encrypted Key Data and Error bits in an EAPOL-Key request frame is not clear

Proposed Change:

Specify that these shall be 1, 0, 0, 0 and 0 respectively. Also at 3206.12 change "Error (bit 10) is set by a Supplicant to report that a MIC failure occurred in a TKIP MSDU. In
case of a MIC failure, a Supplicant shall set this bit to 1 only when the Request (bit 11) is 1." to "Error (bit 10) is set to 1 by a Supplicant to report that a MIC failure occurred in a TKIP MSDU (in which case the Request bit (bit 11) is also set to 1); it is set to 0 otherwise."

Resolution:

REVISED (SEC: 2022-09-14 20:02:28Z) - Make the changes shown under “Proposed changes” for CID 1840 in 11-22/353r8 <https://mentor.ieee.org/802.11/dcn/22/11-22-0353-08-000m-resolutions-for-some-comments-on-11me-d1-0-lb258.docx>, which make the changes requested by the commenter, except that the Key MIC Present bit is not necessarily 0.

Document 22-353r8 shows the approved changes as follows with the relevant text highlighted:

In D1.4:

At 3225.7 after the first para under “9) Request (bit 11)”, add a para:

In an EAPOL-Key request frame, the Secure bit is set to 1, the Key MIC Present bit is set to 1 if not using an AEAD cipher and is set to 0 otherwise, and the Install and Encrypted Key Data bits are set to 0.

At 3224.63 change as follows:

8) Error (bit 10) is set to 1 by a Supplicant to report that a MIC failure occurred in a TKIP MSDU~~. In case of a MIC failure, a Supplicant shall set this bit to 1 only when~~ (in which case the Request (bit 11) is also set to 1); it is set to 0 otherwise.

At 3227.1 change as follows:

h) **Key MIC.** When the (#1823)Key MIC Present subfield (of the Key Information field) is equal to 1, the (#1830)Key MIC field is a MIC of the EAPOL-Key frame (see Figure 12-33 (EAPOL-Key frame format(#1406))), i.e., from and including the Protocol Version field of the EAPOL PDU (see Figure 12-33 (EAPOL-Key frame format(#1406))) to and including the Key Data field, calculated with the Key MIC field set to 0. If the Encrypted Key Data subfield (of the Key Information field) is equal to 1, the Key Data field is encrypted prior to computing the MIC. When using an AEAD cipher, the (#1830)Key MIC field is not present. When not using an AEAD cipher, when the Key MIC Present subfield (of the Key Information field) is equal to 0, the Key MIC field is set to 0. The length of this field depends on the negotiated AKM as defined in 12.7.3 (EAPOL-Key frame construction and processing) (see Table 12-11 (Integrity and key wrap algorithms)).

The proposed change in the comment has a minor grammar error, so for clarity, the proposed resolution is to use revised instead of accept.

### Proposed changes for CID 7027

**12.7.2 EAPOL-Key frames**

*Modify 12.7.2 as shown (REVme-D5.0 P3090 L61):*

9)

Request (bit 11) is set to 1 by a Supplicant to request that the Authenticator initiate either a 4-way handshake or group key handshake, and is set to 1 by a Supplicant in a Michael MIC Failure Report frame(#1853). The Supplicant shall not set this bit to 1 in on-going 4-way handshakes, i.e., the Key Ack bit (bit 7) shall not be set to 1 in any message in which the Request bit is 1. The Authenticator shall never set this bit to 1.

(#1840)In an EAPOL-Key request frame, the Secure bit is set to 1, the Key MIC Present bit is set to 1 if not using an AEAD cipher and is set to 0 otherwise, and the Install bit is set to 0.

In a Michael MIC Failure Report frame(#1853), setting the bit is not a request to initiate a new handshake. However, the recipient may initiate a new handshake on receiving such a message.

# CID 7029 (SEC)

Clause Number: 12.7.6.3 Page: 3105 Line: 7

Comment:

REVme/D5.0 is not exactly clear on what is supposed to be included in the PMKID List field of the RSNE in 4-way handshake message 2 during an FT initial mobility domain association. If the non-AP STA tried to use PMKSA caching for this association, it would have included one or more PMKIDs in the RSNE in (Re)Association Request frame. The rules for 4-way handshake message 2 require the PMKR1Name to "included in the PMKID List field of the RSNE" (P3104 L23). However, there is no explicit statement indicating whether the PMKIDs included in the (Re)Association Request frame are to be removed or retained (and if retained, into which position in the list the calculated PMKR1Name should be inserted). This has resulted in interoperability issues due to different interpretations (there are Supplicants that replace the PMKIDs and there are Supplicants that prepend PMKR1Name to the beginning of the PMKID List; and there are Authenticators that require the PMKID List to have only PMKR1Name and there are Authenticators that allow more than on PMKiD to be included as long as the listed values include PMKR1Name and the other values match values included in the (Re)Association Request frame). The rules for the Authenticator to validate the 4-way handshake message 3 feel problematic: "If the MIC or AEAD decryption is valid and this message 2 is part of a fast BSS transition initial mobility domain association or an association started through the FT protocol, the Authenticator checks that all fields of the RSNE other than the PMKID List field and, if present, the RSNXE bitwise matches the fields from the (Re)Association Request frame" implies that the PMKID Count field shall have the same value, but that is not always the case (e.g., when PMKSA caching is not attempted, this changes from 0 to 1). Furthermore, not checking anything about the contents of the PMKID List field does not match the expectations for security validation of values that the transmitter shall set. In other words, the Authenticator should really be explicitly required to verify that PMKR1Name is present (while allowing other values to be included).

Proposed Change:

At P3105 L7, replace "If the MIC or AEAD decryption is valid and this message 2 is part of a fast BSS transition initial mobility domain association or an association started through the FT protocol, the Authenticator checks that all fields of the RSNE other than the PMKID List field and, if present, the RSNXE bitwise matches the fields from the (Re)Association Request frame and that the FTE and MDE are the same as those provided in the AP’s (Re)Association Response frame." with "If the MIC or AEAD decryption is valid and this message 2 is part of a fast BSS transition initial mobility domain association or an association started through the FT protocol, the Authenticator checks that all fields of the RSNE other than the PMKID Count and PMKID List fields and, if present, the RSNXE bitwise matches the fields from the (Re)Association Request frame and that the FTE and MDE are the same as those provided in the AP’s (Re)Association Response frame, and the Authenticator verifies that the PMKR1Name calculated according to the procedures of 12.7.1.6.4 is included in the PMKID List field of the RSNE."

Proposed Resolution:

REVISED - Incorporate changes under the “Proposed changes for CID 7029” section of <this doc>.

### Discussion

The comment identifies two issues: (1) PMKID Count field might have a different value when PMKSA cache was used for FT initial mobility domain association, i.e., it would not “match bitwise”, and (2) possible interpretations of the current text could result in interoperability issues. The comment proposes a clear fix for (1) and a change that clarifies AP/Authenticator behavior to reduce likelihood of interoperability issues due to (2).

### Proposed changes for CID 7029

**12.7.6.3 4-way handshake message 2**

*Modify 12.7.6.2 as shown (REVme-D5.0 P3105 L7):*

If the MIC or AEAD decryption is valid and this message 2 is part of a fast BSS transition initial mobility domain association or an association started through the FT protocol, the Authenticator checks that all fields of the RSNE other than the PMKID Count and PMKID List fields and, if present, the RSNXE bitwise matches the fields from the (Re)Association Request frame and that the FTE and MDE are the same as those provided in the AP’s (Re)Association Response frame. If the MIC or AEAD decryption is valid and this message 2 is not part of a fast BSS transition initial mobility domain association and this message 2 is not part of an association started through the FT protocol, the Authenticator checks that the RSNE and, if present, the RSNXE bitwise matches that from the (Re)Association Request frame, and the Authenticator checks that the PMKR1Name calculated according to the procedures of 12.7.1.6.4 is included in the PMKID List field of the RSNE.

If all these conditions are met, the Authenticator constructs message 3. Otherwise, the Authenticator uses MLME-DEAUTHENTICATE.request primitive to terminate the association.

# CID 7005 (GEN)

Clause Number: 4.3.22.2 Page: 297 Line: 6

Comment:

The description of BSS max idle period management claims that the AP would not disassociate a STA due to nonreceipt of frames. This is misleading since a STA would be disassociated if it were not to respond to various frames in time. For example, the Authenticator state machine would disassociate such a STA if it does not reply to EAPOL-Key group key message 1 (WNM Sleep Mode is the only defined exception for avoiding that). Another example would be session timeout or EAP reauthentication timeout (no exception is available for either).

Proposed Change:

Add the following NOTE to the end of 4.3.22.2: "NOTE--The AP might disassociate the STA if that STA does not respond to group key handshake. The STA might use WNM sleep mode (see 4.3.22.33) to allow power saving to be improved for such a case.

Proposed Resolution:

REVISED - Incorporate changes under the “Proposed changes for CID 7005” section of <this doc>.

### Discussion

This CID 7005 and the following CIDs 7006, 7007, 7025 are related to use cases that need very long sleep periods for battery powered devices, e.g., sensors sending out periodic information seldomly while not necessarily needing to receive any Data frames for significant amount of time. CID 7005, 7006, 7007 propose changes to clarify use of BSS max idle period with impact from periodic rekeying of group keys. They also imply that there is limited deployment of WNM sleep mode (which would address this specific use case) and desire for a simpler mechanism. CID 7025 proposes a simplification to WNM-Sleep mode by removing the requirement to use TFS.

### Proposed changes for CID 7005

**4.3.22.2 BSS max idle period management**

*Modify 4.3.22.2 as shown (REVme-D5.0 P297 L6):*

BSS max idle period management enables an AP to indicate a time period during which the AP does not disassociate a STA due to nonreceipt of frames from the STA. This supports improved STA power saving and AP resource management.

NOTE—The AP might disassociate the STA if, for example, that STA does not respond to group key handshake message 1. The STA might use WNM sleep mode (see 4.3.22.33) to avoid getting disassociated in such a case.

# CID 7006 (MAC)

Clause Number: 11.21.13 Page: 2779 Line: 50

Comment:

The claim that "a STA can refrain from transmitting frames to its associated AP without being disassociated" based on the BSS max idle period is misleading. The NOTE at lines 61-62 tries to point this out, but that has clearly been missed by multiple people based on some discussions and expectations that have been brought up in contexts related to extreme power saving use cases (e.g., a sensor that does not need group addressed frames and is associated just to be able to transmit a frame ever now and then). WNM sleep mode has been defined for such cases, but that requires some more functionality for the STA and likely because of that, it has been less preferred option..

Proposed Change:

At P2779 L50 replace "indicates the time period during which a STA can refrain from transmitting frames to its associated AP without being disassociated" with "indicates the time period during which a STA can refrain from transmitting frames to its associated AP without being disassociated due to inactivity".

At P2779 L62, add following to the end of the NOTE: "An example of such other reason is rekeying of group keys. WNM sleep mode (see 11.2.3.15) might be used to avoid disconnection due to not replying to group key handshake."

Proposed Resolution:

REVISED - Incorporate changes under the “Proposed changes for CID 7006” section of <this doc>.

### Discussion

See CID 7005.

### Proposed changes for CID 7006

**11.21.13 BSS max idle period management**

*Modify 11.21.13 as shown (REVme-D5.0 P2779 L49-L62):*

The Max Idle Period field of the BSS Max Idle Period element indicates the time period during which a STA can refrain from transmitting frames to its associated AP without being disassociated due to inactivity. A non-AP STA is considered inactive if the AP has not received a Data frame, PS-Poll frame, or Management frame (protected or unprotected as specified in this paragraph) of a frame exchange sequence initiated by the STA for a time period greater than or equal to the time specified by the Max Idle Period field. If the Idle Options field requires protected keepalive frames, then the AP may disassociate the STA if no protected frames are received from the STA for a duration of BSS max idle period. If the Idle Options field allows unprotected or protected keepalive frames, then the AP may disassociate the STA if no protected or unprotected frames with 48-bit TA and RA fields are received from the STA for a duration of BSS max idle period.

NOTE—The AP can disassociate or deauthenticate the STA at any time for other reasons even if the STA satisfies the keep-alive frame transmission requirements. An example of such other reason is group key rekeying. WNM sleep mode (see 11.2.3.15) might be used to avoid disconnection due to not replying to group key handshake message 1.

# CID 7007 (SEC)

Clause Number: 12.7.10.1 Page: 3129 Line: 26

Comment:

The KEYERROR state of the Authenticator state machine forces a STA to be disconnected if it does not reply to group key handshake message 1 within GTimeoutCtr attempts. This is appropriate for many cases, but it might be convenient to allow an optional exception for an AP/Authenticator to skip this disconnection if the STA/Supplicant is expected to be using long sleep in cases where BSS max idle period has been advertised. While WNM sleep mode is available to avoid this, some offline discussions seemed to have preferred to not have to support it for some low power use cases. As such, the AP/Authenticator skipping the disconnection here could be of some use.

Proposed Change:

At P3129 L34, add a new NOTE immediately following Figure 12-56: 'NOTE: The Authenticator might skip the "Disconnect = true" step in the KEYERROR state if BSS max idle period is advertised and the non-AP STA is expected to be in long sleep mode during the group key handshake.'

Proposed Resolution:

ACCEPTED

 OR

REVISED - Incorporate changes under the “Proposed changes for CID 7007” section of <this doc>.

### Discussion

See CID 7005.

TODO: Discussed 2024-03-12. Needs more work. Bring back during the adhoc

Rev 2:

* Add non-AP STA guidance in 11.21.13.
* Modify normative Figure 12-56 instead of just a NOTE on potentially differing behavior.
* Note how the AP/Authenticator can send the current group keys to STAs that were skipped during rekeying.

### Proposed changes for CID 7007

**11.21.13 BSS max idle period management**

*Modify 11.21.13 as shown (REVme-D5.0 P2779 L61):*

If dot11BssMaxIdlePeriod is nonzero, an AP shall include the BSS Max Idle Period element in the (Re)Association Response frame. Otherwise, an AP shall not include the BSS Max Idle Period element in the (Re)Association Response frame. A non-S1G STA may send protected or unprotected keepalive frames, as indicated in the Idle Options field.

Extended BSS max idle period values are those that had a nonzero unified scaling factor (Table 9-78 (Unified Scaling Factor subfield encoding)) value signaled by an S1G STA. A non-AP S1G STA may include the BSS Max Idle Period element in transmitted (Re)Association Request frames to indicate a preferred BSS max idle period value. The S1G AP chooses a value for BSS max idle period based on the S1G STA’s preferred BSS max idle period (if any) and the type of the S1G STA. The S1G AP indicates its chosen value to the S1G STA in the (Re)Association Response frame.

If dot11WirelessManagementImplemented is true, dot11BSSMaxIdlePeriod is nonzero and dot11BSSMaxIdlePeriodIndicationByNonAPSTA is true, then a non-AP non-S1G STA shall include a BSS Max Idle Period element in the (Re)Association Request frame. If the BSS Max Idle Period element is present in the (Re)Association Request frame received by a non-S1G AP that has dot11BSSMaxIdlePeriodIndicationByNonAPSTA equal to true, then the non-S1G AP may choose the non- AP STA’s preferred maximum idle period. The non-S1G AP indicates its chosen value to the non-S1G STA in the (Re)Association Response frame.

The value chosen by the AP is the value that the AP will use in making disassociate decisions based on the timeout value equal to BSS max idle period for the non-AP STA that is the recipient of the (Re)Association Response frame. An AP may provide different values for BSS max idle period to different STAs.

A STA may send at least one protected or unprotected keepalive frame per BSS max idle period, as 45 indicated in the Idle Options field. When a STA transmits an unprotected keepalive frame, it shall use a frame that has 48-bit TA and RA fields.

The Max Idle Period field of the BSS Max Idle Period element indicates the time period during which a STA can refrain from transmitting frames to its associated AP without being disassociated. A non-AP STA is considered inactive if the AP has not received a Data frame, PS-Poll frame, or Management frame (protected or unprotected as specified in this paragraph) of a frame exchange sequence initiated by the STA for a time period greater than or equal to the time specified by the Max Idle Period field. If the Idle Options field requires protected keepalive frames, then the AP may disassociate the STA if no protected frames are received from the STA for a duration of BSS max idle period. If the Idle Options field allows unprotected or protected keepalive frames, then the AP may disassociate the STA if no protected or unprotected frames with 48-bit TA and RA fields are received from the STA for a duration of BSS max idle period.

NOTE 1—The AP can disassociate or deauthenticate the STA at any time for other reasons even if the STA satisfies the keep-alive frame transmission requirements.

NOTE 2—A non-AP STA might be in doze state during a group key handshake when the BSS advertises BSS max idle period. The Authenticator might allow such a STA to remain associated even if it missed a new group key. If beacon protection is enabled, such a STA might not have the current BIGTK available when waking up from doze state. This might result in discarding Beacon frames due to MME validation errors. A STA might wait for a new BIGTK to be received before disconnecting from the BSS due to beacon loss in cases where it did not wake up to receive buffered frames during a long doze state.

**12.7.10 RSNA Authenticator key management state machine**

**12.7.10.1 General**

*Modify 12.7.10.1 as shown (REVme-D5.0 P3129 L34):*



**Figure 12-56—Authenticator state machines, part 3 available and in use.**

*Modify Figure 12-56 by replacing “Disconnect = true” in the KEYERROR state with*

*“If rekeying required OR STA not in doze state*

*Disconnect = true”.*

*Add the following paragraph at REVme-D5.0 P3129 L34:*

NOTE—The Authenticator might skip the "Disconnect = true" step in the KEYERROR state if a BSS max idle period is advertised by the AP and the non-AP STA is expected to remain in doze state during the group key handshake. When the AP does not disconnect a non-AP STA in such a case, the AP might send the current group keys using group key handshake with this non-AP STA when it wakes up from the doze state.

When a second STA associates, the group key state machine is already initialized, and a GTK is already available and in use.

When the GTK is to be updated the variable GTKRekey is set to 1. The SETKEYS state updates the GTK and triggers all of the PTK group key state machines that currently exist—one per associated STA. Each PTK group key state machine sends the GTK to its STA. When all of the STAs have received the GTK (or failed to receive the key), the SETKEYSDONE state is executed which updates the APs encryption/integrity engine with the new key.

# CID 7025 (MAC)

Clause Number: 11.2.3.15.1 Page: 2515 Line: 62

Comment:

Setting dot11WNMSleepModeActivated to true mandates dot11TFSActivated to be set to true. There does not seem to be any clear justification for having to always use TFS with WNM Sleep Mode and this requirement adds unnecessary complexity for cases where the non-AP STA is not interested in receiving any frames and instead, it just wants to be able to retain an association for a long period of time while minimizing power use. For example, a battery powered sensor device might use this to send period reports. There does not seem to be much, if any, deployment of WNM Sleep Mode. However, there is interest in being able to deploy battery powered sensor devices and similar IoT devices that need to minimize both power use and complexity. Removing this requirement for TFS would make it simpler to support such devices both on the non-AP STA side and also on the AP side. This might promote deployment of WNM Sleep Mode and enable new use cases for WLAN.

Proposed Change:

At P1745 L2 (in Figure 9-1284), replace "one or more TFS Request elements" with "zero or more TFS Request elements".

At P1745 L25, replace "one or more TFS Request elements" with "zero or more TFS Request elements".

At P1745 L48 (in Figure 9-1285), replace "one or more TFS Response elements" with "zero or more TFS Response elements".

At P1747 L59, replace "one or more TFS Response elements" with "zero or more TFS Response elements".

At P2515 L62, replace "When dot11WNMSleepModeActivated is true, dot11TFSActivated shall be true" with "When dot11WNMSleepModeActivated is true, dot11TFSActivated should be true".

At P2516 L30 (in 11.2.3.15.2), replace "MLME-SLEEPMODE.request primitive shall also include a valid TFSRequest parameter" with "MLME-SLEEPMODE.request primitive may also include a valid TFSRequest parameter".

Proposed Resolution:

ACCEPTED

 OR

REVISED - Incorporate changes under the “Proposed changes for CID 7025” section of <this doc>.

### Discussion

See CID 7005.

TODO: Discussed 2024-03-12. Needs more work. Bring back during the adhoc.

Rev 2: No updates – no comments have been received from offline review after the last discussion.

The design for WNM Sleep Mode and TFS was in these documents:

<https://mentor.ieee.org/802.11/dcn/07/11-07-0737-00-000v-sleep-mode-with-ap-filtering.ppt>

<https://mentor.ieee.org/802.11/dcn/07/11-07-2148-00-000v-traffic-filtering-and-sleep-mode-presentation.ppt>

While the use cases included consideration for AP side filtering, there was no explicit justification for mandating it. In fact, the normative text for this did not mandate TFS: <https://mentor.ieee.org/802.11/dcn/07/11-07-2169-00-000v-traffic-filtering-and-sleep-mode-normative-text.doc>

<https://mentor.ieee.org/802.11/dcn/07/11-07-2169-00-000v-traffic-filtering-and-sleep-mode-normative-text.doc>

TFS was practically mandated for WNM sleep mode in P802.11v LB133 CID 280:

Comment:

TFS Request Element shalll always be included in the Sleep Mode request frame. If not, how to wake up a sleep mode station?

Proposed Change:

change "Zero or more TFS Request elements" to "one or more TFS Request
elements" throughout the draft. change "Zero or more TFS Response elements" to "one or more TFS Response elements" throughout the draft.

Resolution:

Accepted

The only provided justification for this was in the AP being able to wake up a non-AP STA in sleep mode. However, that does not seem to be needed for all use cases and as such, it seems acceptable to relax this requirement about TFS having to be always negotiated for WNM sleep mode as proposed in the comment.

While it would be possible to add a new AP capability indication (e.g., a new Extended Capabilities element bit) for WNM Sleep Mode Request frame without any TFS Request elements, there does not seem to be clear justification for this considering the limited, if any, current deployment of WNM sleep mode support.

### Proposed changes for CID 7025

**9.6.13.19 WNM Sleep Mode Request frame format**

*Modify 9.6.13.19 as shown (REVme-D5.0 P1745 L2 and L25):*

The WNM Sleep Mode Request frame is sent by a non-AP STA to the AP to enter the WNM sleep mode. The format of the WNM Sleep Mode Request frame Action field is defined in Figure 9-1284 (WNM Sleep Mode Request frame Action field format).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | zero or more TFS Request elements |  |
|  | Category | WNM Action | Dialog Token | WNM Sleep Mode Element | TFS Request Elements | OCI Element (optional) |
| Octets: | 1 | 1 | 1 | 6 | variable | 0 or 6 |

**Figure 9-1284—WNM Sleep Mode Request frame Action field format**

The Category field is defined in 9.4.1.11 (Action field).

The WNM Action field is defined in 9.6.13.1 (WNM Action field).

The Dialog Token field is defined in 9.4.1.12 (Dialog Token field). It is a nonzero value chosen by the non-AP STA sending the WNM Sleep Mode Request frame to identify the request/response transaction.

The WNM Sleep Mode Element field contains a WNM Sleep Mode element that is requested by a non-AP STA, as described in 9.4.2.80 (WNM Sleep Mode element).

The TFS Request Elements field contains zero or more TFS Request elements to specify the traffic filters that are requested by a non-AP STA, as defined in 9.4.2.78 (TFS Request element).

The OCI Element field is optionally present, and contains an OCI element as defined in 9.4.2.235 (OCI element).

**9.6.13.20 WNM Sleep Mode Response frame format**

*Modify 9.6.13.20 as shown (REVme-D5.0 P1745 L48):*

The WNM Sleep Mode Response frame is sent by an AP in response to a WNM Sleep Mode Request frame or is sent without solicitation by an AP to a non-AP STA upon the AP’s deletion of all traffic filter sets established according to the traffic filtering agreement between the AP and the non-AP STA. The format of the WNM Sleep Mode Response frame Action field is defined in Figure 9-1285 (WNM Sleep Mode Response frame Action field format).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Category | WNM Action | Dialog Token | Key Data Length | Key Data |
| Octets: | 1 | 1 | 1 | 2 | variable |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | zero or more TFS Response elements |  |
|  | WNM Sleep Mode Element | TFS Response Elements | OCI Element (optional) |
| Octets: | variable | variable | 0 or 6 |

**Figure 9-1285—WNM Sleep Mode Response frame Action field format**

The Category field is defined in 9.4.1.11 (Action field).

The WNM Action field is defined in 9.6.13.1 (WNM Action field).

*Modify 9.6.13.20 as shown (REVme-D5.0 P1747 L59):*

The WNM Sleep Mode Element field contains a WNM Sleep Mode element, as described in 9.4.2.80 (WNM Sleep Mode element).

The TFS Response Elements field contains zero or more TFS Response elements to specify the traffic filters, as defined in 9.4.2.79 (TFS Response element).

The OCI Element field is optionally present, and contains an OCI element as defined in 9.4.2.235 (OCI element).

**11.2.3.15 WNM sleep mode**
**11.2.3.15.1 WNM sleep mode capability**

*Modify 11.2.3.15.1 as shown (REVme-D5.0 P2515 L62):*

Implementation of the WNM sleep mode capability is optional for a WNM STA. A STA that implements WNM sleep mode has dot11WNMSleepModeImplemented equal to true. When dot11WNMSleepModeImplemented is true, dot11WirelessManagementImplemented shall be true. A STA where dot11WNMSleepModeActivated is true is defined as a STA that supports WNM sleep mode. A STA supporting WNM sleep mode shall set the WNM Sleep Mode field of the Extended Capabilities element to 1. When dot11WNMSleepModeActivated is true, dot11TFSActivated should be true.

**11.2.3.15.2 WNM sleep mode non-AP STA operation**

*Modify 11.2.3.15.2 as shown (REVme-D5.0 P2516 L30):*

To use the WNM sleep mode service, the non-AP STA’s SME shall issue an MLME-SLEEPMODE.request primitive to send a WNM Sleep Mode Request frame. The MLME-SLEEPMODE.request primitive shal include a valid SleepMode parameter with a WNM Sleep Mode element. The Action Type field in the WNM Sleep Mode element shall be set to “Enter WNM sleep mode” and the WNM Sleep Interval field shall be included. The WNM Sleep Interval field shall be less than the BSS max idle period (see 11.21.13 (BSS max idle period management)). The MLME-SLEEPMODE.request primitive may also include a valid TFSRequest parameter as defined in the TFS Request element that the AP shall use as triggers to set the STA’s TIM bit.

# CID 7028 (SEC)

Clause Number: 12.5.4.3.4 Page: 3040 Line: 17

Comment:

The security of QMF depends on the ACI of the MMPDU being protected. This is similar to how the security of QoS for Data frames depends on the TID being protected. ACI is protected when using CCMP, but not when using GCMP. ACI is encoded in the ACI subfield in the Sequence Number field. That field is masked out from AAD construction for both CCMP and GCMP and as such, AAD does not protect it (unlikely it does for QoS Data frames with QC being included). CCMP protects the ACI subfield value by defining the priority value of the MPDU to be equal to its value (P3021 L28). This priority value is then included in the CCM nonce (see Figure 12-21) and that provides protection to the ACI value. GCMP on the other hand does not include the priority value in the GCM nonce (see Figure 12-30). Consequently, there is no protection for the ACI value and attacker can modify it without the frame recipient being able to detect the modification based on GCMP processing. This enabled attacks that could be used to reorder Robust Management frames between different access categories. Protection of ACI with GCMP is inconvenient since there is no room in the GCM nonce for the priority value. The standard could be extended to construct the AAD for GCMP to include a new field for the QMF cases (e.g., a "virtual" octet with the ACI encoded in it at the end of the AAD) or by not masking the ACI subfield of the Sequence Number field in QMFs. This would make the AAD construction different for GCMP compared to CCMP (since we should not change CCMP definition for this and break compatibility with the original design). It might be acceptable to modify GCMP for QMF due to limited, if any, interest in deploying QMF so far. It would also be possible to negotiate use of the extended AAD for GCMP when QMF is used. That said, if there is no interest in deploying QMF, there may not be much benefit from coming up with more complex solutions for this than simply disallow use of QMF with GCMP. This comment proposes an unconditional change to the AAD construction for GCMP for QMFs to unmask the ACI field. This is not compatible with previous definition. However, this is believed to be acceptable due to no known deployment of QMF with GCMP. This comment could be satisfied with a similar change done based on negotiated capability (e.g., and RSNXE bit) or by disallowing use of QMF with GCMP.

Proposed Change:

At P3023 L45-46, replace "SC – MPDU Sequence Control field, with the Sequence Number subfield (bits 4–15 of the Sequence Control field) masked out" with "SC – MPDU Sequence Control field, with the QMF Sequence Number field (bits 4-13 of the Sequence Control field) masked out in QMFs and with the Sequence Number subfield (bits 4–15 of the Sequence Control field) masked out in frames that are not QMFs" .

Proposed Resolution:

### Discussion

This comment proposes one direction. See 342r1 for another and a proposed resolution for the comment.

# CID 7000

Clause Number: Page: Line:

Comment:

Proposed Change:

Proposed Resolution:

### Discussion

Abcd

### Proposed changes for CID 7000

**12.7.6.2 4-way handshake message 1**

*Modify 12.7.6.2 as shown (REVme-D5.0 P3036 L9):*

Abcd