IEEE P802.11
Wireless LANs

|  |
| --- |
| Resolutions for some comments on 11me/D0.0 (CC35) |
| Date: 2021-09-15 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Mark RISON | Samsung Cambridge Solution Centre | SJH, CB4 0DS, U.K. | +44 1223 434600 | at samsung (a global commercial entity) I'm the letter emme then dot rison |

Abstract

This submission proposes resolutions for various CIDs on 11me/D0.0. Green indicates material agreed to in the group, yellow material to be discussed, red material rejected by the group and cyan material not to be overlooked. The “Final” view should be selected in Word.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 360Mark RISON | For AAD construction, what is the difference between "not modified" and "not masked"? How does "masked to <n>" differ from "set to <n>" anyway? | In 12.5.3.3.3, 12.5.4.3, 12.5.3.3.1, 12.5.3.3.6, 12.5.4.5, 12.5.4.6, 12.5.5.3.1 change "masked to" to "set to" and "unmasked" to "not modified" (case-preservingly) |

Discussion:

The intent of using the expression “masked to” as opposed to “set to” is to avoid any suggestion that the field is actually modified: it is just the copy of the field used in the AAD construction etc. that is. However, this is not specified, and the term is not used consistently. In addition there is one instance of the confusing term “unmasked”. Furthermore, “masked to 0” suggests one could mask to something else, but in programming terms masking is generally understood as referring to using a bitwise AND function to clear bits, i.e. to set them to 0.

Proposed resolution:

REVISED

At the end of Subclause 12.1 add:

Masking refers to forcing certain bits in copies of fields in another structure to 0. The original field itself is not modified.

At 2573.31, 2574.34 change “always set to 1” to “not modified (left as 1)”.

At 2573.35 change “Unmasked otherwise” to “Not modified otherwise”.

At 2571.21, 2585.8 change “may change when” to “might change when” (matching 2571.54).

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 587Thomas DERHAM12.6.32622 | Unlike "AP MFPR", the value of "STA MFPR" (in RSNE sent by non-AP STA) does not affect the peer (AP) behavior if STA MFPC is 1. Clarify since it is a frequent source of confusion. | See comment |

Discussion:

The description in Table 12-5—Robust management frame selection in an infrastructure BSS et al. suffers from the following:

* “No action” is unclear
* 3 combinations are missing
* The behaviour of STAs that are not conformant with any revision of the standard is described
* There is vague wording about “shall not try to”

Taking each of these points in turn:

* There is an issue with STAs that conform to a revision of the standard before 802.11w-2009, since for those STAs the MFPC and MFPR bits are reserved. This means that they can be expected to set them to 0 on tx, but also to ignore them on rx, and so attempt association even if the AP has MFPR set
* All combinations should be shown, but the valid and invalid combinations should be clearly distinguished
* Unless a strong argument is made to the contrary, we should only define conformant behaviour. We do not need to get involved in discussions about nasal demons
* Do. Or do not. There is no “try”

Proposed changes:

Change Table 12-5 to the following (note to the Editor: columns and rows have been moved, modified and added, and a NOTE added):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Non-AP STA MFPC** | **Non-AP STA MFPR** | **Non-AP STA action** | **AP MFPC** | **AP MFPR** | **AP action** | **MFP used?** |
| 0 | 0 | The STA may associate with the AP | 0 | 0 | The AP may accept associations from the STA | No |
| 1 | 0 | The STA may associate with the AP | 0 | 0 | The AP may accept associations from the STA | No |
| 0 | 0 | The STA may associate with the AP | 1 | 0 | The AP may accept associations from the STA | No |
| 1 | 0 or 1 | The STA may associate with the AP | 1 | 0 or 1 | The AP may accept associations from the STA | Yes |
| 1 | 1 | The STA shall not associate with the AP | 0 | 0 | N/A | N/A |
| 0 | 0 | The STA should not associate with the AP (see NOTE) | 1 | 1 | The AP shall reject associations from the STA with the Status Code ROBUST\_MANAGEMENT\_POLICY\_VIOLATION | N/A |
| 0 | 1 | The STA shall not use this combination |  |  |  | N/A |
|  |  |  | 0 | 1 | The AP shall not use this combination | N/A |
| NOTE—STAs conformant with a previous revision of this standard might not ascribe a meaning to the MFPC and MFPR subfields. |

Change 9.4.2.24.4 RSN capabilities as follows:

— Bit 6: MFPR. A STA sets this bit to 1 ~~to advertise that protection of robust Management frames is mandatory. A STA sets this bit to 1~~ when dot11RSNAProtectedManagementFramesActivated is true and dot11RSNAUnprotectedManagementFramesAllowed is false to advertise that protection of robust Management frames is mandatory; otherwise it sets this bit to 0. ~~If a STA sets this bit to 1, then that STA only allows RSNAs with STAs that provide Management Frame Protection.~~

— Bit 7: MFPC. A STA sets this bit to 1 when dot11RSNAProtectedManagementFramesActivated is true to advertise that protection of robust Management frames is enabled; otherwise it sets this bit to 0.

Change 12.6.3 RSNA policy selection in an infrastructure BSS as follows:

An AP and a non-AP STA shall use Table 12-5 (Robust management frame selection in an infrastructure BSS) and the values of the MFPC and MFPR bits advertised in the RSNEs to determine if ~~it~~they may associate ~~with a non-AP STA~~, and if so whether management frame protection is enabled. ~~An non-AP STA shall use Table 12-5 (Robust management frame selection in an infrastructure BSS) and the values of the MFPC and MFPR bits advertised in the RSNEs to determine if it may associate with an AP. Management frame protection is enabled when dot11RSNAProtectedManagementFramesActivated is set to 1. Management frame protection is negotiated when an AP and non-AP STA set the Management Frame Protection Capable field to 1 in their respective RSNEs in the (re)association procedure, and both parties confirm the Management Frame Protection Capable bit set to 1 in the 4-way handshake, FT 4-way handshake, FT fast BSS transition protocol, or the (Re)Association Request and (Re)Association Response frames of FILS authentication.~~

In 12.6.19 Protection of robust Management frames change “a STA that advertised MFPC = 1” to “any STA that advertised MFPC = 1” (for consistency with the other occurrences).

In 12.6.20 Robust management frame selection procedure delete the first paragraph, “A STA with dot11RSNAProtectedManagementFramesActivated equal to true shall negotiate management frame protection with a STA that advertised MFPC = 1.”.

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 587 in <this document>, which clarify the negotiation of MFP in an infrastructure BSS, including interoperation with pre-11w devices.

|  |  |  |
| --- | --- | --- |
| CID 213Mark RISON12.5 | The Length field in Figure 12-36--GTK KDE format is not defined. I suspect it's the Length in Figure 12-35--KDE format but then the formula doesn't work, because the OUI and Data Type fields already consume 4 octets from the Length | Change - 4 to -6 in F12-36, change the hyphen to a minus, and where F12-36 is referred to add " (where Length is the Length field in the KDE)"; ditto where F12-42 is referred to and where Figure 12-47 is referred to (and the hyphen->minus there) |

Discussion:

The comment is correct, but it’s simpler to just say the field is of variable size (though this reveals that the spec is inconsistent as to the capitalisation of “variable”).

Proposed resolution:

REVISED

In Figures 12-36, 12-42 and 12-47 change the size shown below the rightmost field to “variable”.

In Figures 9-76, 9-216, 9-218, 9-219, 9-277, 9-278 (2x), 9-280 (2x), 9-568, 9-570, 9-589, 9-600, 9-963, 23-44, 23-45, 23-46, 25-35, 25-36 change “Variable” to “variable”.

In 9.4.2.50 RIC Descriptor element change “Variable parameters” to “Variable Parameters” in the figure and in the body change “Variable parameters contain” to “The Variable Parameters field contains”.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 223Mark RISON12.7.8.22653.29 | "b) The AP's RSNE indicates that WEP-40 (OUI  00-0F-AC:1) or WEP-104 (OUI 00-0F-AC:5) are enabled as either pairwise or group cipher suites; [...]Violation of any of these cases would cause the TPK handshake to leak the TPK." -- if the issue is that WEP is no longer secure, then the same is true for TKIP (and indeed TKIP is explicitly banned in 12.7.8.4.2) | Add TKIP (00-0F-AC:2) to the list in b) |

Discussion:

Jouni MALINEN comments:

That sounds fine to me since this is within a "STA may refuse to initiate" location, i.e., it would not make any existing implementation noncompliant with the new standard. Furthermore, no one should really be using RSN with TKIP as the pairwise cipher (i.e., TKIP was used with the original WPA but never as the pairwise cipher with WPA2==RSN).

Also, 00-0F-AC:1 is not an OUI.

Proposed resolution:

REVISED

Change "b) The AP's RSNE indicates that WEP-40 (OUI 00-0F-AC:1) or WEP-104 (OUI 00-0F-AC:5) are enabled as either pairwise or group cipher suites;"

to "b) The AP's RSNE indicates that WEP-40 (00-0F-AC:1), WEP-104 (00-0F-AC:5) or TKIP (00-0F-AC:2) are enabled as either pairwise or group cipher suites;".

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 395Mark RISON12 | Is the MIC really always 16 octets for GCMP? For CCMP it's "variable" (compare the "Encapsulated GCMP/CCMP MPDU" figures) | As it says in the comment |

Discussion:

Dan HARKINS comments:

GCM can have different tag lengths and there is a discussion of the matter in appendix B of the GCM reference we use in our standard but GCMP, our instantiation of GCM, doesn't have a variable length tag.

Proposed resolution:

REJECTED

Yes it is. GCM can have different tag lengths, and there is a discussion of the matter in appendix B of the GCM reference we use in our standard, but GCMP, our instantiation of GCM, doesn't have a variable length tag.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 356Mark RISON12.5.5.12583.54 | If "GCM also requires a unique nonce value for each frame protected by a given temporal key.", then then how does the GCM nonce achieve this? It just has the A2 and the PN, and doesn't have e.g. the TID+Management bits that ensure uniqueness in the CCM nonce | As it says in the comment |

Discussion:

The PN is unique in GCMP:

**12.5.5.3.1 General**

Increment the PN, to obtain a fresh PN for each MPDU, so that the PN never repeats for the same temporal key.

**12.5.5.3.2 PN processing**

The PN is incremented by a positive number for each MPDU.

**12.5.5.3.6 GCM originator processing**

The PN values sequentially number each MPDU. Each transmitter shall maintain a single PN (48-bit counter) for each PTKSA and GTKSA. The PN shall be implemented as a 48-bit strictly increasing integer

Regarding the priority and frame type fields in the CCM nonce, Jouni MALINEN comments:

I'm not sure there was really any strong reason for adding Priority in the CCM nonce in 802.11i since QC is covered by AAD and the RSN design ended up requiring a single TX PN counter. After 802.11ae got added, it is needed for protecting ACI for QMF. Similarly, the Management bit in the CCM nonce from 802.11w does not really look critical to add for the same reason (it is covered in AAD and there is only a single TX PN counter). Both of these cases might have been based on design discussions speculating on use of separate PN counters for transmitting different frames. On the other hand, 802.11ah addition of PV1 actually has a real need for either the Priority field or the new PV1 field in the nonce to keep the CCM nonce unique since PV1 does use separate TX PN counters. It might have been able to work with the existing Priority field without adding PV1 since protocol version is covered in AAD, but the PV1 case of AAD and nonce design is quite different, so it feels more reasonable to explicitly include this in the nonce just in case even if not absolutely required for ensuring unique nonce values for all frames with the same key.

Proposed resolution:

REJECTED

The PN is unique, so as the GCM nonce contains the PN, the GCM nonce is unique. The reason the CCM nonce needs a priority field is to protect the ACI under QMF and to support multiple PNs in S1G STAs (which do not support GCMP). The PV1 field, however, is not strictly necessary, since the AAD covers the protocol version.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 190Mark RISON12.7.6.52645.31 | If the Key Data is empty then the length is 0 and there's no meaning to saying it's encrypted | Either add a NOTE to explain what might be included even though Key Data is "none required" (and that the Encrypted Key Data = 1 is ignored in that case) or change line 17 to say Encrypted Key Data = 0, line 30 to say Key Data Length = 0 and line 31 to say Key Data = none |
| ~~CID 589~~~~Thomas DERHAM~~~~12.7.1.3~~~~2641~~ | ~~Key Data field in M1 and M2 "need not be encrypted" - be more precise about conditions under which it can be, or must be, encrypted~~ | ~~See comment~~ |

Discussion:

It is not clear what it means for a zero-length field to be encrypted, especially since the Key Data field has padding rules that would cause padding to be added for a zero-length unencrypted field.

In M4 the Key Data field can contain vendor-specific KDEs/elements.

Subclauses 12.7.6.2 4-way handshake message 1 and 12.7.6.3 4-way handshake message 2 define when the Key Data field in M1 and M2 is encrypted (specifically: never and only when an AEAD cipher is used respectively). More generally, these and subsequent subclauses define the content and encryption of the Key Data field, and the duplication is just causing spec rot.

Proposed changes:

Change 12.7.2 EAPOL-Key frames as follows:

i) **Key Data Length.** This field is an unsigned integer. This represents the length of the Key Data field in octets. If the Encrypted Key Data subfield (of the Key Information field) is 1, the length is the length of the Key Data field after encryption, including any padding. The Key Data Length field may be 0 when the Encrypted Key Data subfield is 1.

If the Encrypted Key Data subfield (of the Key Information field) is 1, the entire Key Data field shall be encrypted. If the Key Data field uses the NIST AES key wrap, then the Key Data field shall be padded before encrypting if the ~~key data~~ length of the key data is nonzero and less than 16 octets, or if it is not a multiple of 8 octets.

— 4-way handshake message 1 is an EAPOL-Key frame with the Key Type subfield equal to 1. Use of the Key Data field to indicate a PMKID when a cached PMKSA is being used in this key derivation is defined in 12.6.10.3 (Cached PMKSAs and RSNA key management). When a cached PMKSA is not being used, inclusion of the PMKID (if derived) is optional. ~~The Key Data field need not be encrypted.~~

— 4-way handshake message 2 is an EAPOL-Key frame with the Key Type subfield equal to 1. ~~The Key Data field shall contain an RSNE, may contain an RSNXE, and need not be encrypted.~~

[…]

— 4-way handshake message 3 is an EAPOL-Key frame with the Key Type subfield equal to 1. ~~The Key Data field shall contain one or two RSNEs, and may contain an RSNXE. If a group cipher has been negotiated, this field shall also include a GTK. This field shall be encrypted if a GTK is included.~~

[…]

~~A GTK shall be included and the unencrypted length of the GTK is six less than the length of the GTK KDE in octets. The entire Key Data field shall be encrypted as specified by the Key Descriptor Version.~~

— 4-way handshake message 4 is an EAPOL-Key frame with the Key Type subfield equal to 1. ~~The Key Data field can be empty.~~

— Group key handshake message 1 is an EAPOL-Key frame with the Key Type subfield equal to 0. ~~The Key Data field shall contain a GTK KDE and shall be encrypted.~~

— Group key handshake message 2 is an EAPOL-Key frame with the Key Type subfield equal to 0. ~~The Key Data field can be empty.~~

Change 12.7.6.5 4-way handshake message 4 as follows:

Encrypted Key Data = 1 when using an AEAD cipher or 0 otherwise

Key Data = none required, but may include one or more vendor specific KDEs and/or Vendor Specific elements

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 190 in <this document>, which clarify that the Key Data Length field can be 0 when the Encrypted Key Data subfield is 1, and that M4 doesn’t require any key data but vendor-specific stuff might be included.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 191Mark RISON12.5.5.3.62586.40 | GCMP's version of CCMP's "A transmitter shall not use an IEEE 802.11 MSDU or A-MSDU priority if this would cause the total number of priorities used during the lifetime of the SA to exceed the number of replay counters supported by the receiver (for a pairwise SA) or all the receivers (for a group SA) for that SA. 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 802.11 MSDU or A-MSDU priority." isn't as clear | Replace the para at the referenced location with the cited text, changing CCMP to GCMP |

Discussion:

The proposal is to change, at the referenced location:

A transmitter shall not use IEEE 802.11 MSDU or A-MSDU priorities without ensuring that the receiver supports the required number of replay counters. The transmitter shall not reorder GCMP 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 802.11 MSDU or A-MSDU priority.

to:

A transmitter shall not use an IEEE 802.11 MSDU or A-MSDU priority if this would cause the total number of priorities used during the lifetime of the SA to exceed the number of replay counters supported by the receiver (for a pairwise SA) or all the receivers (for a group SA) for that SA. The transmitter shall not reorder GCMP 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 802.11 MSDU or A-MSDU priority.

Proposed resolution:

REVISED

Make the changes shown under CID 191 in <this document> under “Discussion”. Note to the commenter: these are the proposed changes.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 394Mark RISON12 | "The Wrapped Key field contains the encrypted GTK as described in 13.8.5 (FT authentication sequence: contents of fourth message)." v "The Wrapped Key field contains the wrapped IGTK being distributed."/"The Wrapped Key field contains the wrapped BIGTK being distributed." inconsistency | Make consistent |

Discussion:

Actually, D0.0 has “The Wrapped Key field contains the wrapped GTK as described in 13.8.5 (FT authentication sequence: contents of fourth message).”, not “… encrypted GTK …”.

Proposed changes:

Change “The Wrapped Key field contains the wrapped GTK as described in 13.8.5 (FT authentication sequence: contents of fourth message).” to “The Wrapped Key field contains the wrapped GTK being distributed.” in 9.4.2.47 Fast BSS Transition element (FTE).

Proposed resolution:

REVISED

At 1151.58, change “The Wrapped Key field contains the wrapped GTK as described in 13.8.5 (FT authentication sequence: contents of fourth message).” to “The Wrapped Key field contains the wrapped GTK being distributed.” in 9.4.2.47 Fast BSS Transition element (FTE).

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 199Mark RISON12.6.3 | There is information on how MFP is negotiated for infrastructure BSS (Table 12-5--Robust management frame selection in an infrastructure BSS) and for IBSS (Table 12-6--Robust management frame selection in an IBSS) but not for TDLS. More generally, the use of MFP on a TDLS direct link is lacking (there's just "After receiving a Deauthentication frame or a Disassociation frame from the AP, a Deauthentication frame with Reason Code LEAVING\_NETWORK\_DEAUTH shall be transmitted via the direct path to all TDLS peer STAs that are in the awake state, if management frame protection has not been negotiated on the TDLS direct link." buried in 11.20.5 TDLS direct-link teardown) | Change "Table 12-6--Robust management frame selection in an IBSS" to "Table 12-6--Robust management frame selection in an IBSS or between TDLS peer STAs". In that table change "The peer STA shall not" to "The STA shall not". At 2598.50 change "An STA" to "A STA" and after that sentence add "A TDLS STA shall use Table 12-6 and thevalues of the MFPC and MFPR bits advertised in the RSNEs to determine if it may establish a TDLS link with another a TDLS peer STA." |
| CID 200Mark RISON12.6.19 | This subclause talks of "associated STA" but MFP can be used with IBSS and TDLS too | Change "associated STA" to "associated or peer STA" throughout this subclause |
| CID 202Mark RISON12.6.15 | "If enabled, management frame protection shall only be used as a required feature (MFPR) in an IBSS." -- what does this mean? It might be trying to say that in an IBSS if you're going to do MFP you have to set MFPR, but that's contradicted by Table 12-6--Robust management frame selection in an IBSS. Even with the "only" (a word that always massively increases the risk of ambiguity) it's not clear what it might be trying to say | Delete the cited sentence |

Discussion:

We need more clarity on the use of MFP between TDLS peer STAs. It is known that there are implementations that (a) set the MFPC and MFPR bits to 0 in the TDLS 3WH even though MFP is in use on the link to the AP and (b) ignore the MFPC and MFPR bits from the TDLS peer STA in the TDLS 3WH (or TDLS discovery procedure, if performed). Note that e.g. ADDBA Request frames are robust Management frames, so it is important to be clear on use of MFP on a TDLS link.

The behaviour is similar to IBSS, in that devices with MFPC and MFPR bits both set to 0 might just ignore them completely, so this needs to be allowed for. On the other hand, people should be encouraged to enable MFP, based on what they advertised to the AP (and what they support):

|  |  |  |  |
| --- | --- | --- | --- |
| MFPC with AP | MFPR with AP | MFPC with TDLS | MFPR with TDLS |
| Was 0 | Was 0 | Should be 1 if supported | Should be 0 |
| Was 1 | Was 0 | Should be 1 | Should be 0 |
| Was 1 | Was 1 | Should be 1 | Should be 1 |

There is no GTK on a TDLS link, however.

As regards IBSS STAs, it seems reasonable to say that since all STAs in the IBSS have to use the same security parameters (so that you don’t have to probe all peer IBSS STAs to find out what they support), MFPR has to be 1 if MFPC is 1.

As regards PBSS STAs, it looks as if the xrefs in 12.6.8 RSNA policy selection in a PBSS have rotted (off-by-one in the second number). In a PBSS a non-PCP STA can choose whether to associate with the PCP; if it does it uses infrastructure BSS-like mechanisms with the PCP; otherwise it uses IBSS-like mechanisms (for all peers inc. the PCP).

Proposed changes:

Change 12.6.3 RSNA policy selection in an infrastructure BSS as follows:

RSNA policy selection in an infrastructure BSS utilizes the normal IEEE 802.11 association procedure. RSNA policy selection is performed by the associating STA. The STA does this by including an RSNE in its ~~(Re)Association Requests~~(re)association requests.

An AP and a non-AP STA shall use Table 12-5 (Robust management frame selection in an infrastructure BSS) and the values of the MFPC and MFPR bits advertised in the RSNEs to determine if they may associate, and if so whether management frame protection is enabled. If either STA does not advertise an RSNE or does not advertise an RSN Capabilities field in an RSNE, this shall be treated as if its MFPC and MFPR bits were 0. A STA in an infrastructure BSS shall, outside the context of TDLS, set the MFPC bit to 1 if dot11RSNAProtectedManagementFramesActivated is true and to 0 otherwise, and set the MFPR bit to 1 if dot11RSNAUnprotectedManagementFramesAllowed is false and to 0 otherwise. dot11RSNAProtectedManagementFramesActivated shall be true if dot11RSNAUnprotectedManagementFramesAllowed is false.

Change 12.6.4 TSN policy selection in an infrastructure BSS as follows:

In a TSN, an RSNA STA shall include the RSNE in its ~~(Re)Association Requests~~(re)association requests.

Change 12.6.8 RSNA policy selection in a PBSS as follows:

RSNA policy selection in a PBSS utilizes the association procedure (11.3.1 (State variables)) if ~~the initiating~~a non-PCP STA chooses to associate with ~~a~~the PCP. RSNA policy selection is performed by the associating STA. The STA does this by including an RSNE in its ~~(Re)Association Requests~~(re)association requests.

If a non-PCP STA chooses to associate with the PCP, t~~T~~he STA follows the procedures in ~~12.5.3 (CTR with CBC-MAC protocol (CCMP))~~ 12.6.3 (RSNA policy selection in an infrastructure BSS) to select RSNA policy with the PCP, with the PCP taking the role of the AP. ~~If the initiating STA chooses not to associate with a peer in a PBSS~~Otherwise, it follows the procedures in ~~12.5.5 (GCM protocol (GCMP))~~ 12.6.5 (RSNA policy selection in an IBSS) to select RSNA policy with the peer (including the PCP, if it chose not to associate).

In 12.6.3 RSNA policy selection in an infrastructure BSS add after the last para:

TDLS STAs shall use Table 12-5t (Robust management frame selection between TDLS STAs) and the values of the MFPC and MFPR bits advertised in the RSNE received from the TDLS peer STA during TDLS discovery or in the RSNEs exchanged in the 3-way handshake with the TDLS peer STA to determine if a TDLS direct link is allowed, and if so whether management frame protection is enabled. If either STA does not advertise an RSN Capabilities field in an RSNE, this shall be treated as if its MFPC and MFPR bits were 0. A TDLS STA should, in the context of TDLS, set the MFPC bit to 1 if dot11RSNAProtectedManagementFramesActivated is true, and shall set it to 0 unless dot11RSNAProtectedManagementFramesActivated is true. A TDLS STA should, in the context of TDLS, set the MFPR bit to the same value as it advertised to the AP with which it is associated.

**Table 12-5t—Robust management frame selection between TDLS STAs**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TDLS initiator STA MFPC** | **TDLS initiator STA MFPR** | **TDLS initiator STA action** | **TDLS responder STA MFPC** | **TDLS responder STA MFPR** | **TDLS responder STA action** | **MFP used?** |
| 0 | 0 | The TDLS initiator STA may establish a TDLS direct link with the TDLS responder STA | 0 or 1 | 0 | The TDLS responder STA may establish a TDLS direct link with the TDLS initiator STA | No |
| 1 | 0 | 0 | 0 |
| 1 | 0 or 1 | The TDLS initiator STA may establish a TDLS direct link with the TDLS responder STA | 1 | 0 or 1 | The TDLS responder STA may establish a TDLS direct link with the TDLS initiator STA | Yes |
| 1 | 1 | The TDLS initiator STA shall not establish a TDLS direct link with the TDLS responder STA (see NOTE 1) | 0 | 0 | See NOTE 1 | N/A |
| 0 | 0 | The TDLS initiator STA should not attempt to establish a TDLS direct link with the TDLS responder STA (see NOTE 2) | 1 | 1 | The TDLS responder STA shall reject attempts by the TDLS initiator STA to establish a TDLS direct link with the Status CodeROBUST\_MANAGEMENT\_POLICY\_VIOLATION (see NOTE 2) | N/A |
| 0 | 1 | The TDLS initiator STA shall not use this combination |  |  |  | N/A |
|  |  |  | 0 | 1 | The TDLS responder STA shall not use this combination | N/A |
| NOTE 1—The TDLS initiator STA might not know the TDLS responder STA’s MFPC and MFPR bits until it receives the TDLS Setup Response frame during the 3-way handshake, if it has not performed TDLS discovery. In this case, if the TDLS responder STA that is not MFPC did not examine the TDLS initiator STA’s MFPC and MFPR bits and so did not use the Status Code ROBUST\_MANAGEMENT\_POLICY\_VIOLATION in the TDLS Setup Response frame, the TDLS initiator STA can cause the TDLS direct link establishment to fail by using the Status Code ROBUST\_MANAGEMENT\_POLICY\_VIOLATION in the TDLS Setup Confirm frame.NOTE 2—A TDLS initiator STA that is not MFPC might not examine the TDLS responder STA’s MFPC and MFPR bits. In this case the TDLS responder STA causes the TDLS direct link establishment to fail by using the Status Code ROBUST\_MANAGEMENT\_POLICY\_VIOLATION in the TDLS Setup Response frame. |

Change 12.6.5 RSNA policy selection in an IBSS as follows:

~~To establish a connection with a peer STA, an RSNA enabled~~ IBSS STAs ~~that implements management frame protection~~ shall use Table 12-6 (Robust management frame selection in an IBSS) and the values of the MFPC and MFPR ~~values~~bits advertised in the RSNEs exchanged in the 4-way handshake ~~initiated by the Authenticator of the STA with the larger MAC address~~ to determine if the communication is allowed~~. M~~, and if so whether management frame protection is enabled ~~when dot11RSNAProtectedManagementFramesActivated is set to 1. The STAs negotiate protection of Management frames when the both STAs set the Management Frame Protection Capable subfield to 1 during the 4-way handshake~~. If a 4-way handshake is not performed and a STA has not received an RSNE from a peer STA (e.g. it has not received a Beacon or Probe Response frame from the peer STA), it shall treat this as if the peer’s MFPC and MFPR bits were 0. An IBSS STA shall set the MFPC bit to 1 if dot11RSNAProtectedManagementFramesActivated is true and to 0 otherwise, and set the MFPR bit to 1 if dot11RSNAUnprotectedManagementFramesAllowed is false and to 0 otherwise. dot11RSNAProtectedManagementFramesActivated shall be true if dot11RSNAUnprotectedManagementFramesAllowed is false and shall not be true otherwise.

In 12.6.15 delete “If enabled, management frame protection shall only be used as a required feature (MFPR) in an IBSS.”

Change Table 12-6 to the following (note to the Editor: columns and rows have been moved, modified and added, and a NOTE added):

**Table 12-6—Robust management frame selection in an IBSS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STA MFPC** | **STA MFPR** | **STA action** | **Peer STA MFPC** | **Peer STA MFPR** | **Peer STA action** | **MFP used?** |
| 0 | 0 | The STA may exchange data with the peer STA | 0 | 0 | The peer STA may exchange data with the STA | No |
| 1 | 1 | The STA may exchange data with the peer STA | 1 | 1 | The peer STA may exchange data with the STA | Yes |
| 1 | 1 | The STA shall not exchange data nor establish a security association with the peer STA | 0 | 0 | N/A | N/A |
| 0 | 0 | The STA should not exchange data nor attempt to establish a security association with the peer STA (see NOTE) | 1 | 1 | The peer STA shall not exchange data with the STA and shall reject security association attempts from the STA with the Status Code ROBUST\_MANAGEMENT\_POLICY\_VIOLATION | N/A |
| 1 | 0 | The STA shall not use this combination |  |  |  | N/A |
| 0 | 1 | The STA shall not use this combination |  |  |  | N/A |
|  |  |  | 1 | 0 | The peer STA shall not use this combination | N/A |
|  |  |  | 0 | 1 | The peer STA shall not use this combination | N/A |
| NOTE—STAs conformant with a previous revision of this standard might not ascribe a meaning to the MFPC and MFPR subfields. |

Change 12.6.19 Protection of robust Management frames as follows:

This subclause defines rules that shall be followed by an RSNA STA~~s~~ on a link with a given peer STA ~~that implement Management Frame protection and have dot11RSNAActivated equal to true~~. In this subclause “peer STA” refers to the AP a non-AP STA is associated with, a STA that has associated with an AP, a STA in the same IBSS or PBSS, or a TDLS peer STA.

A STA with dot11RSNAProtectedManagementFramesActivated equal to false shall transmit and receive unprotected individually addressed robust Management frames to and from any ~~associated~~peer STA and shall discard protected individually addressed robust Management frames received from any ~~associated~~peer STA.

A STA with dot11RSNAProtectedManagementFramesActivated equal to true and dot11RSNAUnprotectedManagementFramesAllowed equal to true shall transmit and receive unprotected individually addressed robust Management frames to and from any ~~associated~~peer STA that advertised MFPC = 0 and shall discard protected individually addressed robust Management frames received from any ~~associated~~peer STA that advertised MFPC = 0.

A STA with dot11RSNAProtectedManagementFramesActivated equal to true and dot11RSNAUnprotectedManagementFramesAllowed equal to true shall transmit and receive protected individually addressed robust Management frames to and from any ~~associated~~peer STA that advertised MFPC = 1, shall discard unprotected individually addressed robust Action frames received from any peer STA that advertised MFPC = 1, and shall discard unprotected individually addressed Disassociation and Deauthentication frames received from any peer STA that advertised MFPC = 1 after the PTK ~~and IGTK have~~has been installed. The receiver shall process unprotected individually addressed Disassociation and Deauthentication frames before the PTK ~~and IGTK are~~has been installed.

A STA with dot11RSNAProtectedManagementFramesActivated equal to true and dot11RSNAUnprotectedManagementFramesAllowed equal to false shall transmit and receive protected individually addressed robust Action frames to and from any peer STA, shall not transmit unprotected individually addressed robust Action frames to any peer STA, and shall discard unprotected individually addressed robust Action frames received from a peer STA after the PTK ~~and IGTK have~~has been installed. The receiver shall process unprotected individually addressed Disassociation and Deauthentication frames before the PTK ~~and IGTK aree~~has been installed.

A STA with dot11RSNAProtectedManagementFramesActivated equal to true shall protect transmitted group addressed robust Management frames using the group management cipher suite.

A STA with dot11RSNAProtectedManagementFramesActivated equal to true shall discard group addressed robust Management frames received from any ~~associated~~peer STA that advertised MFPC = 1 if the frames are unprotected or if a matching IGTK is not available.

A STA with dot11RSNAProtectedManagementFramesActivated equal to true and dot11RSNAUnprotectedManagementFramesAllowed equal to false shall discard received group addressed robust Management frames that are unprotected or ~~for which~~if a matching IGTK is not available.

A STA with dot11RSNAProtectedManagementFramesActivated equal to false shall transmit group addressed robust Management frames unprotected and shall ignore the protection on received group addressed robust Management frames.

NOTE—BIP does not provide protection against forgery by associated (if in an infrastructure BSS, and optionally in a PBSS) and authenticated STAs. A STA that has left the ~~group~~BSS can successfully forge group addressed robust Management frames until the IGTK is updated.

Protection of group addressed robust Management frames shall be provided by a service in the MLME as described in 11.12 (Group addressed management frame protection procedures).

Management frame protection cannot be applied until the PTK and (except for a TDLS direct link) IGTK has been established with the STA. A STA shall not transmit robust Action frames until it has installed the PTK for the peer STA, or in the case of group addressed frames, has installed the IGTK. The STA shall discard any robust Action frames received before the PTK and IGTK, as appropriate, are installed.

Make changes as follows:

**9.4.2.24.4 RSN capabilities**

— Bit 6: MFPR. A STA sets this bit to 1 to advertise that protection of robust Management frames is mandatory~~. A STA sets this bit to 1 when dot11RSNAProtectedManagementFramesActivated is~~ ~~true and dot11RSNAUnprotectedManagementFramesAllowed is false~~; otherwise it sets this bit to 0. If a STA sets this bit to 1, then that STA only allows RSNAs with STAs that provide Management Frame Protection.

— Bit 7: MFPC . A STA sets this bit to 1 ~~when dot11RSNAProtectedManagementFramesActivated is true~~ to advertise that protection of robust Management frames is enabled; otherwise it sets this bit to 0.

**9.4.2.117 Authenticated Mesh Peering Exchange element**

The IGTKdata field is ~~present when dot11RSNAProtectedManagementFramesActivated equals true~~optional.

**11.13 SA Query procedures**

If ~~dot11RSNAProtectedManagementFramesActivated is true~~ a STA and its peer on a given link both set the MFPC bit to 1, then the STA shall support the SA Query procedure.

**12.5.3.4.4 PN and replay detection**

c) If ~~dot11RSNAProtectedManagementFramesActivated is true,~~ the recipient set the MFPC bit on a given link to 1, it 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 a single replay counter for received individually addressed robust PV1 Management frames and shall use the PN from the received frame to detect replays.

d) […] If ~~dot11RSNAProtectedManagementFramesActivated is true,~~ the receiver set the MFPC bit on a given link to 1, it 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.

**12.5.5.4.4 PN and replay detection**

***[ditto]***

In 11.2.3.18 AP power management change “(Re)Association Requests frames” to “(Re)Association Request frames”.

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 199, 200, 202 in <this document>, which clarify the MFP behaviour for TDLS, IBSS, MBSS and PBSS.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 225Mark RISON12 | Can non-QoS Data frames be encrypted with CCMP/GCMP (and TKIP) and if so what value is used for the priority (e.g. in the nonce/MIC), since at the MA SAP it might be "Contention"? | As it says in the comment |

Discussion:

For CCMP, the QC field is optional in the AAD, per 12.5.3.3.3 Construct AAD: “7) QC – QoS Control field contains the MSDU priority, if present.” and Table 12-3—AAD length for PV0 MPDUs; the nonce per 12.5.3.3.4 Construct CCM nonce contains “the priority value of the MPDU” which is defined in 12.5.3.3.1 General to be 0 for non-QoS Data frames (see b)3)).

For GCMP, the AAD is the same as for CCMP, per 12.5.5.3.3 Construct AAD; the nonce does not have a priority, per 12.5.5.3.4 Construct GCM nonce.

For TKIP, the MIC is computed over a structure that includes a 1-octet priority (see Figure 12-9—TKIP MIC processing format in 12.5.2.3.2 Motivation for the TKIP MIC), which “refers to the priority parameter of the MA-UNITDATA.request primitive”. It is not clear what this means if the priority parameter is Contention. There are statements that “At QoS STAs associated in a QoS BSS, MSDUs with a priority of Contention are considered equivalent to MSDUs with TID 0. At STAs associated in a non-QoS BSS, all MSDUs with an integer priority are

considered equivalent to MSDUs with a priority of Contention.” but these do not help here (transmission of a non-QoS Data frame by a non-QoS STA or in a non-QoS BSS).

However, TKIP is deprecated and not being maintained anymore, so we can close our eyes and think of something else.

Proposed resolution:

REJECTED

Non-QoS Data frames can be encrypted by CCMP/GCMP, and in that case 0 is used for the priority (see 12.5.3.3.1, 12.5.3.3.3, 12.5.3.3.4, 12.5.5.3.3, 12.5.5.3.4).

It is not clear whether non-QoS Data frames can be encrypted by TKIP, and if they can, what is used for the priority (see 12.5.2.3.2 and 5.1.1.3). However, TKIP is deprecated and not being maintained anymore.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 179Mark RISON12.7.6.82648.62 | "Message 3 differs from message 2 bynot asserting the Ack bit and from message 4 by asserting the Ack Bit." -- asserting bits is not defined, the field is called Key Ack and Bit has the wrong capitalisation | Change to "Message 3 differs from message 2 byhaving the Key Ack bit of the Key Information field set to 0 and from message 4 by having the Key Ack bit set to 1." In 12.7.6.6 change "in which the Ack bit is 1" to "in which the Key Ack bit of the Key Information field is 1". In 12.7.10.3 change "have the Ack bit set to 1" to "have the Key Ack bit of the Key Information field set to 1" and "the Ack bit is 0" to ""the Key Ack bit is 0" |
| CID 180Mark RISON12.7.6.82648.62 | "Message 3 differs from message 2 bynot asserting the Ack bit and from message 4 by asserting the Ack Bit." -- wait, what, how can M3 differ from M2 by not asserting something and also from M4 by asserting the same thing? | Err, dunno. What is this trying to say? |

Discussion:

That statement is obviously nonsensical.

More generally, the Key MIC bit is not useful as a discriminant anymore, because it is not set if an AEAD cipher is used (i.e. for FILS). The information given in 12.7.6.1 General is duplication and has rotted. Here is what the bits are in the Key Information field for the 4WH, the GKH and EAPOL-Key request frames including Michael MIC failure reports (TKIP only):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Secure (b9) | Key MIC (b8) | Key Ack (b7) | Install [pairwise] (b6) | Key Type (b3) | Encrypted Key Data (b12) | Request (b11) | Error (b10) |
| M1 | 0? | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| M2 | 0\* | 1 iff !AEAD | 0 | 0 | 1 | 1 iff AEAD | 0 | 0 |
| M3 | 1 | 1 iff !AEAD | 1 | 1 (mostly) | 1 | 1 | 0 | 0 |
| M4 | 1 | 1 iff !AEAD | 0 | 0 | 1 | 1 iff AEAD | 0 | 0 |
| G1 | 1 | 1 iff !AEAD | 1 | 0 | 0 | 1 | 0 | 0 |
| G2 | 1 | 1 iff !AEAD | 0 | 0 | 0 | 1 iff AEAD | 0 | 0 |
| Request (not Michael MIC failure) | 1? | ? | 0 | ? | any | ? | 1 | 0? |
| Michael MIC failure | 1 | 1 | 0 | 0? | 0? | 0? | 1 | 1 |

\* Some Supplicant implementations known to set the Secure bit to 1 in M2 when rekeying, however. This means that M2 and M4 cannot be distinguished using the Key Information field. (It is not known whether any Authenticator implementations set the Secure bit to 1 in M1 when rekeying.)

Proposed changes:

Change 12.7.2 EAPOL-Key frames as follows:

7) Secure (bit 9) is set to 1 once ~~the initial key exchange is complete~~ transmissions can be secured.

The Authenticator shall set the Secure bit to 0 in message 1 of the initial 4-way handshake, and should set the Secure bit to 0 in message 1 of a rekeying 4-way handshake ~~all EAPOL-Key frames sent before the Supplicant has the PTK and the GTK~~. The Authenticator shall set the Secure bit to 1 in all other EAPOL-Key frames ~~it sends to the Supplicant containing the last key needed to complete the Supplicant’s initialization~~; this includes message 3 of the 4-way handshake.

The Supplicant shall set the Secure bit to 0 in message 2 of the initial 4-way handshake, and should set the Secure bit to 0 in message 2 of a rekeying 4-way handshake ~~all EAPOL-Key frames it sends before it has the PTK and the GTK and before it has received an EAPOL-Key frame from the Authenticator with the Secure bit equal to 1 (this should be before receiving message 3 of the 4-way handshake)~~. The Supplicant shall set the Secure bit to 1 in all other EAPOL-Key frames ~~sent after this until it loses the security association it shares with the Authenticator~~; this includes message 4 of the 4-way handshake and EAPOL-Key request frames.

8) Error (bit 10) is set to 1 by a Supplicant to report that a MIC failure occurred in a TKIP MSDU. Otherwise the Error bit is set to 0. ~~In case of a MIC failure, a~~The Supplicant shall not set this bit to 1 ~~only when~~unless the Request (bit 11) is set to 1. The Authenticator shall not set this bit to 1.

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, i.e. a frame with the Error bit (bit 10) also set to 1. 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 Supplicant shall not set this bit to 1 unless the Secure bit (bit 9) is set to 1, i.e. EAPOL-Key request frames shall be secured. The Authenticator shall ~~never~~not set this bit to 1.

In a Michael MIC Failure Report frame, setting the bit to 1 is not a request to initiate a new handshake. However, the recipient may initiate a new handshake on receiving such a message.

In 12.7.6.8 4-way handshake analysis change:

It is critical to the correctness of the 4-way handshake that at least one bit differs in each message. Within the 4-way handshake, message 1 can be recognized as the only one in which the Key MIC bit is 0, meaning message 1 does not include the MIC, while message 2 to message 4 do. Message 3 differs from message 2 by not asserting the Ack bit and from message 4 by asserting the Ack Bit. Message 2 differs from message 4 by including the RSNE.

Request messages are distinct from 4-way handshake messages because the former asserts the Request bit and 4-way handshake messages do not. Group key handshake messages are distinct from 4-way handshake messages because they assert a different key type.

to:

It is critical to the correctness of the 4-way handshake that at least one bit differs in each message. Within the 4-way handshake, the messages are distinguished by the Encrypted Key Data bit (0 in message 1 only), the Key Ack bit (1 in message 1 and message 3 only) and the presence of RSNEs and/or Multi-Band elements (in message 2 but not in message 4).

Group key handshake messages are distinct from 4-way handshake messages because they set the Key Type bit differently, and are distinguished from each other by the Key Ack bit (1 in message 1 only). Request messages are distinct from 4-way handshake and group key handshake messages because they set the Request bit differently.

Change 12.7.4 EAPOL-Key frame notation as follows:

The following notation is used throughout the remainder of 12.7 (Keys and key distribution) and 13.4 (FT initial mobility domain association) to represent EAPOL-Key frames:

EAPOL-Key(S, M, A, I, K, Reserved, KeyRSC, ANonce/SNonce, MIC, {Key Data})

where

S ~~means~~indicates whether ~~the initial key exchange is complete; t~~ transmissions can be secured. This is the Secure bit of the Key Information field

M ~~means~~indicates whether the MIC is available in the message ~~[This should be set in all messages except message 1 of a 4-way handshake.~~ This is the Key MIC bit of the Key Information field. This parameter is ignored, and the Key MIC bit is set to 0, when using an AEAD cipher.

NOTE—An AEAD cipher is used w~~W~~hen the negotiated AKM is 00-0F-AC:14, 00-0F-AC:15, 00-0F-AC:16, or 00-0F-AC:17~~, this Key MIC bit is set to 0 regardless of the M parameter value~~.

A ~~means~~indicates whether a response is required to this message. ~~This is used when the receiver should respond to this message.~~ This is the Key Ack bit of the Key Information field~~.~~

I ~~is the Install bit:~~ indicates whether to install ~~(1) or not install (0) for~~ the pairwise key. This is the Install bit of the Key Information field.~~]~~

K is the key type: P (Pairwise), G (Group)~~; t~~. This is the Key Type bit of the Key Information field

Reserved is reserved

KeyRSC is the key RSC~~; t~~. This is the Key RSC field

ANonce/SNonce is the Authenticator or Supplicant nonce, respectively ~~(~~. This is the Key Nonce field~~.~~

MIC is the integrity check~~, which is generated using the KCK~~. This is the Key MIC field of the EAPOL-Key frame, where *null* represents the absence of the field. This parameter is ignored, and the field is absent, when using an AEAD cipher

~~When the negotiated AKM is 00-0F-AC:14, 00-0F-AC:15, 00-0F-AC:16, or 00-0F-AC:17, this Key MIC field is not included regardless of the MIC parameter value.)~~

At 2645.24, change “Key Nonce = 0” to “Key Nonce = 0, or SNonce – same as message 2”.

At 2639.8 change:

Message 4: Supplicant → Authenticator: EAPOL-Key(1,1,0,0,P,0,0,0,MIC,{})

to

Message 4: Supplicant → Authenticator: EAPOL-Key(1,1,0,0,P,0,0,0 or SNonce,MIC,{})

Change 12.7.6.1 General as follows:

RSNA defines a protocol using EAPOL-Key frames called the *4-way handshake*. The handshake completes the IEEE 802.1X authentication process. The information flow of the 4-way handshake is as follows (see 12.7.4):

Message 1: Authenticator → Supplicant: EAPOL-Key(0,0,1,0,P,0,0,ANonce,~~0~~*null*,{} or {PMKID})

Message 2: Supplicant → Authenticator: EAPOL-Key(0,1,0,0,P,0,0,SNonce,MIC,{RSNE} or {RSNE, OCI KDE} or {RSNE, RSNXE} or {RSNE, OCI KDE, RSNXE})

Message 3: Authenticator → Supplicant: EAPOL-Key(1,1,1,1,P,0,KeyRSC,ANonce,MIC,{RSNE,GTK[N]} or {RSNE, GTK[N], OCI KDE} or {RSNE, GTK[N], RSNXE} or {RSNE, GTK[N], OCI KDE, RSNXE})

Message 4: Supplicant → Authenticator: EAPOL-Key(1,1,0,0,P,0,0,0,MIC,{}).

NOTE—The I parameter and MIC parameter are ignored when using an AEAD cipher (see 12.7.4).

The FT initial mobility domain association uses the FT 4-way handshake to establish an initial PTKSA,

GTKSA, if management frame protection is enabled, an IGTKSA, and if beacon protection is enabled, a BIGTKSA, that is based on this protocol. The FT 4-way handshake protocol is described in 13.4 (FT initial mobility domain association).

The following apply for the remainder of Subclause 12.7.6:

~~— EAPOL-Key(.) denotes an EAPOL-Key frame conveying the specified argument list, using the notation introduced in 12.7.4 (EAPOL-Key frame notation).~~

— ANonce is a nonce ~~that~~from the Authenticator ~~contributes~~ for PTK generation. ~~ANonce has the same value in message 1 and message 3.~~

— SNonce is a nonce from the Supplicant for PTK generation.

~~— P means the pairwise bit is set.~~

~~— The MIC is computed over the body of the EAPOL-Key frame (with the Key MIC field first zeroed before the computation) using the KCK defined in 12.7.1.3 (Pairwise key hierarchy) for PTK generation.~~

— RSNE represents the appropriate RSNEs.

~~— GTK[N] represents the GTK with its key identifier.~~

— The OCI KDE contains the ~~current~~ operating channel information for the operating channel in which the EAPOL-Key frame is sent. ~~OCI KDE is present when dot11RSNAOperatingChannelValidationActivated is true on the Supplicant in Message 2 and Authenticator in Message 3. Otherwise it is absent.~~

~~— The RSNXE, when included in message 2, contains the RSNXE that the Supplicant sent in its (Re)Association Request frame, and when included in message 3, contains the RSNXE that the Authenticator sent in its Beacon or Probe Response frame. RSNXE is present in message 2 if this element is present in the (Re)Association Request frame that the Supplicant sent, and is present in message 3 if this element is present in the Beacon or Probe Response frame that the Authenticator sent.~~

Change 12.7.7.1 General as follows:

The Authenticator uses the ~~G~~group key handshake to send a new GTK and, if management frame protection is negotiated, a new IGTK, and if beacon protection is enabled, a new BIGTK to the Supplicant. ***<remove para break>*** The Authenticator may initiate the exchange when a Supplicant is disassociated or deauthenticated. The information flow of the group key handshake is as follows (see 12.7.4):

Message 1: Authenticator → Supplicant: EAPOL-Key(1,1,1,0,G,0,Key RSC,0, MIC, {GTK[N], IGTK[M], BIGTK[Q]})

Message 2: Supplicant → Authenticator: EAPOL-Key(1,1,0,0,G,0,0,0,MIC,{})

NOTE—The I parameter and MIC parameter are ignored when using an AEAD cipher (see 12.7.4).

The following apply for the remainder of Subclause 12.7.7:

— Key RSC denotes the last TSC or PN sent using the GTK.

~~— GTK[N] denotes the GTK with its key identifier as defined in 12.7.2 (EAPOL-Key frames) using the KEK defined in 12.7.1.3 (Pairwise key hierarchy) and associated IV.~~

~~— IGTK[M], when present, denotes the IGTK with its key identifier as defined in 12.7.2 (EAPOL-Key frames) using the KEK defined in 12.7.1.3 (Pairwise key hierarchy) and associated IV.~~

~~— BIGTK[Q], when present, denotes the BIGTK with its key identifier as defined in 12.7.2 (EAPOL-Key frames) using the KEK defined in 12.7.1.3 (Pairwise key hierarchy) and associated IV.~~

~~— The MIC is computed over the body of the EAPOL-Key frame (with the MIC field zeroed for the computation) using the KCK defined in 12.7.1.3 (Pairwise key hierarchy).~~

— The OCI KDE ~~represents~~contains the ~~current~~ operating channel information ~~using~~ for the operating channel in which the EAPOL-Key frame is sent. ~~OCI KDE is included when dot11RSNAOperatingChannelValidationActivated is true on the STA sending the message.~~

Change 13.4.2 FT initial mobility domain association in an RSN as follows:

The R1KH and S1KH then perform an FT 4-way handshake. ~~The EAPOL-Key frame notation is defined~~

~~in 12.7.4 (EAPOL-Key frame notation).~~ The information flow of the FT 4-way handshake is as follows (see 12.7.4):

R1KH→S1KH: EAPOL-Key(0, 0, 1, 0, P, 0, 0, ANonce, ~~0~~*null*, {})

S1KH→R1KH: EAPOL-Key(0, 1, 0, 0, P, 0, 0, SNonce, MIC, {RSNE[PMKR1Name], MDE, FTE, RSNXE})

R1KH→S1KH: EAPOL-Key(1, 1, 1, 1, P, 0, 0, ANonce, MIC, {RSNE[PMKR1Name], MDE, GTK[N], IGTK[M], BIGTK[Q], FTE, TIE[ReassociationDeadline], TIE[KeyLifetime], RSNXE})

S1KH→R1KH: EAPOL-Key(1, 1, 0, 0, P, 0, 0, 0, MIC, {})

NOTE—The I parameter and MIC parameter are ignored when using an AEAD cipher (see 12.7.4).

The message sequence is described in 12.7.6 (4-way handshake).

Change 12.7.2 EAPOL-Key frames as follows:

h) **Key MIC.** When ~~the negotiated AKM is not 00-0F-AC:14, 00-0F-AC:15, 00-0F-AC:16, or 00-0F-AC:17~~ not using an AEAD cipher, the ~~EAPOL K~~key MIC is a MIC of the EAPOL-Key frame~~s~~, from and including the EAPOL protocol version field 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 1, the Key Data field is encrypted prior to computing the MIC. When using an AEAD cipher, the ~~EAPOL K~~key MIC is not present. The length of this field depends on the negotiated AKM as defined in 12.7.3 (EAPOL-Key frame construction and processing).

NOTE—An AEAD cipher is used when the negotiated AKM is 00-0F-AC:14, 00-0F-AC:15, 00-0F-AC:16, or 00-0F-AC:17.

In 12.7.6.2 4-way handshake message 1 change the second “Key MIC = 0” to “Key MIC = not present” (the one at 2640.28).

Change “Key MIC = Not present” to “Key MIC = not present” (5x in 12.7).

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 179 and CID 180 in <this document>, which address this and many other issues with the 4WH and the GKH.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 454Mark RISON | There are references to mesh beacons (including three to "mesh Beacon frame"s and one to "Mesh Beacon frame") but mesh doesn't seem to have its own beacons (unlike DMG and S1G) | Delete "mesh"/"Mesh" before "beacon"/"Beacon" except in "esh beacon collision avoidance" and "esh beaconing" and MIB attribute names |

Discussion:

SAKODA Kazuyuki has confirmed that:

Yes, mesh STA reuses regular beacon frames. It sounds reasonable to delete “mesh”/”Mesh” before “beacon”/”Beacon” except “beacon collision avoidance” and “esh beaconing” and MIB names

Proposed resolution:

REVISED

Delete “mesh” at:

2370.11, in “the most recently received mesh beacon from the peer”

2372.3, in “its mesh Beacon frame” and in “a mesh Beacon frame”

2373.26, in “the STA’s most recent mesh Beacon frame”

At 2416.49, change “The Mesh Beacon frame shall not include a QMF Policy element.” to “A mesh STA shall not include a QMF Policy element in a Beacon frame.”

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 162Mark RISON | In Table 9-43--Action frame body and Action No Ack frame body, 11.12 Group addressed management frame protection procedures, 12.5.3(.1) CTR with CBC-MAC protocol (CCMP), 12.5.5(.1) GCM protocol (GCMP) it is not sufficiently clear that "Group addressed privacy" only applies in the context of an MBSS | In 11.12 change "For group addressed Management frames that" to "In an MBSS, for group addressed Management frames that". Add "(MBSS only)" in each of the other locations |

Discussion:

SAKODA Kazuyuki has commented that:

Probably, it is more reader friendly to mention that “Group addressed privacy is only for MBSS” somewhere in the standard. But, I am quite not sure if adding “In an MBSS” or “MBSS only” at many places will be a right solution. If we say “in an MBSS”, then readers may think “then what happens in other BSSs?”

The answer to that question is that in 11.12 what happens in other BSSes is stated immediately afterwards, and in other locations group addressed privacy does not apply in other BSSes.

Proposed changes:

In 11.12 Group addressed management frame protection procedures:

~~F~~In an MBSS, for group addressed Management frames that are specified with Yes in the Group Addressed Privacy column of Table 9-51 (Category values), the group addressed frame protection service shall take the following actions:

— The frames shall be encapsulated and protected with the MGTK using the group cipher negotiated during the AMPE exchange.

For all other group addressed Management frames, the group addressed frame protection service shall take the following actions:

In Table 9-43—Action frame body and Action No Ack frame body:

The MME is present when management frame protection is enabled at the AP, the frame is a group addressed robust Action frame, and (MBSS only) the category of the Action frame does not support group addressed privacy as indicated by Table 9-51 (Category values).

In 12.5.3.1 General (under 12.5.3 CTR with CBC-MAC protocol (CCMP)):

When CCMP is selected as the RSN pairwise cipher and management frame protection is negotiated, individually addressed robust Management frames and (MBSS only) the group addressed Management frames that receive “Group Addressed Privacy” as indicated in Table 9-51 (Category values) shall be protected with CCMP.

In 12.5.5.1 GCMP overview (under 12.5.5 GCM protocol (GCMP)):

When GCMP is selected as the RSN pairwise cipher and management frame protection is negotiated, individually addressed robust Management frames and (MBSS only) the group addressed Management frames that receive “Group Addressed Privacy” as indicated in Table 9-51 (Category values) shall be protected with GCMP.

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 162 in <this document>, which explicitly show the locations where the additional text suggested by the commenter is inserted.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 387Mark RISON12 | REVmd CID 4049 resolution directs use of both 1 and 2 if pairwise is "use group" and group is CCMP. Jouni suggested during teleconf on 2020-06-12 that (a) that combination should not be allowed and (b) the 2 option should not allow "or group cipher" in the condition | As it says in the comment |
| CID 406Mark RISON12.7.22629.1 | "i) The value 1 shall be used for all EAPOL-Key frames to a STA when the negotiated AKMis 00-0F-AC:1 or 00-0F-AC:2 and the pairwise cipher is TKIP or "Use group ciphersuite". In this case, the "Deprecated" row in Table 12-10 (Integrity and key wrap algo-rithms) is used.ii) The value 2 shall be used for all EAPOL-Key frames to a STA when the negotiated AKMis 00-0F-AC:1 or 00-0F-AC:2 and either the pairwise or the group cipher is an enhanceddata cryptographic encapsulation mechanism other than TKIP. In this case, the matchingrow in Table 12-10 (Integrity and key wrap algorithms) is used." means you have to use both value 1 and value 2 if pairwise = "use group" and group = CCMP | Add a statement to say that the pairwise = "use group" and group = CCMP combination is not allowed |

Discussion:

We have in 12.7.2:

1) Key Descriptor Version (bits 0–2) shall be set to 0 on all transmitted EAPOL-Key frames indicating that the EAPOL-Key frame is constructed and processed according to the negotiated AKM as described in Table 12-10 (Integrity and key wrap algorithms) except under the following circumstances:

i) The value 1 shall be used for all EAPOL-Key frames to a STA when the negotiated AKM is 00-0F-AC:1 or 00-0F-AC:2 and the pairwise cipher is TKIP or “Use group cipher suite”. In this case, the “Deprecated” row in Table 12-10 (Integrity and key wrap algorithms) is used.

ii) The value 2 shall be used for all EAPOL-Key frames to a STA when the negotiated AKM is 00-0F-AC:1 or 00-0F-AC:2 and either the pairwise or the group cipher is an enhanced data cryptographic encapsulation mechanism other than TKIP. In this case, the matching row in Table 12-10 (Integrity and key wrap algorithms) is used.

iii) The value 3 shall be used […]

So if the AKM is 00-0F-AC:1 or 00-0F-AC:2, the pairwise cipher is “Use group cipher suite” and the group cipher suite is CCMP, then 1)i) is saying you use a key descriptor version of 1, and 1)ii) is saying you use a key descriptor version of 2.

The rules on “Use group cipher suite” are:

A STA that has associated with management frame protection enabled shall not use pairwise cipher suite selectors WEP-40, WEP-104, TKIP, or “Use group cipher suite.”

An S1G STA shall not use the pairwise cipher suite selectors WEP-40, WEP-104, TKIP, or “Use group cipher suite”.

The cipher suite selector 00-0F-AC:0 (Use group cipher suite) is valid only as the pairwise cipher suite. An AP specifies the selector 00-0F-AC:0 (Use group cipher suite) for a pairwise cipher suite if it does not support any pairwise cipher suites. If an AP specifies 00-0F-AC:0 (Use group cipher suite) as the pairwise cipher selection, this is the only pairwise cipher selection the AP advertises.

If any cipher suite other than TKIP, WEP-104, or WEP-40 is enabled, then the AP supports pairwise keys, and thus the suite selector 00-0F-AC:0 (Use group cipher suite) is not a valid option.

An HT STA shall not use either of the pairwise cipher suite selectors: “Use group cipher suite” or TKIP to communicate with another HT STA.

When the GTK is used to encrypt individually addressed traffic (the selectable cipher suite is “Use group cipher suite”), the GTKSA is bidirectional.

In order to accommodate local security policy, a STA may choose not to associate with an AP that does not support any pairwise cipher suites. An AP may indicate that it does not support any pairwise keys by advertising 00-0F-AC:0 (Use group cipher suite) as the pairwise cipher suite selector.

Table 12-10 (Integrity and key wrap algorithms) indicates the particular algorithms to use when constructing and processing EAPOL-Key frames and FT authentication sequence. The AKM of “Deprecated” indicates an AKM of 00-0F-AC:1 or 00-0F-AC:2 when either TKIP or “Use group cipher suite” is the negotiated pairwise cipher.

It is arguable that the highlighted text above disallows the pairwise cipher being “Use group cipher suite” and the group cipher suite being CCMP. However “is enabled” is a bit vague.

Proposed resolution:

REVISED

At 1088.20 change “If any cipher suite other than TKIP, WEP-104, or WEP-40 is enabled, then the AP supports pairwise keys, and thus the suite selector 00-0F-AC:0 (Use group cipher suite) is not a valid option.” to “If an AP advertises a group cipher suite other than TKIP, WEP-104, or WEP-40, then the AP supports pairwise keys, and thus the pairwise suite selector 00-0F-AC:0 (Use group cipher suite) is not a valid option.”

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 432Mark RISON | There are references to "enhanced data cryptographic encapsulation mechanisms" but this term is never actually defined. The examples seem to suggest it means "non-WEP". If so, just say that. If not, define | Change "enhanced data cryptographic encapsulation" case-insensitively to "RSNA" (4x in C4, 8x in C12) |

Discussion:

There is a definition of sorts in 4.3.8 Robust security network association (RSNA):

— Enhanced data cryptographic encapsulation mechanisms, such as counter mode with cipher-block chaining message authentication code protocol (CCMP), Galois/counter mode protocol (GCMP), and, optionally, temporal key integrity protocol (TKIP)

However, it would be clearer just to refer to RSNAs.

The proposed changes are:

**4.3.8 Robust security network association (RSNA)**

The following features are defined for an RSNA:

— Enhanced authentication mechanisms for STAs

— Key management algorithms

— Cryptographic key establishment

— ~~Enhanced data cryptographic encapsulation~~RSNA mechanisms, such as counter mode with cipher-block chaining message authentication code protocol (CCMP), Galois/counter mode protocol (GCMP), and, optionally, temporal key integrity protocol (TKIP)

**4.5.4 Access control and data confidentiality services**

**4.5.4.1 General**

An RSNA uses the IEEE 802.1X authentication service along with ~~enhanced data cryptographic encapsulation~~RSNA, such as TKIP, CCMP, and GCMP, to provide access control. The IEEE 802.11 station management entity (SME) provides key management via an exchange of EAPOL-Key frames. Data confidentiality and data integrity are provided by RSN key management together with the ~~enhanced data cryptographic encapsulation~~RSNA mechanisms.

**4.5.4.7 Replay detection**

The replay detection mechanism defines a means by which a STA that receives a Data or robust Management frame from another STA can detect whether the received frame is an unauthorized retransmission. This replay protection mechanism is provided for Data frames for STAs that use ~~enhanced data cryptographic encapsulation~~RSNA mechanisms. The replay protection mechanism is also provided for robust Management frames for STAs that use CCMP, GCMP, and broadcast/multicast integrity protocol (BIP).

**12.7.1 Key hierarchy**

**12.7.1.1 General**

RSNA defines the following key hierarchies:

a) Pairwise key hierarchy, to protect individually addressed traffic

b) GTK, a hierarchy consisting of a single key to protect group addressed traffic

NOTE 1—Pairwise key support with ~~enhanced data cryptographic encapsulation~~RSNA mechanisms allows a receiving STA to detect MAC address spoofing and data forgery.

An RSNA STA shall support at least one pairwise key for any <TA,RA> pair for use with ~~enhanced data cryptographic encapsulation~~RSNA mechanisms. The <TA,RA> identifies the pairwise key, which does not correspond to any WEP key identifier.

In a a mixed environment, an AP may simultaneously communicate with some STAs using WEP with shared WEP keys and to STAs using ~~enhanced data cryptographic encapsulation~~RSNA mechanisms with pairwise keys.

STAs using ~~enhanced data cryptographic encapsulation~~RSNA mechanisms in a TSN shall support pairwise keys and WEP default key 0 simultaneously. It is invalid for the STA to negotiate the No Pairwise subfield when an ~~enhanced data cryptographic encapsulation~~RSNA mechanism other than TKIP is one of the configured ciphers.

**12.7.2 EAPOL-Key frames**

ii) The value 2 shall be used for all EAPOL-Key frames to a STA when the negotiated AKM is 00-0F-AC:1 or 00-0F-AC:2 and either the pairwise or the group cipher is an ~~enhanced data cryptographic encapsulation~~RSNA mechanism other than TKIP. In this case, the matching row in Table 12-10 (Integrity and key wrap algorithms) is used.

However, there is also:

**4.5.4.5 Key management**

The enhanced data confidentiality, data authentication, and replay protection mechanisms require fresh cryptographic keys and corresponding security associations.

Presumably “confidentiality” here is the same thing as “cryptographic encapsulation” elsewhere.

Proposed resolution:

REVISED

Make the proposed changes and additionally change “enhanced data confidentiality” to “RSNA” in 4.5.4.5.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 507Mark RISON12 | Message 1 uses the following values for each of the EAPOL-Key frame fields:" -- "Key Data = PMKID for the PMK being used during PTK generation" -- Jouni said in Vienna that it can contain other stuff, and that it should be referring to PMKID KDE. Also, should have general statement for all the Key Data = lines to say this is the minimum set of things that need to be included in the message, but other stuff (e.g. VS) may also be included | As it says in the comment |

Discussion:

The PMKID is carried in a PMKID KDE.

We need to be clear on whether the {Key Data} contents shown are exhaustive (not, since they do not show VS elements/KDEs).

Proposed resolution:

REVISED

In 12.7.6.2 change “Key Data = PMKID for the PMK being used during PTK generation” to “Key Data = PMKID KDE containing the PMKID for the PMK being used during PTK generation”.

At the end of the bullet list in 12.7.6.1 General add a bullet “One or more vendor specific KDEs and/or Vendor Specific elements may appear in the {Key Data}.”

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 296Mark RISON | Stuff that refers to the United States / USA in the context of FCC stuff needs to also cover the other places that fall under FCC jurisdiction (e.g. Puerto Rico). Reference:https://en.wikipedia.org/wiki/Federal\_Communications\_Commission :The FCC's mandated jurisdiction covers the 50 states, the District of Columbia, and the territories of the United States.https://en.wikipedia.org/wiki/Territories\_of\_the\_United\_States :Populated US territories are AS, GU, MP, PR and VI | As it says in the comment |

Discussion:

The FCC’s remit extends beyond the USA, and similarly the CEPT’s remit extends beyond the EU.

Proposed resolution:

REVISED

Change “the United States” to “the United States and its territories” in 4.3.12.3 Contention based protocol (CBP) in nonexclusively licensed bands, 5.1.4 MSDU format, 18.4.3 Operating channel frequencies, 22.3.14 Channelization, B.4.3 IUT configuration, Table D-1—Regulatory requirement list, Table D-3—Maximum STA transmit power classification for the 5.85–5.925 GHz band in the United States (caption itself), Table D-4—Maximum STA transmit power and maximum BW allowed for the S1G band, E.1 Country information and operating classes (just before Table E-1), Table E-1—Operating classes in the United States (caption itself), Table E-5—S1G operating classes, E.2.2 3650–3700 MHz band in the United States (heading itself), E.2.3 5.9 GHz band in the United States (5.850–5.925 GHz) (heading itself), E.2.5 TVWS band in the United States and Canada (54–698 MHz) (heading itself), Table E-9—Device Identification Information Value fields, E.2.5 TVWS band in the United States and Canada (54–698 MHz) (final NOTE).

Change “the China, United States and Europe” to “China, the United States and its territories, and Europe” in 15.4.4.2 Operating frequency range, 16.3.6.2 Operating frequency range.

Change “USA” to “USA and its territories” at 4167.62, 4168.50, 4297.19 (also prepend “the”), 4324.33 (also prepend “the”).

In 3.3 Definitions specific to IEEE 802.11 operation in some regulatory domains and 3.5 Acronyms and abbreviations in some regulatory domains, at the end of the first para add: “US is to be understood as also including the United States’ territories, and EU is to be understood as including all CEPT member states”. At the end of 4353.2 and 4354.2 add “US is to be understood as also including the United States’ territories.”.

Change “U.S.” to “United States and its territories” in the “emergency alert system (EAS)” definition in 3.1, at 231.62 (NOTE), 11.11 DSE procedures (2x), 22.3.14 Channelization (2x), D.2.2 Transmit power levels. At 4354.35 change “NOTE 3—In the United States, an example of full Map 1 for a U.S. GDD non-AP STA” to “NOTE 3—In the United States and its territories, an example of full Map 1 for a GDD non-AP STA”.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 397Mark RISON9.2.4.3.4787.26 | "This field, in an infrastructure BSS, is the MAC address currently inuse by the STA in the AP of the BSS." needs a line for an MBSS | As it says in the comment |
| CID 391Mark RISON14 | There is no BSSID in an MBSS, so how are things that carry a BSSID handled in an MBSS context (e.g. MLME-SCAN primitives)? Is the BSSID in that case to be interpreted as the MAC address of a mesh STA (if it's not the wildcard BSSID)? | As it says in the comment |

Discussion:

SAKODA Kazuyuki has commented that:

Yes, the concept of BSSID does not exist in an MBSS, but it is very hard to describe mesh STA normative behaviors without referring to BSSID. So, the standard leaves a description saying that the BSSID is the address of the transmitter and is equal to the Data frame’s TA.

In 9.3.2.1.2 (Address and BSSID fields), P840.L1, there is a sentence saying “If the STA is a member of an MBSS, the BSSID is the address of the transmitter and is equal to the Data frame’s TA.”

Proposed resolution for CID 397:

REVISED

In 9.2.4.3.4 BSSID field before the final para add: “This field in an MBSS is set to the same value as the TA field (see 9.3.2.1.2).”

Proposed resolution for CID 391:

REVISED

At the end of 14.13.3.2 Beacon reception for mesh STA add a para: “An MBSS does not have a BSSID. Any field containing a BSSID contains the MAC address of a particular (originating or target) mesh STA, or the wildcard BSSID where permitted.”.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 315Mark RISON1.5 | Need to define all the log operators in Subclause 1.5 and use them consistently (maybe have ln, log10 and log2). From a quick grep of md/D3.2, I find- 2 instances of ln- 2 instances of log10- 4 instances of log\_10- 6 instances of log2 (OK, defined in 1.5)- 2 instances of log\_2- 5 instances of log | As it says in the comment |

Discussion:

There was no objection on the 2021-05-11 teleconf to going with the proposal to use ln, log10 and log2, unsubscripted.

Proposed changes:

REVISED

After the line with the definition of log2 (x) in 1.5 add the following lines, with x being italicised (4x):

log10 (x) is the logarithm of x to the base 10. For example, log2 (100) is 2.

ln (x) is the logarithm of x to the base of natural logarithms. For example, ln (10) is approximately 2.3.

In Equations (9-2), (19-75), (25-26), (25-35) (2x), (25-67) and in 20.4.4.1.2 Transmit EVM, 20.5.4.1.1 Transmit EVM, 23.3.17.4.2 Transmitter center frequency leakage, 24.5.4.1.1 Transmit EVM change the log\_10 (subscript) to log10 (no subscript).

In D.2.4 Transmit Mask M change the log to log10 (5x).

In K.4.2 Surplus Bandwidth Allocation change the Ln to ln.

In Equation (R-4) and in 9.4.2.5.5 ADE mode change the log\_2 (subscript) to log2 (no subscript).

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 238Mark RISON3 | All abbreviations need to be expanded on first use in each definition. For example "destination mesh station (STA): A mesh STA that is the final destination of a MAC service data unit (MSDU)" needs expansion of "MAC" | As it says in the comment. I suggest writing a script that will, given Subclauses 3.1 and 3.2 as plain text, identify missing required expansions |

Discussion:

Script written. Ignores things that are names (of frame or fields or elements, or organisations (e.g. IEEE), or enumeration values).

Proposed changes:

Change the definitions in Clause 3 as follows:

authenticator address (AA): The medium access control (MAC) address of the IEEE 802.1X Authenticator’s station (STA).

basic service set (BSS): A set of stations (STAs) that have successfully synchronized using the MLME-JOIN.request service primitive~~s~~ and one STA that has used the MLME-START.request primitive. Alternatively, a set of STAs that have used the MLME-START.request primitive specifying matching mesh profiles where the match of the mesh profiles has been verified via the scanning procedure. Membership in a BSS does not imply that wireless communication with all other members of the BSS is possible.

destination mesh station (STA): A mesh STA that is the final destination of a medium access control (MAC) service data unit (MSDU). This mesh STA might reside in a proxy mesh gate that might forward the MSDU to a STA outside of the mesh basic service set (MBSS). A destination mesh STA might be an end station as defined in IEEE Std 802.

infrastructure: An infrastructure comprises a distribution system (DS), one or more ~~more~~ access points (APs), zero or one portals, and zero or more mesh gates. It is also the logical location of distribution and integration service functions of an extended service set (ESS).

over-the-~~DS (~~distribution-system (over-the-DS) fast basic service set (BSS) transition (FT): An FT method in which the station (STA) communicates with the target access point (AP) via the current AP.

Supplicant address (SPA): The medium access control (MAC) address of the IEEE 802.1X Supplicant’s station (STA).

validated access point (AP): An AP that has either been explicitly configured as a neighbor or learned through a mechanism such as the Beacon report.

e) A Clause 17 (Orthogonal frequency division multiplexing (OFDM) PHY specification) PPDU transmitted by a VHT station (STA) using the 20 MHz transmit spectral mask defined in Clause 21 (Very high throughput (VHT) PHY specification).

b) A 40 MHz non-HT duplicate PPDU (TXVECTOR parameter CH\_BANDWIDTH equal to NON\_HT\_CBW40) transmitted by a non-very high throughput (non-VHT) station (STA) using the 40 MHz transmit spectral mask defined in Clause 19 (High-throughput (HT) PHY specification).

centralized authentication controlled (CAC) station (STA): A sub 1 GHz (S1G) non-access point (non-AP) STA ~~with dot11S1GCentralizedAuthenticationControlActivated equal to true. A CAC STA~~ that supports a CAC access point (AP) to alleviate wireless medium contention when a large number of STAs are trying to, or are expected to, reconnect to the AP at about the same time.

centralized authentication controller (CAC) access point (AP): A sub 1 GHz (S1G) AP ~~with dot11S1GCentralizedAuthenticationControlActivated equal to true. A CAC AP~~ that is able to alleviate wireless medium contention when a large number of stations (STAs) are trying to, or are expected to, reconnect to the AP at about the same time.

controlled access phase (CAP): A time period during which the hybrid coordinator (HC) maintains control of the medium. It might span multiple consecutive hybrid coordination function (HCF) controlled channel access (HCCA) transmission opportunities (TXOPs).

deep sleep mode: A mesh power management mode with respect to a neighbor peer mesh station (STA) in which a mesh ~~station (~~STA~~)~~ alternates between awake and doze states and is not expected to receive beacons from this neighbor peer mesh STA.

EAPOL-Key frame: A Data frame that carries all or part of an IEEE 802.1X Extensible Authentication Protocol (EAP) over local area network (LAN) (EAPOL) protocol data unit (PDU) of type EAPOL-Key.

EAPOL-Start frame: A Data frame that carries all or part of an IEEE 802.1X Extensible Authentication Protocol (EAP) over local area network (LAN) (EAPOL) protocol data unit (PDU) of type EAPOL-Start.

extended rate physical layer (PHY) using complementary code keying (CCK) modulation (ERP-CCK): A mode of operation of a PHY operating under Clause 18 (Extended Rate PHY (ERP) specification) rules, where the TXVECTOR/RXVECTOR parameter MODULATION~~=~~ is equal to ERP-CCK.

MODULATION=ERP-DSSS " in " extended rate physical layer (PHY) using direct sequence spread spectrum (DSSS) modulation (ERP-DSSS): A PHY operating under Clause 18 (Extended Rate PHY (ERP) specification) rules, where the TXVECTOR/RXVECTOR parameter MODULATION~~=~~ is equal to ERP-DSSS.

extended rate physical layer (PHY) using direct sequence spread spectrum (DSSS) or complementary code keying (CCK) modulation (ERP-DSSS/CCK): A PHY operating under Clause 18 (Extended Rate PHY (ERP) specification) rules, where the TXVECTOR/RXVECTOR parameter MODULATION~~=~~ is equal to ERP-CCK or ~~MODULATION=~~ERP-DSSS.

extended rate physical layer (PHY) using direct sequence spread spectrum (DSSS) or complementary code keying (CCK) modulation (ERP-DSSS/CCK): A PHY operating under Clause 18 (Extended Rate PHY (ERP) specification) rules, where the TXVECTOR/RXVECTOR parameter MODULATION~~=~~ is equal to ERP-CCK or MODULATION=ERP-DSSS.

extended rate physical layer (PHY) using orthogonal frequency division multiplexing (OFDM) modulation (ERP-OFDM): A mode of operation of a PHY operating under Clause 18 (Extended Rate PHY (ERP) specification) rules, where the TXVECTOR/RXVECTOR parameter MODULATION~~=~~ is equal to ERP-OFDM.

fast basic service set (BSS) transition (FT) 4-way handshake: A pairwise key management protocol used during FT initial mobility domain association. This handshake confirms mutual possession of a pairwise master key, the pairwise master key (PMK) R1 (PMK-R1), by two parties and distributes a group temporal key (GTK).

fast initial link setup (FILS) access point (AP): An access point that implements FILS ~~and for which dot11FILSActivated is true~~.

fast initial link setup (FILS) station (STA): A station that implements ~~fast initial link setup (~~FILS~~) and for which dot11FILSActivated is true~~.

flexible multicast stream (FMS) identifier (FMSID): An identifier assigned by the access point (AP) to a particular group addressed stream subsequent to a successful FMS ~~R~~request procedure.

future channel guidance: future channel guidance communicates likely future channel information so that stations (STAs) can efficiently move their activity when the absence of Beacon frames is noticed.

high-throughput (HT) null data physical layer (PHY) protocol data unit (PPDU) (NDP) announcement: A ~~physical layer (PHY) protocol data unit (~~PPDU~~)~~ that contains one or more +HTC frames (i.e., frames with an HT Control field) that have the HT NDP Announcement subfield equal to 1.

light sleep mode: A mesh power management mode with respect to a neighbor peer mesh station (STA) in which a mesh ~~station (~~STA~~)~~ alternates between awake and doze states and is expected to receive beacons from this neighbor peer mesh STA.

mesh power management mode tracking: Operation to observe the peering-specific mesh power management modes from the peer mesh stations (STAs) and to maintain the peering-specific mesh power management modes for each peer mesh STA.

minimum downlink transmission time (DTT) to uplink transmission time (UTT) ~~[DTT2UTT]~~ spacing: The minimum time within a power save multi-poll (PSMP) sequence between the end of a station’s (STA’s) PSMP-DTT and the start of its PSMP-UTT.

non-high-throughput (non-HT) SIGNAL field (L-SIG) transmit opportunity (TXOP) protection: A protection mechanism in which protection is established by the non-HT SIG Length and Rate fields indicating a duration that is longer than the duration of the physical layer (PHY) protocol data unit (PPDU) itself.

null data physical layer (PHY) protocol data unit (PPDU) (NDP): A ~~physical layer (PHY) protocol data unit (~~PPDU~~)~~ that carries no Data field.

null data physical layer (PHY) protocol data unit (PPDU) (NDP) 1M (NDP\_1M): An NDP carrying medium access control (MAC) information (CMI) frame that is transmitted using the S1G\_1M format.

null data physical layer (PHY) protocol data unit (PPDU) (NDP) 2M (NDP\_2M): An NDP carrying medium access control (MAC) information (CMI) frame that is transmitted using the S1G\_SHORT format.

null data physical layer (PHY) protocol data unit (PPDU) (NDP) carrying medium access control information (CMAC) PPDU: A ~~physical layer (PHY) protocol data unit (~~PPDU~~)~~ with no Data field used by the PHY to provide to the medium access control (MAC) the service of carrying MAC information in the SIGNAL field of the sub 1 GHz (S1G) PPDU.

~~pairwise master key (PMK)~~ R0 key holder (R0KH): The component of robust security network association (RSNA) key management of the Authenticator that is authorized to derive and hold the pairwise master key (PMK) R0 (PMK-R0), derive the PMK R1s (PMK-R1s), and distribute the PMK-R1s to the R1 key holders (R1KHs).

~~pairwise master key (PMK)~~ R0 key holder identifier (R0KH-ID): An identifier that names the holder of the pairwise master key (PMK) R0 (PMK-R0) in the Authenticator.

pairwise master key (PMK) R0 (PMK-R0) name (PMKR0Name): An identifier that names the PMK-R0.

~~pairwise master key (PMK)~~ R1 key holder (R1KH): The component of robust security network association (RSNA) key management of the Authenticator that receives a pairwise master key (PMK) R1 (PMK-R1) from the R0 key holder (R0KH), holds the PMK-R1, and derives the pairwise transient keys (PTKs).

~~pairwise master key (PMK)~~ R1 key holder identifier (R1KH-ID): An identifier that names the holder of a pairwise master key (PMK) R1 (PMK-R1) in the Authenticator.

pairwise master key (PMK) R1 (PMK-R1) name (PMKR1Name): An identifier that names a PMK-R1.

~~pairwise master key (PMK)~~ S0 key holder (S0KH): The component of robust security network association (RSNA) key management of the Supplicant that derives and holds the pairwise master key (PMK) R0 (PMK-R0), derives the PMK R1s (PMK-R1s), and provides the PMK-R1s to the S1 key holder (S1KH).

~~pairwise master key (PMK)~~ S0 key holder identifier (S0KH-ID): An identifier that names the holder of the pairwise master key (PMK) R0 (PMK-R0) in the Supplicant.

~~pairwise master key (PMK)~~ S1 key holder (S1KH): The component of robust security network association (RSNA) key management in the Supplicant that receives a pairwise master key (PMK) R1 (PMK-R1) from the S0 key holder (S0KH), holds the PMK-R1, and derives the pairwise transient keys (PTKs).

~~pairwise master key (PMK)~~ S1 key holder identifier (S1KH-ID): An identifier that names the holder of the pairwise master key (PMK) R1 (PMK-R1) in the Supplicant.

pairwise transient key (PTK): A concatenation of session keys derived from the pairwise master key (PMK) or from the PMK R1 (PMK-R1).

synchronized access point (AP) or personal basic service set (PBSS) control point (PCP): A China directional multi-gigabit (CDMG) AP or PCP that is operating on a 1.08 GHz channel but still transmitting its DMG Beacon frames on the relevant 2.16 GHz channel with the AP or PCP Role subfield of the Dynamic Bandwidth Control element (9.4.2.220 (Dynamic Bandwidth Control element)) set to 1 and synchronizing with the synchronizing AP or PCP on the relevant 2.16 GHz channel.

TVHT\_MODE\_1 physical layer (PHY) protocol data unit (PPDU): One of the following PPDUs: A Clause 22 (Television very high throughput (TVHT) PHY specification) TVHT\_W Very High Throughput (VHT) PPDU or TVHT\_W ~~NON\_HT~~non-high-throughput (non-HT) PPDU.

TVHT\_MODE\_2C physical layer (PHY) protocol data unit (PPDU): One of the following PPDUs: A Clause 22 (Television very high throughput (TVHT) PHY specification) TVHT\_2W Very High Throughput (VHT) PPDU or TVHT\_2W ~~NON\_HT~~non-high-throughput (non-HT) PPDU.

TVHT\_MODE\_2N physical layer (PHY) protocol data unit (PPDU): One of the following PPDUs: A Clause 22 (Television very high throughput (TVHT) PHY specification) TVHT\_W+W Very High Throughput (VHT) PPDU or TVHT\_W+W ~~NON\_HT~~non-high-throughput (non-HT) PPDU.

TVHT\_MODE\_4C physical layer (PHY) protocol data unit (PPDU): One of the following PPDUs: A Clause 22 (Television very high throughput (TVHT) PHY specification) TVHT\_4W Very High Throughput (VHT) PPDU or TVHT\_4W ~~NON\_HT~~non-high-throughput (non-HT) PPDU.

TVHT\_MODE\_4N physical layer (PHY) protocol data unit (PPDU): One of the following PPDUs: A Clause 22 (Television very high throughput (TVHT) PHY specification) TVHT\_2W+2W Very High Throughput (VHT) PPDU or TVHT\_2W+2W ~~NON\_HT~~non-high-throughput (non-HT) PPDU.

Change the following expansions in 3.4:

PMK-R0 pairwise master key~~, first level~~ R0 (first level)

PMK-R1 pairwise master key~~, second level~~ R1 (second level)

R0KH ~~PMK-~~R0 key holder (PMK-R0 key holder in the Authenticator)

R0KH-ID ~~PMK-~~R0 key holder identifier (PMK-R0 key holder identifier in the Authenticator)

R1KH ~~PMK-~~R1 key holder (PMK-R0 key holder in the Authenticator)

R1KH-ID ~~PMK-~~R1 key holder identifier (PMK-R0 key holder identifier in the Authenticator)

S0KH ~~PMK-R~~S0 key holder (PMK-R0 key holder in the Supplicant)

S0KH-ID ~~PMK-R~~S0 key holder identifier (PMK-R0 key holder identifier in the Supplicant)

S1KH ~~PMK-R~~S1 key holder (PMK-R1 key holder in the Supplicant)

S1KH-ID ~~PMK-R~~S1 key holder identifier (PMK-R1 key holder identifier in the Supplicant)

Change “FMS Request procedure” to “FMS request procedure” at 181.18.

Change “FMS Request” to “the FMS request procedure” at 373.61 (rightmost cell), 385.9 (rightmost cell).

Change “FMS” to “FMS request procedure” at 543.9 (rightmost cell), 544.41 (rightmost cell).

At 1593.30 change “a previous FMS Request” to “an existing FMS” (note no “Request”).

Change “FMS Request” to “FMS Request frame or Reassociation Request frame” at 2170.29, 2171.19/21/21.

Change “NON\_HT PPDU” to “non-HT PPDU” throughout (36 instances excluding xrefs, and except that at the following locations change to “non-high-throughput (non-HT) PPDU”: 200.51, 201.12, 201.48, 202.44, 202.58, the 5 locations in the TVHT\_MODE\_\* definitions already shown above).

Change “NON\_HT\_DUP PPDU” to “non-HT duplicate PPDU” in 22.3.8.2.4.

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 238 in <this document>, which address the issue raised by the commenter.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 550Michael MONTEMURRO9.4.1.11906.31 | The "Error" row in Table 9-51 looks as though it should be a "Reserved" row. | Change "Error" to "Reserved" at cited location. |

Discussion:

This is about:

**10.28.4 Response to an invalid Action and Action No Ack frame**

If a STA receives an individually addressed Action and Action No Ack frame with an unrecognized Category field or some other syntactic error and the MSB of the Category field equal to 0, then the STA shall return the Action and Action No Ack frame to the source without change except that the MSB of the Category field is set to 1.

Note, however, that a frame can’t be both an Action frame and an Action No Ack frame.

Proposed resolution:

REVISED

Put "10.28.4 (Response to an invalid Action and Action No Ack frame)" in the "See subclause" cell for that row.

In 10.28.4 change the first two “Action and Action No Ack frame”s to “Action or Action No Ack frame” and the last to “frame”.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 186Mark RISON122650.52 | It's not clear what to do in the handshakes if a verification fails. E.g. in 12.7.7.2 Group key handshake message 1 the e) might imply that you respond in any case. Should say ignore the received message if any verification step fails | Add an "Otherwise," at 2641.60, 2644.18, 2644.26. Also add some escape/otherwise words to the lettered list processes at the end of 12.7.6.4, 12.7.6.5, 12.7.7.2, 12.7.7.3 |
| CID 188Mark RISON12 | It's not sufficient to construct the message, you have to transmit it too! | Change "onstructs" to "onstructs and sends" at 2640.47, 2642.27. Change "creating" to "constructing" at 2650.52 |

Discussion:

Work in progress

As it says in the comment. It turns out there’s quite a lot to fix/clean up, including:

* The points at which the key replay counter (the one for non-request frames) is incremented on the Authenticator are not clear; in particular the value in the very first message is not clear
* There are two key replay counters, one for EAPOL-Key request frames, and one for other EAPOL-Key frames, but which is used is not specified
* Some of the key replay counter checks are vague
* The key replay checks on M4 are very odd
* The requirements for KDEs/subelements under OCVC are ambiguous
* It is not clear whether reserved fields are included in RSNE/RSNXE comparisons
* The use of a second RSNE is not clear
* Provision is not made for M3 being retransmitted
* It is not clear whether the 4WH can be used for something other than PTK generation
* An Authenticator is not required to ignore frames it is not expecting
* TBC: The same SNonce should be reused until a valid M3 has been received, to avoid DoS attacks
* TBC: An unencrypted M1 should be rejected once a valid M3 has been received, to avoid DoS attacks

For reference, here is what 12.7.2 says under d):

* **Key Replay Counter**. This field is represented as an unsigned integer, and is initialized to 0 when the PMK is established. The Supplicant shall use the key replay counter in the received EAPOL-Key frame when responding to an EAPOL-Key frame. It carries a sequence number that the protocol uses to detect replayed EAPOL-Key frames.

The Supplicant and Authenticator shall track the key replay counter per security association. The key replay counter shall be initialized to 0 on (re)association. The Authenticator shall increment the key replay counter on each successive EAPOL-Key frame.

When replying to a message from the Authenticator, the Supplicant shall use the Key Replay Counter field value from the last valid EAPOL-Key frames received from the Authenticator. The Authenticator should use the key replay counter to identify invalid messages to silently discard. The Supplicant should also use the key replay counter and ignore EAPOL-Key frames with a Key Replay Counter field value smaller than or equal to any received in a valid message. The local Key Replay Counter field should not be updated until after the EAPOL-Key MIC is checked and is found to be valid. In other words, the Supplicant never updates the Key Replay Counter field for message 1 in the 4‑way handshake, as it includes no MIC. This implies the Supplicant needs to allow for retransmission of message 1 when checking for the key replay counter of message 3.

The Supplicant shall maintain a separate key replay counter for sending EAPOL-Key request frames to the Authenticator; the Authenticator also shall maintain a separate replay counter to filter received EAPOL-Key request frames.

NOTE 1—The key replay counter does not play any role beyond a performance optimization in the 4-way handshake. In particular, replay protection is provided by selecting a never-before-used nonce value to incorporate into the PTK. It does, however, play a useful role in the group key handshake.

Proposed changes:

Make the following changes in 12.7.6/7:

* 4-way handshake
* General

RSNA defines a protocol using EAPOL-Key frames called the *4-way handshake*. The handshake completes the IEEE 802.1X authentication process. The information flow of the 4-way handshake is as follows:

Message 1: Authenticator  Supplicant: EAPOL-Key(0,0,1,0,P,0,0,ANonce,0,{} or {PMKID})

Message 2: Supplicant  Authenticator: EAPOL-Key(0,1,0,0,P,0,0,SNonce,MIC,{RSNE} or {RSNE, OCI KDE} or {RSNE, RSNXE} or {RSNE, OCI KDE, RSNXE})

Message 3: AuthenticatorSupplicant:
EAPOL-Key(1,1,1,1,P,0,KeyRSC,ANonce,MIC,{RSNE,GTK[N]} or
{RSNE, GTK[N], OCI KDE} or {RSNE, GTK[N], RSNXE} or
{RSNE, GTK[N], OCI KDE, RSNXE})

Message 4: Supplicant  Authenticator: EAPOL-Key(1,1,0,0,P,0,0,0,MIC,{}).

The FT initial mobility domain association uses the FT 4-way handshake to establish an initial PTKSA, GTKSA, if management frame protection is ~~enabled~~negotiated, an IGTKSA, and if beacon protection is enabled, a BIGTKSA, that is based on this protocol. The FT 4-way handshake protocol is described in 13.4 (FT initial mobility domain association).

The following apply:

* EAPOL-Key() denotes an EAPOL-Key frame conveying the specified argument list, using the notation introduced in 12.7.4 (EAPOL-Key frame notation).
* ANonce is a nonce that the Authenticator contributes for PTK generation. ANonce has the same value in message 1 and message 3.
* SNonce is a nonce from the Supplicant for PTK generation.
* P means the pairwise bit is set.
* The MIC is computed over the body of the EAPOL-Key frame (with the Key MIC field first zeroed before the computation) using the KCK defined in 12.7.1.3 (Pairwise key hierarchy) for PTK generation.
* RSNE represents the appropriate RSNEs.
* GTK[N] represents the GTK with its key identifier.
* OCI KDE contains the current operating channel information for the operating channel in which the EAPOL-Key frame is sent. OCI KDE is present when dot11RSNAOperatingChannelValidationActivated is true on the Supplicant in Message 2 and Authenticator in Message 3. Otherwise it is absent.
* RSNXE, when included in message 2, contains the RSNXE that the Supplicant sent in its (Re)Association Request frame, and when included in message 3, contains the RSNXE that the Authenticator sent in its Beacon or Probe Response frame. RSNXE is present in message 2 if this element is present in the (Re)Association Request frame that the Supplicant sent, and is present in  message 3 if this element is present in the Beacon or Probe Response frame that the Authenticator  sent.

NOTE—While the MIC calculation is the same in each direction, the Key Ack bit is different in each direction. It is set in EAPOL-Key frames from the Authenticator and 0 in EAPOL-Key frames from the Supplicant. 4‑way handshake requests from the Supplicant have the Request bit equal to 1. The Authenticator and Supplicant need to check these bits to stop reflection attacks. It is important that message 1 contents not be used to update state, in particular the keys in use, until the data are validated with message 3.

If dot11RSNAOperatingChannelValidationActivated is true and a channel switch is requested while the handshake is in progress, the handshake should be aborted.

* 4-way handshake message 1

Message 1 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type **=** N – see 12.7.2 (EAPOL-Key frames)

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA-1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0

Key Type = 1 (Pairwise)

Reserved = 0

Install = 0

Key Ack = 1

Key MIC = 0

Secure = 0

Error = 0

Request = 0

Encrypted Key Data = 0

Reserved = 0 – unused by this protocol version

Key Length = Cipher-suite dependent; see Table 12-7 (Cipher suite key lengths)

Key Replay Counter = key replay counter for EAPOL-Key frames that are not EAPOL-Key request frames (see 12.7.2 under d) = *n* ~~– to allow Authenticator or initiator STA to match the right message 2 from Supplicant or peer STA~~

Key Nonce = ANonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = 0

Key Data Length = length of Key Data field in octets

Key Data = PMKID for the PMK being used during PTK generation

~~Processing for PTK generation is as follows:~~

The Authenticator constructs and sends message 1 to the Supplicant at the end of a successful IEEE 802.1X authentication, after (re)association completes for a STA that has authenticated with SAE or PSK authentication is negotiated, when a cached PMKSA is used, or after a STA requests a new pairwise key.

***<insert para break>***

~~On reception of message 1, t~~The Supplicant silently discards message 1 if ~~determines whether the Key Replay Counter field value has been used before with the current PMKSA. If~~ the Key Replay Counter field ~~value~~ is less than or equal to the ~~current local value~~ key replay counter for EAPOL-Key frames that are not EAPOL-Key request frames (see 12.7.2 under d)~~, the Supplicant discards the message~~. ***<para break>***

Otherwise, the Supplicant:

* Generates a new nonce SNonce.
* Derives the PTK.
* Constructs (as described in 12.7.6.3) and sends message 2 to the Authenticator.
* 4-way handshake message 2

Message 2 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type **=** N – see 12.7.2 (EAPOL-Key frames)

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA-1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0 – same as message 1

Key Type = 1 (Pairwise) – same as message 1

Reserved = 0

Install = 0

Key Ack = 0

Key MIC = 0 when using an AEAD cipher or 1 otherwise

Secure = 0 – same as message 1

Error = 0 – same as message 1

Request = 0 – same as message 1

Encrypted Key Data = 1 when using an AEAD cipher or 0 otherwise

Reserved = 0 – unused by this protocol version

Key Length = 0

Key Replay Counter = *n* ~~– to let the Authenticator or initiator STA know to which message 1 this corresponds~~

Key Nonce = SNonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = Not present when using an AEAD cipher; otherwise, MIC(KCK, EAPOL) – MIC computed over the body of this EAPOL-Key frame with the Key MIC field first initialized to 0

Key Data Length = length of Key Data field in octets

* Key Data =
* included RSNE – the sending STA’s RSNE for PTK generation or peer RSNE for the current operating band, and when this message 2 is part of a fast BSS transition initial mobility domain association or an association started through the FT protocol, the PMKR1Name calculated by the S1KH according to the procedures of 12.7.1.6.4 (PMK-R1) is included in the PMKID List field of the RSNE and the FTE and MDE are also included, or
* The sending STA’s Multi-band element for PTK generation for a supported band other than the current operating band if dot11MultibandImplemented is true, or
* The sending STA’s RSNE and Multi-band element(s) for generating a single PTK for all involved bands, if dot11MultibandImplemented is true and both the Authenticator and the Supplicant use the same MAC address in the current operating band and the other supported band(s); or
* The sending STA’s RSNE and Multi-band element(s) for generating a different PTK for each involved band, if dot11MultibandImplemented is true and the Joint Multi-band RSNA subfield of the RSN capabilities field is 1 for both the Authenticator and the Supplicant, and either the Authenticator or the Supplicant uses different MAC addresses for different bands.
* Additionally, contains an OCI KDE when dot11RSNAOperatingChannelValidationActivated is true on the Supplicant.
* The RSNXE that the Supplicant sent in its (Re)Association Request frame, if this element is present in the (Re)Association Request frame that the Supplicant sent.

~~Processing for PTK generation is as follows:~~

~~The Supplicant sends message 2 to the Authenticator.~~

~~On reception of message 2, t~~The Authenticator ~~checks that~~ silently discards message 2 if the ~~key replay counter corresponds to the outstanding~~ Key Replay Counter field does not correspond to a message 1 for which it has not already received a message 2 with the same Key Replay Counter field, or if it has already sent a message 3 during the 4-way handshake. ~~If not, it silently discards the message.~~

~~I~~Otherwise, if dot11RSNAOperatingChannelValidationActivated is true and the Supplicant RSNE indicates OCVC capability, the Authenticator silently discards message 2 if any of the following are true:

* The OCI KDE or FTE OCI subelement is missing in the message
* ~~C~~The channel information in the OCI KDE does not match the current operating channel parameters (see 12.2.9 (Requirements for Operating Channel Validation))

Otherwise, the Authenticator~~:~~

* ~~Derives PTK.~~
* ***<delete “a)” and “b)” and the line breaks>*** ~~V~~verifies the message 2 MIC or AEAD decryption operation result. ***<join these two sentences, deleting “1)”>***
* If the calculated MIC does not match the MIC that the Supplicant included in the EAPOL-Key frame or the AEAD decryption operation returns failure, the Authenticator silently discards message 2.

Otherwise: ***<delete “2)”>***

* 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 match~~es~~ 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 all fields of the RSNE and, if present, the RSNXE, bitwise match~~es~~ th~~at~~e fields from the (Re)Association Request frame.
* ***<delete “i)”>*** If the~~se are not exactly the same~~y do not match bitwise, the Authenticator ~~uses MLME-DEAUTHENTICATE.request primitive to terminate the association~~ discards message 2 and deauthenticates.

Otherwise, the Authenticator:

i) Increments the key replay counter for EAPOL-Key frames that are not EAPOL-Key request frames (see 12.7.2 under d).

ii) Derives the PTK.

* ***<make this into “iii)”>*** ~~If they do match bitwise, the Authenticator c~~Constructs (as described in 12.7.6.4) and sends message 3 to the Supplicant.
* ***<make this into “iv)”>*** If management frame protection is being negotiated, ~~the AP~~ initializes the SA Query Transaction Identifier to an implementation-specific non-negative integer value, valid for the current pairwise security association.
* 4-way handshake message 3

Message 3 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type **=** N – see 12.7.2 (EAPOL-Key frames)

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA-1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0 – same as message 1

Key Type = 1 (Pairwise) – same as message 1

Reserved = 0

Install = 0/1 – For PTK generation, 0 only if the AP does not support key mapping keys, or if the STA has the No Pairwise bit (in the RSN Capabilities field) equal to 1and only the group key is used.

Key Ack = 1

Key MIC = 0 when using an AEAD cipher or 1 otherwise

Secure = 1 (keys installed)

Error = 0 – same as message 1

Request = 0 – same as message 1

Encrypted Key Data = 1

Reserved = 0 – unused by this protocol version

Key Length = Cipher-suite dependent; see Table 12-7 (Cipher suite key lengths)

Key Replay Counter = key replay counter for EAPOL-Key frames that are not EAPOL-Key request frames (see 12.7.2 under d),*~~n~~*~~+1~~ *m*, where *m* > *n*

Key Nonce = ANonce – same as message 1

EAPOL-Key IV = 0 (Version 2) or random (Version 1)

Key RSC = For PTK generation, starting TSC or PN that the Authenticator’s STA uses in MPDUs protected by GTK.

Key MIC = Not present when using an AEAD cipher; or otherwise, MIC(KCK, EAPOL) or MIC(SKCK, EAPOL) – MIC computed over the body of this EAPOL-Key frame with the Key MIC field first initialized to 0

Key Data Length = length of Key Data field in octets

Key Data =

* For PTK generation for the current operating band, the AP’s Beacon/Probe Response frame’s RSNE for the current operating band, and, optionally, a second RSNE that is the Authenticator’s pairwise cipher suite assignment for the current operating band, and, if a group cipher has been negotiated, the GTK and the GTK’s key identifier (see 12.7.2 (EAPOL-Key frames)) for the current operating band, and if management frame protection is negotiated, the IGTK KDE, and if beacon protection is enabled, the BIGTK KDE, and when this message 3 is part of a fast BSS transition initial mobility domain association or an association started through the FT protocol, the PMKR1Name calculated according to the procedures of 12.7.1.6.4 (PMK-R1) in the PMKID List field of the RSNE and the FTE with the same contents as in the (Re)Association Response frame, the MDE with the same contents as in the (Re)Association Response frame, the reassociation deadline timeout set to the minimum  of dot11FTReassociationDeadline and the key lifetime in the TIE[ReassociationDeadline], and the PTK lifetime in the TIE[KeyLifetime]; or
* For PTK generation for a supported band other than the current operating band, the Authenticator’s Beacon/DMG Beacon/Announce/Probe Response/Information Response frame’s Multi-band element associated with the supported band, and optionally a second Multi-band element that indicates the Authenticator’s pairwise cipher suite assignment for the supported band, and, if group cipher for the supported band is negotiated, the Multi-band GTK KDE for the supported band if dot11MultibandImplemented is true, or
* For generating a single PTK for all involved bands, the Authenticator’s Beacon/DMG Beacon/Announce/Probe Response/Information Response frame’s RSNE and Multi-band element(s), and optionally, additional RSNE and Multi-band element(s) that indicate the Authenticator’s assignment of one pairwise cipher suite for all involved bands; if a group cipher for all involved bands is negotiated, the GTK and the GTK’s key identifier for all involved bands, if dot11MultibandImplemented is true and both the Authenticator and the Supplicant use the same MAC address in the current operating band and the other supported band(s), or
* For generating different PTKs for the current operating band and other supported band(s), the Authenticator’s Beacon/DMG Beacon/Announce/Probe Response/Information Response frame’s RSNE and Multi-band element(s), and optionally, additional RSNE and Multi-band elements that are the Authenticator’s pairwise cipher suite assignments for one or more involved bands; if group ciphers for the involved bands are negotiated, the Multi-band GTK KDEs for the involved bands, if dot11MultibandImplemented is true and the Joint Multi-band RSNA subfield is 1 for both the Authenticator and Supplicant, and either the Authenticator or the Supplicant uses different MAC addresses for different bands.
* Additionally, contains an OCI KDE when dot11RSNAOperatingChannelValidationActivated is true on the Authenticator.
* The RSNXE that the Authenticator sent in its Beacon or Probe Response frame, if this element is present in the Beacon or Probe Response frame that the Authenticator sent.

~~Processing for PTK generation is as follows:~~

If the Extended Key ID for Individually Addressed Frames subfield of the RSN Capabilities field is 1 for both the Authenticator and the Supplicant, then the Authenticator assigns a new Key ID for the PTKSA in the range of 0 to 1 that is different from the Key ID assigned in the previous handshake and uses the MLME-SETKEYS.request primitive to install the new key to receive individually addressed MPDUs protected by the PTK with the assigned Key ID. Otherwise Key ID 0 is used and installation of the key is deferred until after message 4 has been received. ~~The Authenticator sends message 3 to the Supplicant.~~

NOTE—If an existing PTK is still in effect, the Authenticator ~~IEEE 802.11~~ MAC continues to transmit protected, individually addressed MPDUs (if any) using the existing key. With the installation of the new key for receive, the Authenticator is able to receive protected, individually addressed MPDUs using either the old key (if present) or the new key.

~~On reception of message 3, t~~The Supplicant silently discards ~~the~~ message 3 if the Key Replay Counter field ~~value has already been used~~ is less than or equal to the key replay counter for EAPOL-Key frames that are not EAPOL-Key request frames (see 12.7.2 under d), or if the ANonce value in message 3 differs from the ANonce value in message 1.

~~I~~Otherwise, if dot11RSNAOperatingChannelValidationActivated is true and the Authenticator RSNE indicates OCVC capability, the Supplicant silently discards message 3 if any of the following are true:

* The OCI KDE or FTE OCI subelement is missing in the message
* ~~C~~The channel information in the OCI KDE does not match the current operating channel parameters (see 12.2.9 (Requirements for Operating Channel Validation))

~~T~~Otherwise, the Supplicant ~~also~~:

* Verifies the RSNE and, if present, the RSNXE. If this message 3 is part of a fast BSS transition initial mobility domain association or an association started through the FT protocol, the Supplicant verifies that the PMKR1Name in the PMKID List field of the RSNE is identical to the value it sent in message 2 and verifies that all other fields of the RSNE are identical to the fields in the RSNE present in the Beacon or Probe Response frames and verifies that the FTE and MDE are the same as in the (Re)Association Response frame. Otherwise, the Supplicant verifies that the RSNE is identical to that the STA received in the Beacon or Probe Response frame. If the RSNXE is present, the Supplicant verifies that the RSNXE is identical to that the STA received in the Beacon or Probe Response frame. ***<insert para break>***

If any of these verification steps indicates a mismatch, the ~~STA~~Supplicant discards message 3 and ~~shall~~ disassociates or deauthenticates. If a second RSNE is provided in the message, the Supplicant uses the pairwise cipher suite specified in the second RSNE or deauthenticates.

* ***<delete the “b)”>*** ~~V~~Otherwise, the Supplicant verifies the message 3 MIC or AEAD decryption operation result. If the calculated MIC does not match the MIC that the Authenticator included in the EAPOL-Key frame or AEAD decryption operation returns failure, the Supplicant silently discards message 3.

Otherwise, the Supplicant: ***<make all the following subbullets, starting with “i)”>***

* ~~Updates the last-seen~~Sets the key replay counter for EAPOL-Key frames that are not EAPOL-Key request frames (see 12.7.2 under d) to the value of the Key Replay Counter field.
* If the Extended Key ID for Individually Addressed Frames subfield of the RSN Capabilities field is 1 for both the Authenticator and Supplicant~~: U~~, uses the MLME-SETKEYS.request primitive to configure the ~~IEEE 802.11~~ MAC to receive individually addressed MPDUs protected by the PTK with the assigned Key ID.
* Constructs (as described in 12.7.6.5) and sends message 4~~.~~ ***<delete line break and “f)”>***
* ~~Sends message 4~~ to the Authenticator.

NOTE—When the 4-way handshake is first used, message 4 is sent in the clear.

* Uses the MLME-SETKEYS.request primitive to configure the ~~IEEE 802.11~~ MAC to send and~~, if the receive key has not yet been installed,~~ to receive individually addressed MPDUs protected by the PTK. The GTK, IGTK if management frame protection is negotiated and BIGTK if beacon protection is enabled are ~~is~~ also configured ~~by~~using the MLME-SETKEYS.request primitive.
* 4-way handshake message 4

Message 4 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type **=** N – see 12.7.2 (EAPOL-Key frames)

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA-1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0 – same as message 1

Key Type = 1 (Pairwise) – same as message 1

Reserved = 0

Install = 0

Key Ack = 0 – this is the last message

Key MIC = 0 when using an AEAD cipher or 1 otherwise

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 1 when using an AEAD cipher or 0 otherwise

Reserved = 0 – unused by this protocol version

Key Length = 0

Key Replay Counter = *~~n+1~~m*

Key Nonce = 0

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = Not present when using an AEAD cipher; or otherwise, MIC(KCK, EAPOL) – MIC computed over the body of this EAPOL-Key frame with the Key MIC field first initialized to 0

Key Data Length = length of Key Data field in octets

Key Data = none required

~~Processing for PTK generation is as follows:~~

~~The Supplicant sends message 4 to the Authenticator. Note that when the 4-way handshake is first used, message 4 is sent in the clear.~~

~~On reception of message 4, t~~The Authenticator ~~verifies that~~ silently discards message 4 if the Key Replay Counter field ~~value~~ is ~~one that it used on this 4-way handshake and is strictly larger than that in any other EAPOL-Key frame that has the Request bit in the Key Information field set to 0 and that has been received during this session; if it is not, it silently discards the message~~ not equal to that in message 3. ***<insert para break>***

OR: The Authenticator silently discards message 4 if the Key Replay Counter field does not correspond to a message 3 for which it has not already received a message 4 with the same Key Replay Counter field.

OR: ~~On reception of message 4, t~~The Authenticator ~~verifies that~~ silently discards message 4 if the 4-way handshake has already completed or if the Key Replay Counter field does not correspond to a message 3 ~~value is one that it used~~ on this 4-way handshake ~~and is strictly larger than that in any other EAPOL-Key frame that has the Request bit in the Key Information field set to 0 and that has been received during this session; if it is not, it silently discards the message~~. ***<insert para break>***

Otherwise~~:~~, t***<delete the “a)” and line break>***

* ~~T~~he Authenticator ~~checks~~verifies the message 4 MIC or AEAD decryption operation result. If the calculated MIC does not match the MIC that the Supplicant included in the EAPOL-Key frame or the AEAD decryption operation returns failure, the Authenticator silently discards message 4.
1. ~~If the MIC is valid, the Authenticator~~ ***<delete the “b)”>***Otherwise, the Authenticator: ***<make all the following subbullets, starting with “i)”>***
2. ~~u~~Uses the MLME-SETKEYS.request primitive to configure the ~~IEEE 802.11~~ MAC to send and, if the receive key has not yet been installed, to receive ~~protected,~~ individually addressed MPDUs ~~using for the new~~ protected by the PTK.
* ~~The Authenticator updates the Key Replay Counter field so that it uses a fresh value~~ Increments the key replay counter for EAPOL-Key frames that are not EAPOL-Key request frames (see 12.7.2 under d), for use if a rekey becomes necessary.
* 4-way handshake implementation considerations

If the Authenticator does not receive a reply to its messages, it shall attempt dot11RSNAConfigPairwiseUpdateCount transmits of the message, plus a final timeout. The retransmit timeout value shall be 100 ms for the first timeout, half the listen interval for the second timeout, and the listen interval for subsequent timeouts. If there is no listen interval or the listen interval is zero, then 100 ms shall be used for all timeout values. If it still has not received a response after these retries, then for PTK generation the Authenticator should deauthenticate ~~the STA~~. The Authenticator increments the key replay counter for EAPOL-Key frames that are not EAPOL-Key request frames (see 12.7.2 under d) prior to each retransmission.

For PTK generation, if the Supplicant does not receive message 1 within the expected time interval (prior to IEEE 802.1X timeout), it should ~~disassociate,~~ deauthenticate~~,~~ and try another AP or peer STA.

The Authenticator should ignore EAPOL-Key frames it is not expecting in reply to messages it has sent or EAPOL-Key frames in which the Key Ack bit is 1. This stops an attacker from sending the first message to the Supplicant who responds to the Authenticator.

An implementation should save the KCK and KEK beyond the 4-way handshake, as they are needed for group key handshakes, and recovery from TKIP MIC failures.

~~The Supplicant uses the MLME-SETKEYS.request primitive to configure the temporal key from 12.7.1 (Key hierarchy) into its STA after sending message 4 to the Authenticator.~~

If the RSNE check for message 2 or message 3 fails, the SME should log an error and deauthenticate ~~the peer~~.

* Sample 4-way handshake

The following is a sample of a 4-way handshake for illustration.

After IEEE 802.1X authentication completes by the AP sending an EAP-Success, the AP initiates the 4-way handshake. See Figure 12-48 (Sample 4-way handshake).****

The 4-way handshake consists of the following steps:

* The Authenticator sends an EAPOL-Key frame containing an ANonce.
1. The Supplicant derives ~~a~~the PTK from ANonce and SNonce.
* The Supplicant sends an EAPOL-Key frame containing SNonce, the RSNE from the (Re)Association Request frame, and a MIC.
* The Authenticator derives the PTK from ANonce and SNonce and validates the MIC in the EAPOL-Key frame.
* The Authenticator sends an EAPOL-Key frame containing ANonce, the RSNE from its Beacon or Probe Response frames, the MIC, the GTK, an indication of whether to install the temporal keys, and if management frame protection is negotiated, the IGTK, and if beacon protection is enabled, the BIGTK.
* The Supplicant sends an EAPOL-Key frame to confirm whether the temporal keys were installed.
* 4-way handshake analysis

The following is an analysis of the 4-way handshake.

This subclause makes the trust assumptions used in this protocol explicit. The protocol assumes the -following:

* The PMK is known only by the Supplicant’s STA and the Authenticator’s STA.
* The Supplicant’s STA uses IEEE 802 address SPA.
* The Authenticator’s STA uses IEEE 802 address AA.

In many instantiations the RSNA architecture immediately breaks the first assumption because the IEEE 802.1X AS also knows the PMK. Therefore, additional assumptions are required:

* The AS does not expose the PMK to other parties.
* The AS does not masquerade as the Supplicant to the Authenticator.
* The AS does not masquerade as the Authenticator to the Supplicant.
* The AS does not masquerade as the Supplicant’s STA.
* The AS does not masquerade as the Authenticator’s STA.

The protocol also assumes this particular Supplicant-Authenticator pair is authorized to know this PMK and to use it in the 4-way handshake. If any of these assumptions are broken, then the protocol fails to provide any security guarantees.

The protocol also assumes that the AS delivers the correct PMK to the AP with IEEE 802 address AA and that the STA with IEEE 802 address SPA hosts the Supplicant that negotiated the PMK with the AS. None of the protocols defined by this standard and IEEE Std 802.1X-2010 permit the AS, the Authenticator, the Supplicant, or either STA to verify these assumptions.

The PTK derivation step

PTK PRF-Length(PMK, “Pairwise key expansion”, Min(AA,SPA) || Max(AA,SPA) || Min(ANonce,SNonce) || Max(ANonce,SNonce))

performs a number of functions:

* Including the AA and SPA in the computation
* Binds the PTK to the communicating STAs and
* Prevents undetected man-in-the-middle attacks against 4-way handshake messages between the STAs with these two IEEE 802 addresses.
* If ANonce is randomly selected, including ANonce
* Guarantees the STA at IEEE 802 address AA that PTK is fresh,
* Guarantees that message 2 and message 4 are live, and
* Uniquely identifies PTK as <AA, ANonce>.
* If SNonce is randomly selected, including SNonce
* Guarantees the STA at IEEE 802 address SPA that PTK is fresh,
* Guarantees that message 3 is live, and
* Uniquely identifies PTK as <SPA, SNonce>.

Choosing the nonces randomly helps prevent precomputation attacks. With unpredictable nonces, a man-in-the-middle attack that uses the Supplicant to precompute messages to attack the Authenticator cannot progress beyond message 2, and a similar attack against the Supplicant cannot progress beyond message 3. The protocol might execute further before an error if predictable nonces are used.

Message 1 delivers ANonce to the Supplicant and initiates negotiation for a new PTK. It identifies AA as the peer STA to the Supplicant’s STA. If an adversary modifies either of the addresses or ANonce, the Authenticator detects the result when validating the MIC in message 2. Message 1 does not carry a MIC, as it is impossible for the Supplicant to distinguish this message from a replay without maintaining state of all security associations through all time (PMK might be a static key).

Message 2 delivers SNonce to the Authenticator so it can derive the PTK. If the Authenticator selected ANonce randomly, message 2 also demonstrates to the Authenticator that the Supplicant is live, that the PTK is fresh, and that there is no man-in-the-middle attack, as the PTK includes the IEEE 802 MAC addresses of both. Inclusion of ANonce in the PTK derivation also protects against replay. The MIC prevents undetected modification of message 2 contents.

Message 3 confirms to the Supplicant that there is no man-in-the-middle attack. If the Supplicant selected SNonce randomly, it also demonstrates that the PTK is fresh and that the Authenticator is live. The MIC again prevents undetected modification of message 3.

While message 4 serves no cryptographic purpose, it serves as an acknowledgment to message 3. It is required to inform the Authenticator that the Supplicant has installed the PTK and GTK and hence can receive encrypted frames.

The PTK and GTK are installed by using MLME-SETKEYS.request primitive after message 4 is sent. The PTK is installed before the GTK.

Then the 4-way handshake uses a correct, but unusual, mechanism to guard against replay. As noted earlier in this subclause, ANonce provides replay protection to the Authenticator, and SNonce to the Supplicant. In most session initiation protocols, replay protection is accomplished explicitly by selecting a nonce randomly and requiring the peer to reflect the received nonce in a response message. The 4-way handshake instead mixes ANonce and SNonce into the PTK, and replays are detected implicitly by MIC failures. In particular, the Key Replay Counter field serves no cryptographic purpose in the 4-way handshake. Its presence is not detrimental, however, and it plays a useful role as a minor performance optimization for processing stale instances of message 2. This replay mechanism is correct, but its implicit nature makes the protocol harder to understand than an explicit approach.

It is critical to the correctness of the 4-way handshake that at least one bit differs in each message. Within the 4‑way handshake, message 1 can be recognized as the only one in which the Key MIC bit is 0, meaning message 1 does not include the MIC, while message 2 to message 4 do. Message 3 differs from message 2 by not asserting the Ack bit and from message 4 by asserting the Ack Bit. Message 2 differs from message 4 by including the RSNE.

Request messages are distinct from 4-way handshake messages because the former asserts the Request bit and 4-way handshake messages do not. Group key handshake messages are distinct from 4-way handshake messages because they assert a different key type.

* Group key handshake
* General

The Authenticator uses the Group key handshake to send a new GTK and, if management frame protection is negotiated, a new IGTK, and if beacon protection is enabled, a new BIGTK, to the Supplicant.

The Authenticator may initiate the exchange when a Supplicant is disassociated or deauthenticated.

Message 1: Authenticator  Supplicant:

EAPOL-Key(1,1,1,0,G,0,Key RSC,0, MIC, {GTK[N], IGTK[M], BIGTK[Q]})

Message 2: Supplicant  Authenticator: EAPOL-Key(1,1,0,0,G,0,0,0,MIC,{})

The following apply:

* Key RSC denotes the last TSC or PN sent using the GTK.
* GTK[N] denotes the GTK with its key identifier as defined in 12.7.2 (EAPOL-Key frames) using the KEK defined in 12.7.1.3 (Pairwise key hierarchy) and associated IV.
* IGTK[M], when present, denotes the IGTK with its key identifier as defined in 12.7.2 (EAPOL-Key frames) using the KEK defined in 12.7.1.3 (Pairwise key hierarchy) and associated IV.
* BIGTK[Q], when present, denotes the BIGTK with its key identifier as defined in 12.7.2 (EAPOL-Key frames) using the KEK defined in 12.7.1.3 (Pairwise key hierarchy) and associated IV.
* The MIC is computed over the body of the EAPOL-Key frame (with the MIC field zeroed for the computation) using the KCK defined in 12.7.1.3 (Pairwise key hierarchy).
* OCI KDE represents the current operating channel information using which the EAPOL-Key frame is sent. OCI KDE is included when dot11RSNAOperatingChannelValidationActivated is true on the STA sending the message.

The Supplicant may trigger a group key handshake by sending an EAPOL-Key frame with the Request bit set to 1 and the type of the Group Key bit.

An Authenticator shall do a 4-way handshake before a group key handshake if both are required to be done.

NOTE—The Authenticator cannot initiate the group key handshake until the 4-way handshake completes successfully.

If dot11RSNAOperatingChannelValidationActivated is true and a channel switch is requested while the handshake is in progress, the handshake should be aborted.

* Group key handshake message 1

Message 1 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type **=** N – see 12.7.2 (EAPOL-Key frames)

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA-1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0

Key Type = 0 (Group)

Install = 0

Key Ack = 1

Key MIC = 0 when using an AEAD cipher or 1 otherwise

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 1

Reserved = 0

Key Length = 0

Key Replay Counter = key replay counter for EAPOL-Key frames that are not EAPOL-Key request frames (see 12.7.2 under d) = *n*~~+2~~

Key Nonce = 0

EAPOL-Key IV = 0 (Version 2) or random (Version 1)

Key RSC = last TSC or PN for the GTK

Key MIC = Not present when using an AEAD cipher; otherwise, MIC(KCK, EAPOL)

Key Data Length = length of Key Data field in octets

Key Data = encrypted, encapsulated

* GTK and the GTK’s key identifier (see 12.7.2 (EAPOL-Key frames))
* When present, IGTK, IGTK’s key identifier, and IPN (see 12.7.2 (EAPOL-Key frames))
* When present, BIGTK, BIGTK’s key identifier, and BIPN (see 12.7.2 (EAPOL-Key frames))
* OCI KDE when dot11RSNAOperatingChannelValidationActivated is true on the Authenticator

The Authenticator constructs and sends message 1 to the Supplicant.

~~On reception of message 1, t~~The Supplicant silently discards message 1 if~~:~~ ***<delete the line break and “a)”>***

* ~~Verifies that~~ the Key Replay Counter field ~~value has not yet been seen before, i.e., its value is strictly larger than that in any other EAPOL-Key frame received thus far during this session~~ is less than or equal to the key replay counter for EAPOL-Key frames that are not EAPOL-Key request frames (see 12.7.2 under d).
1. ***<delete the “e)”>*** ~~I~~Otherwise, if dot11RSNAOperatingChannelValidationActivated is true and the Authenticator RSNE indicates OCVC capability, the Supplicant silently discards message 1 if any of the following are true:
* The OCI KDE is missing in the message
* ~~C~~The channel information in the OCI KDE does not match the current operating channel parameters (see 12.2.9 (Requirements for Operating Channel Validation))
* ***<delete the “c)”>*** ~~V~~Otherwise, the Supplicant verifies ~~that~~ the message 1 MIC ~~is valid~~ or AEAD decryption operation result~~, i.e., it uses the KCK that is part of the PTK to verify that there is no data integrity error, or that the AEAD decryption steps succeed~~. If the calculated MIC does not match the MIC that the Authenticator included in the EAPOL-Key frame or the AEAD decryption operation returns failure, the Supplicant silently discards message 1.

Otherwise, the Supplicant: ***<make all the following subbullets, starting with “i)”>***

* Sets the key replay counter for EAPOL-Key frames that are not EAPOL-Key request frames (see 12.7.2 under d) to the value of the Key Replay Counter field.
* Uses the MLME-SETKEYS.request primitive to configure the ~~temporal~~ GTK ~~and~~, ~~the~~ IGTK ~~when present~~ if management frame protection is negotiated, and ~~the~~ BIGTK if beacon protection is enabled~~, into the MAC~~.
* ~~Responds by creating and sending~~Constructs (as described in 12.7.7.3) and sends message 2 ~~of the group key handshake~~ to the Authenticator ~~and incrementing the replay counter~~.

NOTE—The Authenticator increments and uses a new Key Replay Counter field value on every message 1 instance, even retries, because the message 2 responding to an earlier message 1 might have been lost. If the Authenticator did not increment the replay counter, the Supplicant discards the retry, and no responding message 2 ever arrives.

* Group key handshake message 2

Message 2 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type **=** N – see 12.7.2 (EAPOL-Key frames)

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA-1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0 – same as message 1

Key Type = 0 (Group) – same as message 1

Install = 0

Key Ack = 0

Key MIC = 0 when using an AEAD cipher or 1 otherwise

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 1 when using an AEAD cipher or 0 otherwise

Reserved = 0

Key Length = 0

Key Replay Counter = *n*~~+2~~ *–* same as message 1

Key Nonce = 0

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = Not present when using an AEAD cipher; otherwise, MIC(KCK, EAPOL)

Key Data Length = length of Key Data field in octets

Key Data = OCI KDE when dot11RSNAOperatingChannelValidationActivated on the Authenticator

The Authenticator silently discards message 2 if the Key Replay Counter field is not equal to that in message 1.

~~On reception of message 2, the Authenticator:~~

* ~~Verifies that the Key Replay Counter field value matches one it has used in the group key handshake.~~
1. ***<delete the “f)”>*** ~~I~~Otherwise, if dot11RSNAOperatingChannelValidationActivated is true and the Supplicant RSNE indicates OCVC capability, the Authenticator silently discards message 2 if any of the following are true:
* The OCI KDE is missing in the message
* ~~C~~The channel information in the OCI KDE does not match the current operating channel parameters (see 12.2.9 (Requirements for Operating Channel Validation))
* ~~Verifies that the MIC is valid, i.e., it uses the KCK that is part of the PTK to verify that there is no data integrity error, or that the AEAD decryption steps succeed.~~

Otherwise, the Authenticator verifies the message 2 MIC or AEAD decryption operation result. If the calculated MIC does not match the MIC that the Supplicant included in the EAPOL-Key frame or the AEAD decryption operation returns failure, the Authenticator silently discards message 2.

Otherwise, the Authenticator increments the key replay counter for EAPOL-Key frames that are not EAPOL-Key request frames (see 12.7.2 under d).

* Group key handshake implementation considerations

If the Authenticator does not receive a reply to its messages, it~~s~~ shall attempt dot11RSNAConfigGroupUpdateCount transmits of the message, plus a final timeout. The retransmit timeout value shall be 100 ms for the first timeout, half the listen interval for the second timeout, and the listen interval for subsequent timeouts. If there is no listen interval or the listen interval is zero, then 100 ms shall be used for all timeout values. If it still has not received a response after th~~is~~ese retries, then the Authenticator’s STA should ~~use the MLME-DEAUTHENTICATE.request primitive to~~ deauthenticate ~~the STA~~. The Authenticator increments the key replay counter for EAPOL-Key frames that are not EAPOL-Key request frames (see 12.7.2 under d) prior to each retransmission.

To prevent key reinstallation attacks, the Supplicant shall maintain a copy of the most recent GTK, most recent IGTK, and most recent BIGTK, installed as a result of receipt of EAPOL-Key frames. The Supplicant shall not install a GTK, an IGTK or a BIGTK when the key to be set matches ~~either~~any of these ~~two~~ keys (see 6.3.19 (SetKeys)).

* Sample group key handshake

The following is a sample of a group key handshake for illustration.

The state machines in 12.7.9 (RSNA Supplicant key management state machine) and 12.7.10 (RSNA Authenticator key management state machine) change the GTK and, when present, IGTK and/or BIGTK in use by the network. See Figure 12-49 (Sample group key handshake).

The following steps occur:

* The Authenticator generates a new GTK, a new IGTK ~~and~~ when management frame protection ~~has been~~is negotiated, and a new BIGTK when beacon protection is enabled. It encapsulates the GTK~~, as necessary~~, the IGTK when management frame protection is negotiated, and the BIGTK when beacon protection is enabled. It sends an EAPOL-Key frame containing the GTK, IGTK when management frame protection is negotiated and BIGTK when beacon protection is enabled (message 1), along with the last TSC or PN used with the GTK (RSC), the last IPN used with the IGTK when management frame protection is negotiated and the last BIPN used with the BIGTK when beacon protection is enabled.
1. On receiving the EAPOL-Key frame, the Supplicant validates the MIC, decapsulates the GTK, the IGTK when ~~present~~ management frame protection is negotiated and the BIGTK when ~~present~~ beacon protection is enabled, and uses the MLME-SETKEYS.request primitive to configure the GTK, ~~PN,~~ IGTK when management frame protection is negotiated~~, RSC, IPN~~, and BIGTK when beacon protection is enabled ~~and BIPN in its STA~~.
2. The Supplicant then constructs and sends an EAPOL-Key frame in acknowledgment to the Authenticator.
* On receiving the EAPOL-Key frame, the Authenticator validates the MIC. If the GTK, the IGTK~~,~~ if ~~present~~ management frame protection is negotiated, and the BIGTK~~,~~ if ~~present~~beacon protection is enabled, have not yet been configured ~~using the MLME-SETKEYS.request primitive~~, after the Authenticator has delivered the GTK, IGTK if management frame protection is negotiated and BIGTK if beacon protection is enabled to all associated STAs, it uses the MLME-SETKEYS.request primitive to configure them.

At 2767.52 change “temporal MGTK” to “MGTK”.

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 186 and CID 188 in <this document>, which address the issue raised by the commenter.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 224Mark RISON | There are no constraints on the cipher to use with TDLS, other than not using WEP or TKIP. Some recommendations should be given, specifically that it should be at least as strong as the cipher used with the AP | After the sentence starting "The pairwise cipher suite list field indicating " in 12.7.8.4.2 TPK handshake message 1 add "A pairwise cipher suite of key size smaller than that used on the connection between the STA and the AP should not be used." and after the sentence starting "If none of the pairwise cipher suites are acceptable" add "The TDLS responder STA should ignore any pairwise cipher suites of key size smaller than that used on the connection between the STA and the AP, and should reject the TDLS Setup Request framewith status code STATUS\_INVALID\_PAIRWISE\_CIPHER if all the pairwise cipher suites are such.". In 12.7.8.4.3 TPK handshake message 2 after the sentence starting "Include a pairwise cipher suite" add "A pairwise cipher suite of key size smaller than that used on the connection between the STA and the AP should not be used." |

Discussion:

It is odd that security is required if security is used on the link to the AP, i.e. the TDLS link is expected to be secure, but there are no requirements, recommendations or even suggestions on the strength of this security (apart from the bottom-of-the-barrel prohibition on WEP and TKIP), i.e. the TDLS link can be less secure than the link to the AP.

In practice, if a given pairwise cipher suite is suitable for the link to the AP, it would typically be suitable for the TDLS link too. This should at least be hinted at.

For reference, the wording (in 12.7.8.1) is:

If any security method (pre-RSNA or RSNA) is enabled on the connection between a STA and the AP, the STA shall require that the TPK security protocol complete successfully before using a direct link. If no security method is enabled on the connection between a STA and the AP, the STA shall not use the TPK security protocol on the direct link. A STA may refuse to set up a TDLS link when the protection on the STA link to the AP is secured with a weak algorithm or when the link between the STA and the AP is not using any security.

Proposed resolution:

REVISED

In 12.7.8.4.2 TPK handshake message 1, after the para starting "The pairwise cipher suite list field indicating" add

“NOTE—The TDLS initiator STA might indicate the same pairwise cipher suite as used on the connection between the STA and the AP.”

and after the sentence starting "If none of the pairwise cipher suites are acceptable" add

“NOTE—The TDLS responder STA might only accept the same pairwise cipher suite as used on the connection between the STA and the AP.”

In 12.7.8.4.3 TPK handshake message 2 after the sentence starting "Include a pairwise cipher suite" add

“NOTE—The TDLS responder STA might select the same pairwise cipher suite as used on the connection between the STA and the AP.”

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 469Mark RISON6.3.3.3.2327.17 | Information from Reduced Neighbor Report elements is missing in the BSSDescriptionSet | As it says in the comment |

Discussion:

An RNR can be present in Beacon, Probe Response and FILS Discovery frames.

For FILS Discovery frames, it is provided in the BSSDescriptionFromFD:



However, for beacons and probe responses it is not provided in the BSSDescription.

Proposed resolution:

REVISED

In 6.3.3.3.2 Semantics of the service primitive copy the Reduced Neighbor Report row from the BSSDescriptionFromFD table to the end of the BSSDescription table, changing “FILS Discovery” to “Beacon or Probe Response”.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 422Mark RISON | There are ~10 references to RA and TA "values" but they're not values, they're fields. Dito 3x SA/DA "values" | Change "value" to "field" in those instances |

Discussion:

RA, TA, SA and DA are fields or addresses, not values, and if they’re addresses they don’t contain addresses, as this would cause a recursive crash. Fields do have values.

Proposed changes:

**4.3.28.1 General**

As described in 4.3.28.2 (Selective reception of group addressed frames), when a GLK AP transmits a Data frame with a four-address MAC header whose RA field contains a group address, the ~~contents of the~~ RA is a synthetic receiver address (SYNRA), and therefore its RA and DA ~~values~~fields are not equal. A GLK non-AP STA supports selective reception of group addressed frames by supporting SYNRA reception.

**9.3.2.1.2 Address and BSSID fields**

When a Data frame carries an MSDU (or fragment thereof), the DA and SA ~~values~~ related to that MSDU are carried in the Address 1, Address 2, Address 3, and Address 4 fields (according to the setting of the To DS and From DS subfields) as defined in Table 9-30 (Address field contents).

When a Data frame carries a basic A-MSDU, the DA and SA ~~values~~ related to each MSDU carried by the A-MSDU are carried within the A-MSDU subframe header. Zero, one, or both of these fields are present in the Address 1 and Address 2 fields as indicated in Table 9-30 (Address field contents).

**9.3.2.1.5 Duration field**

The Duration field calculation for the Data frame is based on the rules in 10.6 (Multirate support) that determine the data rate at which the Control frames in the frame exchange sequence are transmitted. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer. All STAs process Duration field values less than or equal to 32 767 from valid Data frames (without regard for the RA, DA, and/or BSSID ~~address~~field values that might be present in these frames) to update their NAV settings as appropriate under the coordination function rules.

**9.3.2.2.2 Basic A-MSDU subframe format**

An A-MSDU contains only MSDUs whose DA and SA parameter values map to the same receiver address

(RA) and transmitter address (TA) field values. The rules for determining RA and TA are independent of whether

the frame body carries an A-MSDU.

**10.3.2.3.2 RIFS**

RIFS shall not be used between frames with different RA field values.

**10.3.6 Group addressed MPDU transfer procedure**

A STA that is not an S1G relay STA shall discard an MPDU with a group address in the Address 1 field unless one of the following cases applies: (1) the value in the Address 1 field matches any value in the dot11GroupAddressesTable or ~~matches~~is the broadcast address ~~value~~, or (2) the STA is a GLK STA and the address in the Address 1 field is a SYNRA.

**10.11 A-MSDU operation**

An A-MSDU contains only MSDUs whose DA parameter values map to a single RA field value (see 9.3.2.2 (Aggregate MSDU (A-MSDU) format)). An A-MSDU contains only MSDUs whose SA parameter values map to a single TA field value (see 9.3.2.2 (Aggregate MSDU (A-MSDU) format)).

**10.23.2.4 Obtaining an EDCA TXOP**

A STA shall save the TXOP holder address for the BSS in which it is associated, which is the MAC address from the Address 2 field of the frame that initiated a frame exchange sequence except when this is a CTS frame, in which case the TXOP holder address is the Address 1 field. If the TXOP holder address is obtained from a Control frame, a VHT STA shall save the nonbandwidth signaling TA ~~value~~ obtained from the Address 2 field. If a non-VHT STA receives an RTS frame with the RA matching the MAC address of the STA and the MAC address in the TA field in the RTS frame matches the saved TXOP holder address, then the STA shall send the CTS frame after SIFS, without regard for, and without resetting, its NAV. If a VHT STA receives an RTS frame with the RA matching the MAC address of the STA and the nonbandwidth signaling TA ~~value~~ obtained from the Address 2 field in the RTS frame matches the saved TXOP holder address, then the STA shall send the CTS frame after SIFS, without regard for, and without resetting, its NAV. If a CMMG STA receives an RTS frame with the RA matching the MAC address of the STA and the TA ~~value~~ obtained from the Address 2 field in the RTS frame matches the saved TXOP holder address, then the STA shall send the CTS frame after SIFS, without regarding for, and without resetting its NAV.

**10.28.3 Duration/ID field processing**

When the contents of a received Duration/ID field, treated as an unsigned integer and without regard for address field values, type, and subtype

**10.39.10 Updating multiple NAVs**

If a STA receives a valid CF-End frame response with RA and TA field values that match the NAVSRC and NAVDST values, in any order, for any NAV, then the STA shall set the associated NAV to the value of the Duration field in the received CF-End frame. If one of NAVSRC or NAVDST of a NAV is 0 and the corresponding NAVDST or NAVSRC, respectively, of the NAV matches the RA or the TA field value of the received valid CF-End frame, then the STA shall set the associated NAV to the value of the Duration field in the received CF-End frame.

If one of NAVSRC or NAVDST of a NAV is 0 and the nonzero NAVDST or NAVSRC of the NAV match either the RA or the TA field value of a received valid frame, the NAVSRC or NAVDST that is 0 shall be set to the RA or TA that does not match the nonzero NAVSRC or NAVDST.

**11.15.12 Switching between 40 MHz and 20 MHz**

A VHT STA is not required to perform any of the behavior described in this subclause associated with Information Request and 20 MHz BSS Width Request.

The following events are defined to be BSS channel width trigger events (TEs):

— TE-A: On any of the channels of the channel set defined in Clause 18 (Extended Rate PHY (ERP) specification), reception of a Beacon frame that does not contain an HT Capabilities element.

— TE-B: On any of the channels of the channel set defined in Clause 18 (Extended Rate PHY (ERP) specification), reception of a 20/40 BSS Coexistence Management, Beacon, Probe Request, or Probe Response frame that contains a value of 1 in a Forty MHz Intolerant field and that has the Address 1 field equal to the receiving STA’s address or to a group address ~~value~~, with no further addressing qualifications.

— TE-C: Reception of a 20/40 BSS Coexistence Management frame with the 20 MHz BSS Width Request field equal to 1 and with an Address 1 field that matches the receiving STA using either individual or group addressing and with a TA field that corresponds to the MAC address of a STA with which the receiver is associated.

— TE-D: Reception of a 20/40 BSS Coexistence Management frame containing at least one 20/40 BSS Intolerant Channel Report element with a nonzero length and with an Address 1 field equal to the receiving STA’s address or to a group address ~~value~~, but with no qualification of the Address 3 field.

**11.16 20/40 BSS Coexistence Management frame usage**

NOTE—A 20/40 BSS Coexistence Management frame is a Class 1 frame and, therefore, can be sent to a STA that supports reception of such frames and that is not a member of the same BSS as the transmitting STA. In such a case, the BSSID field of the frame is set to the wildcard BSSID ~~value~~, regardless of whether the Address 1 field contains an individual or group address ~~value~~.

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 422 in <this document>, which refer to TAs etc. as TAs or as TA field values, as appropriate. Also group addresses as group addresses, not group address values.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 350Mark RISON3.2162.23 | Definition of MAC service tuple says "source address, destination addresses, priority, drop eligibility, service class, optional set of service\_access\_point\_identifiers, and optional indication of whether the supplied MSDU is in Ethertype protocol discrimination (EPD) or logical link control (LLC) protocol discrimination (LPD) format [...] are all passed as parameters across the MAC service access point (SAP)" but 5.2.3.2 has "source address, destination address, routing information, data, priority, drop eligible, service class, station vector, MSDU format" and 5.2.4.2 has "source address, destination address, routing information, data, reception status, priority, drop eligible, service class, station vector, MSDU format", so "optional set of service\_access\_point\_identifiers", "station vector" are only mentioned in one but not the other | Align the lists |

Discussion:

Here, green highlights parameters that match and cyan parameters that are fixed, so not an issue. Grey highlights the issues.

5.2.3.2 says:

MA-UNITDATA.request(

source address,

destination address,

routing information,

data,

priority,

drop eligible,

service class,

station vector,

MSDU format

)

5.2.4.2 says:

MA-UNITDATA.indication(

source address,

destination address,

routing information,

data,

reception status,

priority,

drop eligible,

service class,

station vector,

MSDU format

)

3.2 says:

**medium access control (MAC) service tuple:** The collection of a MAC service data unit (MSDU) along with the associated source address, destination addresses, priority, drop eligibility, service class, optional set of service\_access\_point\_identifiers, and optional indication of whether the supplied MSDU is in Ethertype protocol discrimination (EPD) or logical link control (LLC) protocol discrimination (LPD) format, which are all passed as parameters across the MAC service access point (SAP) and are all except the service\_access\_point\_identifiers delivered across the distribution system between access points (APs), mesh gates, and the portal of an extended service set (ESS).

In fact, the “(optional) set of service\_access\_point\_identifiers” is the “station vector” (sic, from 802.1AC -- it’s not really a vector because it doesn’t have an ordering); it is optional because it only applies to GLK. However, the MA SAP at least generally describes parameters as potentially having null values rather than being optional (cf. drop eligibility (11aa) and MSDU format (11ak)). Having said that, it is not clear what a non-GLK STA, or more precisely a non-EPD STA (definition in 4.3.29: “An EPD STA is a STA that supports EPD format MSDUs.”) is supposed to do for the MSDU format.

Also, a MAC service tuple only has one destination address, and the station vector and MSDU format are not present in MA-UNITDATA-STATUS.indication.

Proposed changes:

Change 3.1 definitions as follows:

**medium access control (MAC) service tuple:** The collection of a MAC service data unit (MSDU) along with the associated source address, destination address~~es~~, priority, drop eligibility, service class, ~~optional set of service\_access\_point\_identifiers~~ station vector, and ~~optional indication of whether the supplied~~ MSDU ~~is in Ethertype protocol discrimination (EPD) or logical link control (LLC) protocol discrimination (LPD)~~ format, which are all passed as parameters across the MAC service access point (SAP) and are all except the ~~service\_access\_point\_identifiers~~ station vector delivered across the distribution system between access points (APs), mesh gates, and the portal of an extended service set (ESS).

Add a Clause 3.2 definition as follows:

**station vector:** A set of service\_access\_point\_identifiers.

Change 5.2.2 GLK MAC data service specification as follows:

A GLK STA coordinates with the GLK convergence function to create a virtual point-to-point LAN for each general link to an associated or peered GLK STA. This point-to-point LAN is presented by the convergence function as a unique Internal Sublayer Service SAP, which is ultimately mapped to an IEEE 802.1Q bridge port. Each such SAP is identified by a locally unique service\_access\_point\_identifier (as defined in IEEE Std 802.1Q), generated by the STA and the convergence function.

~~In a GLK STA the MAC data plane architecture’s MAC service uses the two parameters, a set of service\_access\_point\_identifiers and an MSDU format indicator.~~

When GLK is in use, the MAC service ~~primitives presented require the presence~~ makes use of the parameter station vector in the MA-UNITDATA primitives ~~on request and supply a non-null value for this parameter on indication~~. In an MA-UNITDATA.request primitive, this parameter is a set of service\_access\_point\_identifiers specifying the one or more general links that are to ~~be used for~~carry th~~is request~~e MSDU. In an MA-UNITDATA.indication primitive, it is a ~~vector~~set of exactly one service\_access\_point\_identifier specifying the general link that carried th~~is~~e MSDU.

When GLK is in use, the MAC service ~~primitives~~ also makes use of the MSDU format parameter. In an MA-UNITDATA.request primitive, this parameter indicates ~~whether~~the format of the supplied MSDU ~~is in~~ (EPD or LPD) ~~format~~. If the format is inappropriate for the transmission that carries this MSDU as described in 5.1.4 (MSDU format), the ~~STA~~MAC service converts the format before transmission. In an MA-UNITDATA.indication primitive, this parameter indicates the format of the received MSDU, which is determined as described in 5.1.4 (MSDU format). The MAC service user uses this information to parse the MSDU correctly.

OR

~~In a GLK STA the MAC data plane architecture’s MAC service uses the two parameters, a set of service\_access\_point\_identifiers and an MSDU format indicator.~~

When GLK is in use, the MAC service ~~primitives presented require the presence~~ makes use of the following parameters in the MA-UNITDATA primitives:

* ~~the parameter s~~Station vector ~~on request and supply a non-null value for this parameter on indication~~. In an MA-UNITDATA.request primitive, ~~this parameter~~it is a set of service\_access\_point\_identifiers specifying the one or more general links that are to ~~be used for~~carry th~~is request~~e MSDU. In an MA-UNITDATA.indication primitive, it is a ~~vector~~set ofexactly one service\_access\_point\_identifier specifying the general link that carried th~~is~~e MSDU.
* ~~When GLK is in use, the MAC service primitives make use of the~~ MSDU format ~~parameter~~. In an MA-UNITDATA.request primitive, ~~this parameter~~it indicates ~~whether~~the format of the supplied MSDU ~~is in~~ (EPD or LPD) ~~format~~. If the format is inappropriate for the transmission that carries this MSDU as described in 5.1.4 (MSDU format), the ~~STA~~MAC service converts the format before transmission. In an MA-UNITDATA.indication primitive, ~~this parameter~~it indicates the format of the received MSDU, which is determined as described in 5.1.4 (MSDU format). The MAC service user uses this information to parse the MSDU correctly.

OR

~~In a GLK STA the MAC data plane architecture’s MAC service uses the two parameters, a set of service\_access\_point\_identifiers and an MSDU format indicator.~~

When GLK is in use, the MAC service ~~primitives presented require the presence~~ makes use of parameters in the MA-UNITDATA primitives as shown in Table 5-x.

~~When GLK is in use, the MAC service primitives presented require the presence of the parameter station vector on request and supply a non-null value for this parameter on indication. In an MA-UNITDATA.request primitive, this parameter is a set of service\_access\_point\_identifiers specifying the one or more general links that are to be used for this request. In an MA-UNITDATA.indication primitive, it is a vector of exactly one service\_access\_point\_identifier specifying the general link that carried this MSDU.~~

~~When GLK is in use, the MAC service primitives make use of the MSDU format parameter. In an MA-UNITDATA.request primitive, this parameter indicates whether the supplied MSDU is in EPD or LPD format. If the format is inappropriate for the transmission that carries this MSDU as described in 5.1.4 (MSDU format), the STA converts the format before transmission. In an MA-UNITDATA.indication primitive, this parameter indicates the format of the received MSDU, which is determined as described in 5.1.4 (MSDU format). The MAC service user uses this information to parse the MSDU correctly.~~

**Table 5-x—MA-UNITDATA primitive parameter usage in GLK**

|  |  |  |
| --- | --- | --- |
| Parameter | MA-UNITDATA.request | MA-UNITDATA.indication |
| station vector | A set of service\_access\_point\_identifiers specifying the one or more general links that are to carry the MSDU. | A set of exactly one service\_access\_point\_identifier specifying the general link that carried the MSDU. |
| MSDU format | Indicates the format of the supplied MSDU (EPD or LPD). If the format is inappropriate for the transmission that carries this MSDU as described in 5.1.4 (MSDU format), the MAC service converts the format before transmission. | Indicates the format of the received MSDU, which is determined as described in 5.1.4 (MSDU format). The MAC service user uses this information to parse the MSDU correctly. |

Change 5.2.3.2 Semantics of the service primitive [MA-UNITDATA.req] as follows:

~~T~~If dot11GLKImplemented is true, the station vector parameter is a set of service\_access\_point\_identifiers (see 5.2.2 (GLK MAC data service specification)) ~~and is not null when dot11GLKImplemented is true and is null or not present otherwise. It~~ that indicates the set of virtual point-to-point LANs for these data transfers, which are mapped to the set of general links over which the MSDU is trans~~ferred~~mitted; it is null otherwise.

~~T~~In an EPD STA, the MSDU format parameter indicates ~~whether~~ the format of the supplied MSDU ~~is in~~ (EPD or LPD) ~~format~~; it is ignored otherwise.

Change 5.2.4.2 Semantics of the service primitive [MA-UNITDATA.ind] as follows:

~~T~~If dot11GLKImplemented is true, the station vector parameter is a set of service\_access\_point\_identifiers (see 5.2.2 (GLK MAC data service specification)) ~~and is not null when dot11GLKImplemented is true. It~~ that indicates ~~only~~ the single virtual point-to-point LAN for this data transfer, which is mapped from the general link over which the MSDU was received; it is null otherwise.

~~T~~In an EPD STA, the MSDU format parameter indicates ~~if~~ the format of the received MSDU ~~is in~~ (EPD or LPD) ~~format~~; it indicates LPD otherwise.

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 350 in <this document>, which clarify the relationship between the so-called station vector and sets of service\_access\_point\_identifiers, clarify the optionality/nullness of MA-UNITDATA parameters and make various editorial improvements.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 488Mark RISON6 | Beacons, DMG Beacons, S1G Beacons and Probe Responses allow for multiple Multiple BSSID elements, but the MLME SAP only allows for a single one | Extend the MLME SAP primitives that carry a Multiple BSSID element to allow more than one such element |

Discussion:

E.g. for beacons:



However, the MLME primitives don’t cover the interworking+GAS thing, and stuff is missing for DMG and S1G beacons.

Abhishek PATIL has done some further digging:

I dug thru 802.11k, .11v, .11u and .11ai specs. Each of these standards added / expanded MBSSID feature and as a result, we see different conditions being added to the Beacon frame for inclusion of MBSSID IE.

--------

**802.11k** was the first standard to introduce MBSSID. The main focus here was RadioMeasurement.

MIB: dot11RadioMeasurementEnabled



\* same conditions added for Probe Response frame

------------

**802.11v** introduced mechanisms for reducing mgmt. frame overhead – i.e., only the TxBSSID transmits beacons, shared AID space and single TIM, nonTxBSSID profile is carried in a subelement contained in the MBSSID IE (introduction of Multiple BSSID-Index element and Non-transmitted BSSID Capability element etc). Introduction of Method B for efficient traffic indication in PVB of TIM IE. In short, all the good stuff to improve efficiency.

MIB: dot11MgmtOptionMultiBSSIDActivated. This was later deprecated and only dot11MultiBSSIDImplemented is currently being used





\* same conditions added for Probe Response frame

----------

**802.11u** – don’t really understand why .11u added new condition for inclusion of MBSSID IE. Since 11v precedes it, it can as well leverage on “dot11MultiBSSIDImplemented equal true”. I think we can remove the conditions added by 11u.



\* same conditions added for Probe Response frame

-------

**802.11ai** mandated support for MBSSID for all FILS STA. We could address the FILS case (in clause 6.3 and 9.3.3) by adding a statement in clause 11.1.3.8 that says a FILS STA (i.e., a STA with dot11FILSActivated equal to true) shall set dot11MultiBSSIDImplemented to true. What do you think? This can eliminate additional conditions being added to clause 6 and 9 for inclusion of MBSSID IE in Beacon and Probe Resp.



FYI, the same conditions would apply to 11ax AP/STA so could expand the statement cover 11ax too.

---------

BTW, while investigating the evolution of MBSSID feature, I came across the following text in baseline standard which needs to be fixed. Please see text on pg 2291 in REVme D0.00:



The highlighted portion would apply ONLY when dot11RMMeasurementPilotActivated is between 2 and 7 and dot11MultiBSSIDImplemented is false. In other words, the paragraph applies to 11k based APs only not to APs that have 11v based MBSSID feature implemented. An AP that has dot11MultiBSSIDImplemented equal true will transmit a single Beacon on behalf of all the BSSIDs in the set (i.e., only TxBSSID would transmit a Beacon or a Probe Resp frame).

Proposed changes:

Change 6.3.3.3.2 Semantics of the service primitive (MLME-SCAN.cfm) as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Valid range | Description | IBSS adoption |
| Multiple BSSID  | A~~s defined in~~ set ofMultiple BSSID elements | As defined in 9.4.2.45 (Multiple BSSID element) | The values from the Multiple BSSID element(s) ~~if such an element was~~ present in the Probe Response, Beacon, DMG Beacon or S1G Beacon frame, if any, else null. | Do not adopt |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid range | Description |
| Multiple BSSID ~~element~~ | A set of Multiple BSSID elements | As defined in 9.4.2.45 (Multiple BSSID element) | ~~Indicates that the BSS is within a multiple BSSID set (see 11.10.14 Multiple BSSID set)). The range of BSSIDs is determined by the BSSID and Multiple BSSID element~~The values from the Multiple BSSID element(s) present in the Measurement Pilot frame, if any. |

Change 6.3.11.2.2 Semantics of the service primitive (MLME-START.req) as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid range | Description |
| Multiple BSSID ~~element~~ | A~~s defined in~~ set of Multiple BSSID elements ~~in 9.4.2.45 (Multiple BSSID element)~~ | As defined in ~~Multiple~~ ~~BSSID element in~~ 9.4.2.45 (Multiple BSSID element) | ~~This~~One or more elements are ~~is~~ ~~optionally~~ present ~~when~~if any of the following conditions are true:* dot11MultiBSSIDImplemented is true
* dot11RMMeasurementPilotActivated is ~~a value~~ between 2 and 7 and the AP is a member of a multiple BSSID set (see 11.10.14 (Multiple BSSID set)) with two or more members~~, or if~~
* ~~dot11MultiBSSIDImplemented is true~~
* ~~. This elements is present when~~ dot11FILSActivated is true and the AP is a member of a ~~M~~multiple BSSID ~~S~~set with two or more members~~.~~
* dot11InterworkingServiceActivated is true and the AP is a member of a multiple BSSID set with two or more members and at least one dot11GASAdvertisementID exists
 |

Change Table 9-32—Beacon frame body as follows:

|  |  |  |
| --- | --- | --- |
| 27 | Multiple BSSID | One or more Multiple BSSID elements are present if any of the following conditions are true:* dot11MultiBSSIDImplemented is true
* dot11RMMeasurementPilotActivated is ~~a value~~ between 2 and 7 and the AP is a member of a multiple BSSID set (see 11.10.14 (Multiple BSSID set)) with two or more members
* ~~, or if dot11MultiBSSIDImplemented is true,~~
* dot11FILSActivated is true and the AP is a member of a multiple BSSID set with two or more members
* ~~or if~~ dot11InterworkingServiceActivated is true and the AP is a member of a multiple BSSID set with two or more members and at least one dot11GASAdvertisementID exists~~.~~
 |

Ditto Table 9-39—Probe Response frame body, row 20.

Change Table 9-45—DMG Beacon frame body as follows:

|  |  |  |
| --- | --- | --- |
| 10 | Multiple BSSID | One or more Multiple BSSID elements are ~~optionally~~ present if any of the following conditions are true:* dot11MultiBSSIDImplemented is true~~.~~
* dot11RMMeasurementPilotActivated is between 2 and 7 and the AP is a member of a multiple BSSID set (see 11.10.14 (Multiple BSSID set)) with two or more members
* dot11FILSActivated is true and the AP is a member of a multiple BSSID set with two or more members
* dot11InterworkingServiceActivated is true and the AP is a member of a multiple BSSID set with two or more members and at least one dot11GASAdvertisementID exists
 |

Change Table 9-46—Minimum and full set of optional elements as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 16 | Multiple BSSID | One or more Multiple BSSID elements are present if any of the following conditions are true:* dot11MultiBSSIDImplemented is true~~;~~
* dot11RMMeasurementPilotActivated is between 2 and 7 and the AP is a member of a multiple BSSID set (see 11.10.14 (Multiple BSSID set)) with two or more members
* dot11FILSActivated is true and the AP is a member of a multiple BSSID set with two or more members
* dot11InterworkingServiceActivated is true and the AP is a member of a multiple BSSID set with two or more members and at least one dot11GASAdvertisementID exists

~~o~~Otherwise not present. | No | Yes |

TBD: is this sufficient for Announce frames? “The Multiple BSSID element is defined in 9.4.2.45 (Multiple BSSID element). The Multiple BSSID element is optionally present. If present, the Multiple BSSID element signals all the BSSIDs in use by the BSS.”

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 488 in <this document>, which address the changes requested by the commenter.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 240Mark RISON | A Mesh TKSA is pairwise and bidirectional, just like a PTKSA. So it would be better to call it a Mesh PTKSA | Change "esh TKSA" to "esh PTKSA" throughout |

Discussion:

As it says in the comment. Also note “TKSA” is not defined. SAKODA Kazuyuki has described making this change as “reasonable”.

Proposed resolution:

ACCEPTED

Note to the Editor: the instances are in 4.5.4.3, 12.6.1.1.1, 12.6.1.1.5, 12.6.1.1.7 (3x inc. heading), 12.6.1.1.9, 12.6.7, 12.6.12 (2x), 12.6.16 (4x), 12.10.1, 12.10.2 (4x), 14.3.4.1, 14.3.4.3, 14.5.1 (2x), 14.6.3, 14.6.4, 14.6.5, 14.7 (2x).

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 462Mark RISON5.2.4 | Since broadcast frames typically have a lower security assurance than unicast frames (e.g. they are encrypted with the GTK, which is known to all members of the BSS and all ex-members until GTK rekeying occurs) it might be helpful for upper layers to know the security context under which an MSDU was received | Add a "security context" parameter to MA-UNITDATA.indication, with possible values None, Pairwise, Group |

Discussion:

As it says in the comment. This would e.g. support Linux’s drop\_unicast\_in\_l2\_multicast option.

Counter-argument:

1) For vanilla 802.11, we can deal with the case of 802.11 RA = group, 802.11 DA = unicast with our new 21/0816 rules (and the "should drop if violate")

2) The case of 802.11 RA = group, IP DA = unicast is the IP layer's problem. It gets the 802.11 DA in the MA-UNITDATA.ind, and per the previous, if the 802.11 RA was group the 802.11 DA will also be group, so it can do the check itself

3) This falls apart for things like GLK (and maybe e.g. mesh and DMG/S1G relay?) that need to allow 802.11 RA = group, 802.11 DA = unicast. That's too bad. All we can do is draw the reader's attention to the security risk

Proposed resolution:

REVISED

In 5.2.4.2, after the “reception status,” line add a “security context,” line and after the para starting “The reception status parameter indicates” add a para “The security context indicates the type of key under which the MSDU was received: None, Pairwise, PeerKey or Group.”

Counter-proposal:

At the end of the footnote corresponding to “A STA that receives a frame containing an A-MSDU that violates these rules should discard it.666” in 21/0816 add “The lack of checks on the SA and DA in the case of mesh and relay operation presents an attack surface that is outside the scope of remediation in this standard. In the case of GLK, use of a SYNRA for an individually addressed MSDU with multiple links indicated in the station vector parameter also presents such an attack surface, which can be avoided by transmitting multiple individually addressed MPDUs instead (see 10.65).”

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 239Mark RISON12.5 | The replay detection/protection subclauses don't cover MBSS links (that use a Mesh TKSA and a Mesh GTKSA rather than a PTKSA and a GTKSA) | Extend 12.5.3.4.4 PN and replay detection [CCMP] and 12.5.5.4.4 PN and replay detection [GCMP] to mention Mesh TKSA and GTKSA together with PTKSA and GTKSA |

Discussion:

As it says in the comment.

Proposed changes:

a) The receiver shall maintain a separate set of replay counters for each PTKSA, GTKSA, ~~and~~ protocol version value, mesh TKSA and mesh GTKSA. 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~~ protocol version value, mesh TKSA and mesh GTKSA, the recipient shall maintain a separate replay counter for each TID […]

and similarly for GCMP. (Note “An S1G STA is a nonmesh STA”, so the “protocol version value” stuff does not apply to mesh SAs.)

Proposed resolution:

REVISED

In 12.5.3.4.4 PN and replay detection change “PTKSA, GTKSA, and protocol version value” to “PTKSA, GTKSA, protocol version value, mesh TKSA and mesh GTKSA” in a) and b).

In 12.5.5.4.4 PN and replay detection change “PTKSA and GTKSA” to “PTKSA, GTKSA, mesh TKSA and mesh GTKSA” in a) and b).

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 294Mark RISON9.4.2.21.71044.14 | The "might"s here are actually "may"s. Since "may" are not allowed in Clause 9, reword as "In this case, some of the elements included in the Reported FrameBody subelement might be truncated, and the subelement itself might be truncated or fragmented overmultiple Beacon Reports when its size exceeds the maximum element size, as described below:-- Truncation of Reported TIM elements such that only the first 4 octets of theelement are reported and the element Length field is modified to indicate the truncated length of 4." etc. | As it says in the comment |

Discussion:

This is really about behaviour for stuff that won’t fit, rather than format. There are a number of additional issues:

* There is discussion of “fragmentation” but these are not what is normally understood as fragments, i.e. MPDU fragments, nor are they the element fragments from 11ai
* There is discussion of support for fragmentation, but it is not clear whether this is at the transmitter or at the receiver. The intent seems to be that support is optional at the transmitter, but a receiver is required to support this, unless it only requests measurements that will always fit (through the Reporting Detail field)
* Some locations refer to Request elements but fail to refer to Extended Request elements too
* The references to the Reporting Detail field are editorially haphazard

Proposed changes:

Change 9.4.2.21.7 Beacon report as follows:

The Reported Frame Body subelement, if present, contains some or all of the ~~requested~~ fields and elements of the frame body of the reported Beacon, Measurement Pilot, or Probe Response frame. If the Reporting Detail field of the Reporting Detail subelement of the corresponding Beacon request equals 0, the Reported Frame Body subelement is not included in the Beacon report. ~~If the Reporting Detail subelement equals 1, all fields and any elements identified in a Request element or Extended Request element in the corresponding Beacon request are included in the Reported Frame Body subelement, in the order that they appeared in the reported frame.~~

~~If the Reporting Detail field equals 2, all fields and elements are included in the order they appeared in the reported frame.~~

Some elements in the Reported Frame Body subelement, if present, or the Reported Frame Body subelement itself, might be large. In this case, some of the elements included in the Reported Frame Body subelement ~~might~~can be truncated or omitted, and the subelement itself ~~might~~can be truncated or fragmented over multiple Beacon ~~R~~reports (see 11.10.9.1.1). ~~when its size exceeds the maximum element size, as described below:~~

NOTE—This is not element fragmentation as defined in 10.28.11.

~~— Reported TIM elements might be truncated such that only the first 4 octets of the element are reported and the element Length field is modified to indicate the truncated length of 4.~~

~~— Reported IBSS DFS elements might be truncated so that only the lowest and highest channel number map are reported and the element Length field is modified to indicate the truncated length of 13.~~

~~— Reported RSNEs might be truncated so that only the first 4 octets of the element are reported and the element Length field is modified to indicate the truncated length of 4.~~

~~— If the length of the Reported Frame Body subelement would cause the Measurement Report element to exceed the maximum element size, when Reported Frame Body subelement fragmentation is not supported, then the Reported Frame Body subelement is truncated so that the last element in the Reported Frame Body subelement is a complete element.~~

~~— If the length of the Reported Frame Body subelement would cause the Measurement Report element to exceed the maximum element size, when Reported Frame Body subelement fragmentation is supported, then the Reported Frame Body subelement is fragmented over multiple Beacon Reports. When the Reported Frame Body subelement is fragmented, then the Reported Frame Body Fragment ID subelement is present in each Beacon Report frame that contains a fragment of this Reported Frame Body subelement. When the Reported Frame Body Fragment ID subelement is present, the reporting STA does not truncate any of the elements included into the Reported Frame Body subelement.~~

The Reported Frame Body Fragment ID subelement is present if a Reported Frame Body subelement is fragmented, and is not present otherwise. The format of the Data field of the Reported Frame Body Fragment ID subelement is shown in Figure 9-232 (Data field format).

The Beacon Report ID field identifies the reported frame for which the Beacon ~~R~~reports ~~instance~~ are sent as a response to a Beacon ~~R~~request. **<para break>**

The ~~responding STA sets the~~ Fragment ID Number field identifies the fragment of the  ~~to 0 for the initial fragment and increments it by 1 for each subsequent fragment in a multi-~~fragmented Reported Frame Body subelement ~~of a Beacon Report identified by the Beacon Report ID~~.

The More Frame Body Fragments field is set to 0 ~~whenever~~in the final fragment of ~~a~~ the fragmented Reported Frame Body subelement ~~is being transmitted~~, otherwise it is set to 1.

~~The Reported Frame Body subelement is fragmented so that the last element in every Reported Frame Body subelement fragment is a complete element.~~

Change 9.4.2.21.7 Beacon report as follows:

If the Reporting Detail field equals 1, a Request element ~~is~~and/or an Extended Request element are optionally present in the list of optional subelements. If included, ~~the Request element lists~~these identify ~~the Element IDs of~~ the elements requested to be reported in the Reported Frame Body subelement of the Beacon report.

[in Table 9-106] All fixed-length fields and any requested elements in the Request element and/or Extended Request element if present

Change 11.10.9.1 Beacon report as follows:

If a STA accepts a Beacon request it shall respond with a Radio Measurement Report frame containing Beacon reports for all observed BSSs matching the BSSID and SSID in the Beacon request, at the level of detail requested in the Reporting Detail field of the Reporting Detail subelement:~~. If the Reporting Detail is 1 and the optional Request subelement is included in the Beacon request, the corresponding Beacon report shall include the list of elements listed in the Request subelement.~~

* If the Reporting Detail field equals 0, the Reported Frame Body subelement shall not be included in the Beacon report(s).
* If the Reporting Detail field equals 1, all fields, and any elements identified in a Request element and/or an Extended Request element if present in the corresponding Beacon request, shall be included, subject to Beacon report truncation (see 11.10.9.1.1), in the Reported Frame Body subelement (or the Reported Frame Body subelements of multiple Beacon reports, in the case of Reported Frame Body subelement fragmentation), in the order that they appeared in the reported frame
* If the Reporting Detail field equals 2, all fields and elements shall be included, subject to Beacon report truncation (see 11.10.9.1.1), in the order they appeared in the reported frame.

***<para break>*** The RCPI in the Beacon report indicates the power level of the received Beacon, Measurement Pilot, or Probe Response frame. For repeated measurements (when the Radio Measurement Request frame contains a nonzero value for the Number of Repetitions field), the transmission of the Beacon report may be conditional on the measured RCPI or RSNI value. If the Radio Measurement Request frame contains a 0 value for the Number of Repetitions field, the Beacon Reporting subelement shall not be included in the Beacon request. If the Radio Measurement Request frame contains a nonzero value for the Number of Repetitions field, and if both dot11RMBeaconMeasurementReportingConditionsActivated and dot11RMRepeatedMeasurementsActivated are true, and if a Beacon Reporting subelement is included in a Beacon request, the STA shall respond with a Beacon report if the indicated Beacon reporting condition is true. Otherwise, the STA shall not respond with a Beacon report. Table 9-105 (Reporting Condition subfield for Beacon report) lists the reporting conditions that are based on the measured RCPI or RSNI levels.

Add a new subclause:

 **11.10.9.1.1 Truncation and/or fragmentation of reported frame body in Beacon report**

If a Reported Frame Body subelement in a Beacon report would exceed the maximum subelement size or would cause the Measurement Report element in the Beacon report to exceed the maximum element size, and the STA transmitting the Beacon report supports Reported Frame Body subelement fragmentation, the Reported Frame Body subelement shall be fragmented as follows:

— The payload of the Reported Frame Body subelement is fragmented across two or more Beacon reports

— A Reported Frame Body Fragment ID subelement is present in each Beacon report

— The Beacon Report ID subfield in the Reported Frame Body Fragment ID subelement of each Beacon report is the same, and is different from that of Beacon reports corresponding to a different reported frame

— The Fragment ID Number subfield in the Reported Frame Body Fragment ID subelement of the first Beacon report is set to 0 and is incremented by 1 for each subsequent Beacon report

— The More Frame Body Fragments field in the Reported Frame Body Fragment ID subelement is set to 1 in all except the last Beacon report

— Elements in the Reported Frame Body subelement are not truncated, split across two Beacon reports, or omitted

NOTE 1—This means the last element in the Reported Frame Body subelement of each Beacon report is a complete element.

NOTE 2—The STA requesting a Beacon report must support Reported Frame Body subelement (de)fragmentation unless it sets the Reporting Detail subelement in the Beacon request to ensure the STA sending the Beacon report does not use fragmentation.

If a Reported Frame Body subelement in a Beacon report would exceed the maximum subelement size or would cause the Measurement Report element in the Beacon report to exceed the maximum element size, and the STA transmitting the Beacon report does not support Reported Frame Body subelement fragmentation, reported elements shall be truncated or omitted as follows to make them fit:

— A TIM element may be truncated such that only the first 4 octets of the element are reported and the element Length field is modified to indicate the truncated length of 4.

— A IBSS DFS element may be truncated so that only the lowest and highest channel number map are reported and the element Length field is modified to indicate the truncated length of 13.

— An RSNE may be truncated so that only the first 4 octets of the element are reported and the element Length field is modified to indicate the truncated length of 4.

— Elements may be omitted from the end of Reported Frame Body subelement

NOTE 3—Elements are not truncated or omitted if the STA transmitting the Beacon report supports Reported Frame Body subelement fragmentation.

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 294 in <this document>, which separate the format from the behaviour, and address various technical and editorial issues.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 477Mark RISON | It is not clear what an "entity" is | Change "peer entity" to "peer STA" (x86) |

Discussion:

We have PHY, MAC, management, 802.1X, FST, sublayer, port access, multi-band and relay entities, but the peer entities in Clause 6 are just peer STAs.

Proposed changes:

Change “peer entity” to “peer STA” at the following locations:

|  |  |  |  |
| --- | --- | --- | --- |
| Page | Line | Subclause | Context |
| 406 | 13 | 6.3.14.2.1 Function | a **peer entity**. 6.3.14.2.2 Semantics of the service primitive The primitive parameters are as |
| 407 | 4 | 6.3.14.2.3 When generated | a **peer entity** to initiate one or more measurements. 6.3.14.2.4 Effect of receipt On |
| 410 | 27 | 6.3.16.2.3 When generated | a **peer entity** to report the results of measuring one or more channels. 6.3.16.2.4 |
| 410 | 41 | 6.3.16.3.1 Function | a **peer entity**. This management report is either a response to an |
| 413 | 33 | 6.3.17.3.1 Function | a **peer entity**. 6.3.17.3.2 Semantics of the service primitive The primitive parameters are as |
| 416 | 34 | 6.3.18.2.3 When generated | a **peer entity** to request that entity to report transmit power and link |
| 451 | 42 | 6.3.29.3.1 Function | a **peer entity**. 6.3.29.3.2 Semantics of the service primitive The primitive parameters are as |
| 457 | 52 | 6.3.32.2.3 When generated | **peer entity** to request that entity to report transmit power and |
| 506 | 6 | 6.3.48.2.1 Function | a **peer entity**. 6.3.48.2.2 Semantics of the service primitive The primitive parameters are as |
| 506 | 41 | 6.3.48.2.3 When generated | a **peer entity** to initiate one or more transactions. 6.3.48.2.4 Effect of receipt On receipt |
| 508 | 18 | 6.3.49.2.3 When generated | a **peer entity** to report the results of one or more transactions. 6.3.49.2.4 Effect |
| 508 | 33 | 6.3.49.3.1 Function | a **peer entity**. 6.3.49.3.2 Semantics of the service primitive The primitive parameters are |
| 511 | 6 | 6.3.51.2.1 Function | a **peer entity**. 6.3.51.2.2 Semantics of the service primitive The primitive parameters are as |
| 511 | 42 | 6.3.51.2.3 When generated | a **peer entity** to initiate one or more diagnostic transactions. 6.3.51.2.4 Effect of receipt On |
| 512 | 61 | 6.3.52.1.3 When generated | a **peer entity** to report the results of one or more diagnostic transactions. Name |
| 513 | 28 | 6.3.52.2.1 Function | a **peer entity**. 6.3.52.2.2 Semantics of the service primitive The primitive parameters are as |
| 514 | 49 | 6.3.53.2.1 Function | a **peer entity**. 6.3.53.2.2 Semantics of the service primitive The primitive parameters are as |
| 515 | 22 | 6.3.53.2.3 When generated | **peer entity** to convey location configuration information. 6.3.53.2.4 Effect of receipt On receipt |
| 517 | 15 | 6.3.53.5.1 Function | a **peer entity**, in response to a Location Configuration Request frame. 6.3.53.5.2 Semantics of |
| 517 | 52 | 6.3.53.5.3 When generated | **peer entity** to convey location configuration information. 6.3.53.5.4 Effect of receipt On receipt |
| 518 | 33 | 6.3.54.2.1 Function | a **peer entity**. 6.3.54.2.2 Semantics of the service primitive The primitive parameters are as |
| 519 | 15 | 6.3.54.2.3 When generated | a **peer entity** to help convey location information. 6.3.54.2.4 Effect of receipt On receipt |
| 520 | 58 | 6.3.55.2.1 Function | a **peer entity**. Figure 6-16—Timing measurement primitives and timestamps capture MLME |
| 521 | 26 | 6.3.55.2.3 When generated | **peer entity**. 6.3.55.2.4 Effect of receipt On receipt of this primitive, the MLME |
| 522 | 15 | 6.3.55.4.1 Function | a **peer entity**. 6.3.55.4.2 Semantics of the service primitive The primitive parameters are as |
| 523 | 14 | 6.3.55.4.3 When generated | a **peer entity**. 6.3.55.4.4 Effect of receipt On receipt of this primitive, the MLME |
| 526 | 47 | 6.3.56.2.1 Function | a **peer entity**. 6.3.56.2.2 Semantics of the service primitive The primitive parameters are as |
| 527 | 50 | 6.3.56.2.3 When generated | a **peer entity**. 6.3.56.2.4 Effect of receipt On receipt of this primitive, the MLME |
| 529 | 10 | 6.3.56.4.1 Function | a **peer entity**. 6.3.56.4.2 Semantics of the service primitive The primitive parameters are as |
| 530 | 33 | 6.3.56.4.3 When generated | a **peer entity**. 6.3.56.4.4 Effect of receipt On receipt of this primitive, the MLME |
| 545 | 41 | 6.3.58.5.3 When generated | a **peer entity** to convey FMS information. 6.3.58.5.4 Effect of receipt On receipt of |
| 546 | 45 | 6.3.59.2.1 Function | a **peer entity**. 6.3.59.2.2 Semantics of the service primitive The primitive parameters are as |
| 547 | 32 | 6.3.59.2.3 When generated | **peer entity**. 6.3.59.2.4 Effect of receipt On receipt of this primitive, the MLME |
| 548 | 51 | 6.3.60.2.1 Function | a **peer entity**, in response to a Collocated Interference Request frame. 6.3.60.2.2 Semantics of |
| 549 | 23 | 6.3.60.2.3 When generated | **peer entity** to convey collocated interference information. 6.3.60.2.4 Effect of receipt On receipt |
| 556 | 42 | 6.3.62.4.1 Function | a **peer entity**, in response to a WNM Sleep Mode Request frame. 6.3.62.4.2 Semantics |
| 556 | 62 | 6.3.62.4.3 When generated | a **peer entity** to convey WNM sleep mode information. Name Type Valid range |
| 575 | 53 | 6.3.67.2.1 Function | a **peer entity**. 6.3.67.2.2 Semantics of the service primitive The primitive parameters are as |
| 576 | 27 | 6.3.67.2.3 When generated | a **peer entity** to initiate a WNM notification. 6.3.67.2.4 Effect of receipt On receipt |
| 578 | 3 | 6.3.68.1.3 When generated | a **peer entity** to report the results of the WNM notification. 6.3.68.1.4 Effect |
| 578 | 18 | 6.3.68.2.1 Function | a **peer entity**. 6.3.68.2.2 Semantics of the service primitive The primitive parameters are as |
| 595 | 58 | 6.3.74.2.3 When generated | a **peer entity** to request the adjustment of the peer entity’s TBTT. Name |
| 595 | 59 | 6.3.74.2.3 When generated | the **peer entity**’s TBTT. Name Type Valid range |
| 617 | 33 | 6.3.80.2.3 When generated | specified **peer entity**. 6.3.80.2.4 Effect of receipt On receipt of this primitive, the MLME |
| 618 | 53 | 6.3.81.2.1 Function | a **peer entity**. 6.3.81.2.2 Semantics of the service primitive The primitive parameters are |
| 619 | 24 | 6.3.81.2.3 When generated | a **peer entity** to communicate QMF policy information. 6.3.81.2.4 Effect of receipt On receipt of |
| 619 | 39 | 6.3.81.3.1 Function | a **peer entity**. 6.3.81.3.2 Semantics of the service primitive The primitive parameters are as |
| 620 | 26 | 6.3.81.3.3 When generated | a **peer entity**. 6.3.81.3.4 Effect of receipt The SME is notified of the receipt |
| 620 | 58 | 6.3.81.4.3 When generated | a **peer entity**. 6.3.81.4.4 Effect of receipt On receipt of this primitive, the MLME |
| 622 | 16 | 6.3.81.6.1 Function | a **peer entity**. 6.3.81.6.2 Semantics of the service primitive The primitive parameters are as |
| 622 | 51 | 6.3.81.6.3 When generated | a **peer entity**. 6.3.81.6.4 Effect of receipt On receipt of this primitive, the parameters |
| 623 | 6 | 6.3.81.7.1 Function | a **peer entity**, in response to a QMF Policy Change frame. 6.3.81.7.2 Semantics of |
| 623 | 41 | 6.3.81.7.3 When generated | a **peer entity** to convey the results of the QMF policy change procedure. 6.3.81.7.4 |
| 635 | 53 | 6.3.84.4.1 Function | a **peer entity**. 6.3.84.4.2 Semantics of the service primitive The primitive parameters are |
| 639 | 35 | 6.3.85.4.1 Function | a **peer entity**. 6.3.85.4.2 Semantics of the service primitive The primitive parameters are |
| 642 | 6 | 6.3.86.3.1 Function | a **peer entity**. 6.3.86.3.2 Semantics of the service primitive The primitive parameters are |
| 676 | 30 | 6.3.96.2.1 Function | a **peer entity**. 6.3.96.2.2 Semantics of the service primitive The primitive parameters are |
| 677 | 4 | 6.3.96.2.3 When generated | the **peer entity**. 6.3.96.2.4 Effect of receipt On receipt of this primitive, the MLME |
| 677 | 50 | 6.3.96.3.2 Semantics of the service primitive | the **peer entity**. ENABLEMENT\_DENIED is used to indicate denial due to restriction |
| 678 | 7 | 6.3.96.3.3 When generated | the **peer entity** or when an unspecified failure occurs. 6.3.96.3.4 Effect of receipt On receipt |
| 678 | 21 | 6.3.96.4.1 Function | a **peer entity**. 6.3.96.4.2 Semantics of the service primitive The primitive parameters are |
| 678 | 38 | 6.3.96.4.2 Semantics of the service primitive | the **peer entity** from which a GDD Enablement Request frame was received. |
| 679 | 6 | 6.3.96.5.1 Function | the **peer entity**. 6.3.96.5.2 Semantics of the service primitive MLME-GDDENABLEMENT.response |
| 679 | 52 | 6.3.96.5.3 When generated | **peer entity**. 6.3.96.5.4 Effect of receipt On receipt of this primitive, the |
| 679 | 22 | 6.3.96.5.2 Semantics of the service primitive | the **peer entity** from which a GDD Enablement Request frame was received. DialogToken |
| 679 | 37 | 6.3.96.5.2 Semantics of the service primitive | the **peer entity**. ENABLEMENT\_DENIED is used to indicate denial due to restriction |
| 683 | 32 | 6.3.97.5.3 When generated | the **peer entity**. 6.3.97.5.4 Effect of receipt On receipt of this primitive, the MLME |
| 693 | 52 | 6.3.102.5.3 When generated | a **peer entity** to convey AID assignment information. 6.3.102.5.4 Effect of receipt On |
| 702 | 29 | 6.3.107.5.3 When generated | a **peer entity** to convey TWT assignment information. 6.3.107.5.4 Effect of receipt On receipt |
| 708 | 29 | 6.3.110.5.3 When generated | a **peer entity** to convey Header Compression information. 6.3.110.5.4 Effect of receipt On receipt |
| 710 | 14 | 6.3.112.2.1 Function | a **peer entity**. 6.3.112.2.2 Semantics of the service primitive The primitive parameters are |
| 710 | 48 | 6.3.112.2.3 When generated | the **peer entity**. 6.3.112.2.4 Effect of receipt On receipt of this primitive, the MLME |
| 710 | 54 | 6.3.112.2.4 Effect of receipt | the **peer entity**. Name Type Valid range |
| 711 | 37 | 6.3.112.3.3 When generated | the **peer entity**. 6.3.112.3.4 Effect of receipt On receipt of this primitive, the SME |
| 711 | 50 | 6.3.112.4.1 Function | a **peer entity**. 6.3.112.4.2 Semantics of the service primitive The primitive parameters are |
| 712 | 64 | 6.3.112.5.3 When generated | a **peer entity** to convey control response MCS negotiation information. Name Type |
| 713 | 4 | 6.3.112.5.4 Effect of receipt | the **peer entity** indicated by the PeerSTAAddress parameter. 6.3.113 S1G relay (de)activation 6.3.113.1 General The |
| 713 | 20 | 6.3.113.2.1 Function | a **peer entity**. 6.3.113.2.2 Semantics of the service primitive The primitive parameters are |
| 713 | 45 | 6.3.113.2.3 When generated | the **peer entity**. 6.3.113.2.4 Effect of receipt On receipt of this primitive, the MLME |
| 713 | 52 | 6.3.113.2.4 Effect of receipt | the **peer entity**. 6.3.113.3 MLME-S1GRELAYACTIVATE.confirm 6.3.113.3.1 Function This primitive reports the result of a relay |
| 714 | 26 | 6.3.113.3.3 When generated | the **peer entity**. 6.3.113.3.4 Effect of receipt On receipt of this primitive, the SME |
| 714 | 39 | 6.3.113.4.1 Function | a **peer entity**. 6.3.113.4.2 Semantics of the service primitive The primitive parameters are |
| 715 | 39 | 6.3.113.5.3 When generated | a **peer entity** to convey relay activation information. 6.3.113.5.4 Effect of receipt On receipt |
| 715 | 46 | 6.3.113.5.4 Effect of receipt | the **peer entity** indicated by the PeerSTAAddress parameter. 6.3.114 DCS procedure 6.3.114.1 General This subclause |

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 477 in <this document>, which identify the specific locations where “peer entity” is to be changed to “peer STA”.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 546Michael MONTEMURRO10.41745.5 | "shall be capable of" is really just "shall support". Shall support is used more frequently (95 vs 28) in the draft and describes the requirment better. | Change all occurences of “shall be capable of” to “shall support”.Change “shall be capable of” to “shall support” at 1745.5, 1745.10, 1780.6, 1782.27, 1782.29, 1782.32, 1907.11, 1907.21, 1910.64, 1911.6, 1949.28, 1949.29, 1949.30, 2146.56, 2227.20, 2363.56, 2396.39, 2853.58, 2883.11, 2939.14, 2944.61, 2944.63, 1945.3, 2952.15, 3109.15, 4350.52, 4350.54At 2169.20, change “shall be capable of supporting a delivery” to “shall support a delivery” |

Discussion:

The comment is accurate. The proposed changes are (modulo one typo):

|  |  |  |  |
| --- | --- | --- | --- |
| Page | Line | Subclause | Context |
| 1745 | 4 | 10.5 MSDU and MMPDU defragmentation | STA **shall be capable of** receiving |
| 1745 | 9 | 10.5 MSDU and MMPDU defragmentation | STA **shall be capable of** receiving |
| 1780 | 6 | 10.11 A-MSDU operation | support, **shall be capable of** receiving at |
| 1782 | 27 | 10.12.2 A-MPDU length limit rules | STA **shall be capable of** receiving |
| 1782 | 28 | 10.12.2 A-MPDU length limit rules | STA **shall be capable of** receiving |
| 1782 | 32 | 10.12.2 A-MPDU length limit rules | 1 **shall be capable of** receiving |
| 1907 | 11 | 10.34.2 HT transmit beamforming with implicit feedback | **Shall be capable of** receiving |
| 1907 | 20 | 10.34.2 HT transmit beamforming with implicit feedback | **Shall be capable of** setting |
| 1910 | 64 | 10.34.2.4.2 Calibration capabilities | **shall be capable of** transmitting |
| 1911 | 6 | 10.34.2.4.3 Sounding exchange for calibration | request **shall be capable of** receiving |
| 1949 | 28 | 10.39.4 DTI transmission rules | STA **shall be capable of** processing |
| 1949 | 29 | 10.39.4 DTI transmission rules | STA **shall be capable of** processing |
| 1949 | 30 | 10.39.4 DTI transmission rules | **shall be capable of** processing |
| 2146 | 56 | 11.1.4.5 Synchronizing with a BSS | STA **shall be capable of** transmitting |
| 2169 | 20 | 11.2.3.14.3 FMS Request procedures | FMS **shall be capable of** supporting |
| 2227 | 20 | 11.4.4.5 TS renegotiation | HC **shall be capable of** receiving |
| 2363 | 56 | 11.21.15 Channel usage procedures | STA **shall be capable of** acting |
| 2396 | 39 | 11.22.3.3.2 Query list procedure | true **shall be capable of** using |
| 2853 | 57 | 15.4.5.4 Transmit power level control | mW **shall be capable of** switching |
| 2883 | 10 | 16.3.7.3 Transmit power level control | mW **shall be capable of** switching |
| 2939 | 14 | 18.1.3 Operational modes | ERP **shall be capable of** sending |
| 2944 | 61 | 18.3.5 PHY receive procedure | receiver **shall be capable of** receiving 1, |
| 2944 | 63 | 18.3.5 PHY receive procedure | and **shall be capable of** receiving the ERP-OFDM modulations at rates of 6, 12, |
| 2945 (typo) | 3 | 18.4 ERP operating specifications (general) | receiver **shall be capable of** detecting |
| 2952 | 15 | 19.1.1 Introduction to the HT PHY | **shall be capable of** transmitting |
| 3109 | 15 | 21.1.1 Introduction to the VHT PHY | **shall be capable of** transmitting |
| 4350 | 52 | E.2.2 3650–3700 MHz band in the United States | STA **shall be capable of** receiving |
| 4350 | 55 | E.2.2 3650–3700 MHz band in the United States | STA **shall be capable of** transmitting |

Proposed resolution:

REVISED

Change “shall be capable of” to “shall support” and “Shall be capable of” to “Shall support” in the locations shown in the table shown under “Discussion” for CID 546 in <this document>.

At 2169.20 delete “supporting” (leaving, after the changes above, “A STA that supports FMS shall support a delivery interval of 1 for any stream.”).

Note to the commenter: this is the proposed changes (modulo case-sensitivity), with one page number typo fixed.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 193Mark RISON12.5.3.4.12577.30 | "The decryption processing prevents replay of MPDUs by validating that the PN in theMPDU is greater than the replay counter maintained for the session." should be ... "and TID" or "and AC" to allow for reordering across TIDs/ACs, where the receiver has indicated support for multiple replay counters (so maybe better as "(or the replay counter corresponding to the TID, if more than one replay counter is supported)" Or actually, since in 802.11 (unlike WMM) you allocate TIDs up to the max num replay counters supported, maybe just say "replay counter maintained for the session for that TID"?) | Change to "The decryption processing prevents replay of MPDUs by validating that the PN in theMPDU is greater than the replay counter maintained for the session for that TID." Ditto at 2587.36. At 2577.49 change "the PN in the CCMPheader is greater than the replay counter maintained for the session and TID/ACI." to "the PN in the MPDU is greater than the replay counter maintained for the session for that TID." |

Discussion:

As the comment points out, in 802.11 (as opposed to WMM) for Data frames you use TIDs up to the max number of replay counters supported, at which point you can’t use any other TIDs for that SA (though this is clearer for CCMP than for GCMP):

**12.5.3.3.7 CCM originator processing**

A transmitter shall not use an IEEE 802.11 MSDU or A-MSDU priority if this would cause the total number of priorities used during the lifetime of the SA to exceed the number of replay counters supported by the receiver (for a pairwise SA) or all the receivers (for a group SA) for that SA. 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 802.11 MSDU or A-MSDU priority.

**12.5.5.3.6 GCM originator processing**

A transmitter shall not use IEEE 802.11 MSDU or A-MSDU priorities without ensuring that the receiver supports the required number of replay counters. The transmitter shall not reorder GCMP 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 802.11 MSDU or A-MSDU priority.

The number of replay counters supported for each PTKSA and GTKSA is indicated in the RSN Capabilities field of the RSNE: it can be 1, 2, 4 or 16 (see Table 9-152—PTKSA/GTKSA replay counters usage). Therefore as pointed out in the comment, the expression “the replay counter maintained for the session” only works if the receiver is only capable of 1 RC per SA and requires further qualification otherwise (this is already done for PV1 (in 12.5.3.4.1.b).6)) but not for PV0 (whether CCMP or GCMP)).

In addition to this, there are replay counters for MFP (one for each of individually addressed robust Management and individually addressed robust PV1 Management) and for QMF (one for each ACI of individually addressed robust Management -- it’s not clear whether this is also used for individually addressed robust PV1 management or whether there’s a separate set of 4), when enabled:

**12.5.3.4.4 PN and replay detection**

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 a single replay counter for received individually addressed robust PV1 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 PV1 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.

It would appear that the MFP/QMF counters are not subject to any maximum per SA limit, i.e. there are always 1/4 of them respectively per SA, irrespective of the number of replay counters support signalled in the RSN Capabilities field of the RSNE.

In addition to this, there are a bazillion replay counters for group addressed robust Management frames under MFP and under QMF, when enabled; see 12.5.4.6 BIP reception.

So the replay counters are per-ACI for QMFs under MFP, per-TID for Data frames (but up to maximum total number per SA) and a singleton for non-QMF robust Management frames under MFP and (if individually addressed) not sent to an AP.

Also note the subtle “the PN in the CCMP header” for PV1, cf. “the PN in the MPDU” for PV0. This is because for PV1 the CCMP header is constructed locally, not actually carried in the MPDU:

**12.5.3.2 CCMP MPDU format**

The CCMP header is not included in secure PV1 MPDUs, but constructed locally at the STA as defined in 12.5.3.3.6 (Construct CCMP header for PV1 MPDUs).

**12.5.3.3.6 Construct CCMP header for PV1 MPDUs**

The CCMP header is not present in secure PV1 MPDUs, but constructed locally at the STA as follows:

The proposed change from “in the CCMP header” to “in the MPDU” missed this point; it would be helpful to be more explicit about this.

Note PV1 is not supported with GCMP, so we don’t need to worry about that. [Having said that, I can’t find the specific statement in the spec!]

Proposed changes:

Change 12.5.5.3.6 GCM originator processing as follows:

A transmitter shall not use an IEEE 802.11 MSDU or A-MSDU priorit~~ies without ensuring that the receiver supports the required number of replay counters~~y if this would cause the total number of priorities used during the lifetime of the SA to exceed the number of replay counters supported by the receiver (for a pairwise SA) or all the receivers (for a group SA) for that SA. The transmitter shall not reorder GCMP 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 802.11 MSDU or A-MSDU priority.

Change 12.5.3.4.1 General (under 12.5.3.4 CCMP decapsulation) as follows:

5) 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, and TID (for Data frames) or ACI (for QMFs).

 […]

6) The decryption processing prevents replay of MPDUs by validating that the PN in the locally constructed CCMP header (see 12.5.3.3.6) is greater than the replay counter maintained for the session, and TID~~/~~ (for Data frames) or ACI (for QMFs).

Change 12.5.5.4.1 General (under 12.5.5.4 GCMP decapsulation) as follows:

e) 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, and TID (for Data frames) or ACI (for QMFs).

Change “TID/ACI” to “TID (for Data frames) or ACI (for QMFs)” at:

* 2571.47 (“1) When the sequence number of the MPDU is less than the previous sequence number and satisfies the BPN update conditions in 12.5.3.3.6 (Construct CCMP header for PV1 MPDUs) for that **TID/ACI**, increment the base PN so that the PN never repeats for the same temporal key and TID/ACI.” in 12.5.3.3(.1) CCMP cryptographic encapsulation)
* 2572.37 (“For PV1 MPDUs, the PN shall never repeat for a series of encrypted MPDUs using the same temporal key and **TID/ACI**.” in 12.5.3.3.2 PN processing)
* 2575.59 (“The locally stored BPN shall be incremented by 1 when the sequence number of the MPDU is less than the previous sequence number for that **TID/ACI** if any of the following two conditions is satisfied:” in 12.5.3.3.6 Construct CCMP header for PV1 MPDUs)

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 193 in <this document>, which address the issues raised by the commenter. Note to the commenter: there is no PN in PV1 MPDUs; instead the PN is generated locally.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 171Mark RISON12.5.3.4.4 | It is not clear what replay counter to use for non-QoS Data frames. 5.1.1.3 suggests TID 0: "At QoS STAs associated in a QoS BSS, MSDUs with a priority of Contention are considered equivalent to MSDUs with TID 0." (this is perhaps tx not rx though) | At the end of item b) in 12.5.3.4.4 PN and replay detection and 12.5.5.4.4 PN and replay detection add "For the purposes of replay detection Data frames that have the QoS subfield of the Subtype subfield equal to 0 are treated as having TID 0.". At the start of each subclause change "To effect replay detection," to "To effect replay detection for (QoS) Data frames," |

Discussion:

12.5.3.4.4 does not explicitly cover replay counters for non-QoS Data frames:

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 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.

b) For each PTKSA, GTKSA, and protocol version value, 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 9.4.2.24 (RSNE)), and shall use the PN from a received frame to detect replayed frames. A replayed frame occurs when the PN 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.

[…]

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.

However, d) (but not b)) is using the concept of a “priority value” for the MPDU, which per CID 212 is actually referring to 12.5.3.3.1 General. For PV0 this is:

If the Type field of the Frame Control field is 10 (Data frame) and there is a QoS Control field present in the MPDU header, the priority value of the MPDU is equal to the value of the TID subfield of the QoS Control field (bits 0 to 3 of the QoS Control field). If the Type field of the Frame Control field is 00 (Management frame) and the frame is a QMF, the priority value of the MPDU is equal to the value in the ACI subfield of the Sequence Number field. Otherwise, the priority value of the MPDU is equal to the fixed value 0.

and for PV1 this is:

If the MPDU is a QoS Data MPDU, the priority value of the MPDU is equal to the value of the PTID subfield 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 value of the MPDU is equal to the value in the ACI subfield of the Sequence Number field. Otherwise, the priority value of the MPDU is equal to the fixed value 0.

Note the former does yield 0 for (PV0) non-QoS Data frames. Also note that the reference in b) to the frame type is bogus, since b) is only about Data frames (c) is for robust Management frames).

Having said that, we also have to account for replay counters for TIDs above 7, a.k.a. TSIDs, used by HCCA, SPCA, HEMM and last but by no means least SEMM (see Table 9-12—TID subfield).

Also, “To effect replay detection, the receiver extracts the PN from the CCMP header.” is not specific enough. It’s not so extracted for groupcast robust Management frames, which are done under BIP not CCMP (the proposed change is wrong here; this subclause also covers individually addressed robust Management frames too).

Finally, it has been pointed out that things like “shall maintain a separate set of replay counters” is untestable as it stands (and it is not clear what these are separate to).

Proposed changes:

Change 12.5.3.4.4 PN and replay detection as follows:

To effect replay detection for Data or individually addressed robust Management frames (see 12.5.4.6 (BIP reception) for group addressed robust Management and protected Beacon frames), the receiver extracts the PN from the CCMP header.

[…]

The following processing rules ~~are~~shall be used to detect replay:

a) The receiver ~~shall~~ maintains a ~~separate~~ set of replay counters for each PTKSA, GTKSA, and protocol version value. The receiver shall initialize~~s~~ these replay counters to 0 when it (re)sets the corresponding temporal key ~~for a peer~~. The replay counter ~~is~~shall be set to the PN ~~value~~ of ~~accepted~~the CCMP MPDU~~s~~ that was most recently accepted per the rules below.

b) For each PTKSA, GTKSA, and protocol version value, the recipient shall maintain a separate replay counter for each ~~TID~~MPDU priority value (see 12.5.3.3.1) and TSID, subject to the limitation of the number of supported replay counters indicated in the RSN Capabilities field (see 9.4.2.24 (RSNE)), and shall use the PN from a received frame to detect replayed frames. A replayed frame occurs when the PN from a received frame is less than or equal to the current replay counter ~~value~~ for the ~~frame’s MSDU or A-MSDU~~MPDU priority value or TSID ~~and frame type~~.

[…]

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 MPDU priority value or TSID ~~of the received MPDU~~. The receiver shall discard MSDUs and MMPDUs whose constituent MPDU PN ~~value~~s are not incrementing in steps of 1.

Change 12.5.3.3.7 CCM originator processing as follows:

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

Change 12.5.5.4.4 PN and replay detection as follows:

To effect replay detection for Data or individually addressed robust Management frames (see 12.5.4.6 (BIP reception) for group addressed robust Management and protected Beacon frames), the receiver extracts the PN from the GCMP header. […] The following processing rules ~~are~~shall be used to detect replay:

a) The receiver ~~shall~~ maintains a ~~separate~~ set of replay counters for each PTKSA and GTKSA. The receiver shall initialize~~s~~ these replay counters to 0 when it (re)sets the corresponding temporal key ~~for a peer~~. The replay counter ~~is~~shall be set to the PN ~~value~~ of ~~accepted~~the GCMP MPDU~~s~~ that was most recently accepted per the rules below.

b) For each PTKSA and GTKSA, the recipient shall maintain a separate replay counter for each ~~TID~~MPDU priority value (see 12.5.3.3.1) and TSID, subject to the limitation of the number of supported replay counters indicated in the RSN Capabilities field (see 9.4.2.24 (RSNE)), and shall use the PN from a received frame to detect replayed frames. A replayed frame occurs when the PN from a received frame is less than or equal to the current replay counter ~~value~~ for the ~~frame’s MSDU or A-MSDU~~MPDU priority value or TSID ~~and frame type~~.

[…]

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 MPDU priority value or TSID ~~of the received MPDU~~. The receiver shall discard MSDUs and MMPDUs whose constituent MPDU PN ~~value~~s are not incrementing in steps of 1.

Change 12.5.5.3.6 GCM originator processing as follows:

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

Change “replay counter value”[s] to “replay counter”[s] in 6.3.19.1.4 Effect of receipt, 12.5.2.6 TKIP replay protection procedures, 12.5.4.6 BIP reception (3x), 12.7.4 EAPOL-Key frame notation (2x), 14.6.3 Mesh Group Key Inform frame construction and processing.

Change “Replay counter value” to “Value” in Table 9-152—PTKSA/GTKSA replay counters usage.

Change “shall be implemented as a” to “shall be a” in 12.5.2.6 TKIP replay protection procedures, 12.5.4.4 BIP replay protection (2x).

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 171 in <this document>, which address the issue raised by the commenter, making use of the MPDU priority value defined in 12.5.3.3.1.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 204Mark RISON11.3 | There are 12 locations where the SME is required to "delete any PTKSA, GTKSA, IGTKSA, BIGTKSA and temporal keys held for communication". Any TPKSA also needs to be deleted too (except maybe for reassociation to the same AP?) | Add ", TPKSA" after ", GTKSA" in each of the 12 locations |
| CID 205Mark RISON11.3 | There are 14 locations where the SME is required to "delete any PTKSA, GTKSA, IGTKSA[, BIGTKSA] and temporal keys held for communication". But the temporal keys associated with an SA will be deleted if the SA is deleted, since they are part of the SA | Delete "and temporal keys" in each of the 14 locations, and replace the preceding comma with an "and" |

Discussion:

Example occurrences:

a) If the STA is in an IBSS, the SME shall delete any PTKSA, GTKSA, IGTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)).

The SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA and temporal keys held for communication with the AP or PCP by using MLME-DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)) before invoking MLME-ASSOCIATE.request primitive.

The TPKSA is not special: it too needs to go when the other SAs go. Having said that, there can be no TPKSA in an IBSS, so no change is needed in IBSS-only contexts (this is why there are only 12 hits for CID 204 and 14 for CID 205). Those locations are missing BIGTKSA, though.

As regards reassoc to the same AP, this is explicitly covered in 11.3.5.4 Non-AP and non-PCP STA reassociation initiation procedures:

If the MLME-REASSOCIATION.request primitive has the new AP’s or PCP’s MAC address in the CurrentAPAddress parameter (reassociation to the same AP or PCP), the following states, agreements, and allocations shall be deleted or reset to initial values:

[…]

10) TPKSAs established with any peers

Indeed, the temporal keys are part of the SA (see 12.6.1.1/12.6.1.2) so disappear with it.

Proposed changes:

Change “delete any PTKSA, GTKSA, IGTKSA and temporal keys” to “delete any PTKSA, GTKSA, IGTKSA and BIGTKSA” in 11.3.4.2 Authentication—originating STA, 11.3.4.3 Authentication—destination STA.

Change “delete any PTKSA, GTKSA, IGTKSA, BIGTKSA and temporal keys” to “delete any PTKSA, GTKSA, IGTKSA, BIGTKSA and TPKSA” in 11.3.4.4 Deauthentication—originating STA, 11.3.4.5 Deauthentication—destination STA, 11.3.5.2 Non-AP and non-PCP STA association initiation procedures, 11.3.5.3 AP or PCP association receipt procedures, 11.3.5.4 Non-AP and non-PCP STA reassociation initiation procedures, 11.3.5.5 AP or PCP reassociation receipt procedures, 11.3.5.6 Non-AP and non-PCP STA disassociation initiation procedures, 11.3.5.7 Non-AP and non-PCP STA disassociation receipt procedure, 11.3.5.8 AP or PCP disassociation initiation procedure, 11.3.5.9 AP or PCP disassociation receipt procedure.

Change “a PTKSA, GTKSA, IGTKSA, or BIGTKSA” to “a PTKSA, GTKSA, IGTKSA, BIGTKSA or TPKSA” in 12.6.1.3.2 Security association in an ESS.

Change “PTKSA, IGTKSA, GTKSA or BIGTKSA” to “PTKSA, IGTKSA, GTKSA, BIGTKSA or TPKSA” (3x) in 12.6.21 RSNA rekeying.

Delete “and temporal keys” in 11.13 SA Query procedures.

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 204 and 205 in <this document>, which add TPKSA or BIGTKSA where missing and delete “and temporal keys” where redundant.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 344Mark RISON | "direct hashing" (technique) should be referred to as "hash-to-element" (11x) so a single term is used throughout. Ditto "directly hashing" should be canonicalised (2x) | As it says in the comment |

Discussion:

The spec has now gone through the 5000-page mark. We don’t need to make it even harder to find stuff by using multiple terms for the same thing.

The resolution to CID 355 (in 21/0716r2) claims that it’s to be described as the hash-to-element method rather than as the hash-to-element technique (though in D0.0 there are 8 instances of “hashing technique” or “element technique” and only 2 instances of “hashing method” or “element method”).

Proposed resolution:

REVISED

Change “direct hashing” to “the hash-to-element method” in Table 9-50—Status codes.

Change “the direct hashing to element technique” to “the hash-to-element method” in Table 9-93—BSS membership selector value encoding.

Change “directly hashing” to “the hash-to-element method” in Table 9-321—Extended RSN Capabilities field.

Change “the direct hashing technique” to “the hash-to-element method” in 12.4.2 Assumptions on SAE.

Change “support for direct hashing” to “support for the hash-to-element method”; “The direct hashing technique” to “The hash-to-element method” in 12.4.4.2.3 Hash-to-curve generation of the password element with ECC groups.

Change “for direct hashing” to “for the hash-to-element method”; “he direct hashing technique” to “he hash-to-element method” (3x) in 12.4.4.3.3 Direct generation of the password element with FFC groups.

Change “directly hashing to a group element” to “the hash-to-element method” in 12.4.5.2 PWE and secret generation.

Change “the direct hashing technique” to “the hash-to-element method” in 12.4.5.4 Processing of a peer’s SAE Commit message (2x).

Change “technique” to “method” in 12.4.2 Assumptions on SAE (first instance), 12.4.4.2.3 Hash-to-curve generation of the password element with ECC groups (first 3 instances), 12.4.4.3.3 Direct generation of the password element with FFC groups (2nd, 3rd and 4th instances).

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 340Mark RISON9.4.2.24.1 | It would be good if some of the RSNE examples showed counts greater than 1 | As it says in the comment |

Discussion:

The RSNE is extremely flexible, with various fields that can be present 0, 1 or more times. Not taking account of this can result and has resulted in interop issues with peers that send (valid) RSNEs with counts that are >1.

Proposed changes:

Change the last example as follows:

IEEE 802.1X or PSK authentication, CCMP-128 pairwise and group data cipher suites (WEP-40, WEP-104, and TKIP are not allowed), GCMP-128 alternative pairwise data cipher suite, two PMKIDs and management frame protection with BIP-CMAC-128 ~~as the~~ group management cipher suite ~~selector~~.

30, // element id, 48 expressed as hexadecimal value

~~1A~~42, // length in octets, ~~26~~66 expressed as hexadecimal value

01 00, // Version 1

00 0F AC 04, // CCMP-128 as group data cipher suite

0~~1~~2 00, // pairwise cipher suite count

00 0F AC 04, // CCMP-128 as pairwise cipher suite

00 0F AC 08, // GCMP-128 as pairwise cipher suite

0~~1~~2 00, // authentication count

00 0F AC 01 // IEEE 802.1X authentication

00 0F AC 02 // PSK authentication

80 00 // management frame protection is enabled but not required

0~~0~~2 00 // ~~No~~ PMKID~~s~~ count

01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 // PMKID

11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 // PMKID

00 0F AC 06~~,~~ // BIP-CMAC-128 as ~~the broadcast/multicast~~group management cipher suite

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID 340 in <this document>, which make the last example into a Christmas tree.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID xxxMark RISON |  |  |

Discussion:

Proposed changes:

Proposed resolution:

REVISED

Make the changes shown under “Proposed changes” for CID xxx in <this document>, which xxx.

**References:**

802.11me/D0.0 except where otherwise specified