IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Combined Proposal | | | | |
| Date: 2022-03-08 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | Email |
| Mark Hamilton | Ruckus/CommScope |  |  | mark.hamilton2152@gmail.com |
|  |  |  |  |  |

Abstract

This document proposes a P802.11bh mechanism for a non-AP STA to be optionally assigned an identifier, generated by the network (ESS), or by the device. The contents of the arbitrary identifier identifier is outside the scope of this standard. Though, an informative example for a network-gerneated identifier following the design presented in 154r0 could be described in the amendment. This proposal also adds an optional mechanism for the network to supply a MAC address that the non-AP can use on a future association, to be recognized pre- and during association processing.

This document is based on 187r1, 158r3 and 427r1. All technical credit for those proposals goes to their authors.

# Discussion

This proposal combines the “network generated device ID” of 11-22/0187r1, the “STA generated device ID” of 11-22/0158r3, and the “MAAD MAC 2” of 11-22/0427r1.

Discussion/introduction (and potential “ToDo”/discussion comments) of each of the above can be found in their respective submissions.

# Proposed text changes

*Note to editor: Text changes are shown against REVme/D1.0.*

*Add following Acronym to 3.4.*

MAAD MAC Address Designation

* Association Request frame format

*Add new rows into Table 9-62 (Association Request frame body) as shown below:*

|  |  |  |
| --- | --- | --- |
| * Association Request frame body | | |
| Order | Information | Notes |
| ... | ... | ... |
| 57 | WUR Mode | The WUR Mode element is optionally present when dot11WUROptionImplemented is true; otherwise, it is not present. |
| 58 | Device ID | The Device ID element is optionally present when using FILS authentication; otherwise, it is not present. |
| 59 | MAAD | The MAAD element is optionally present when using FILS authentication; otherwise it is not present |
| Last | Vendor Specific | One or more Vendor Specific elements are optionally present. These elements follow all other elements. |

* Association Response frame format

*Add new rows into Table 9-63 (Association Response frame body) as shown below:*

|  |  |  |
| --- | --- | --- |
| * Association Response frame body | | |
| Order | Information | Notes |
| ... | ... | ... |
| 76 | WUR Mode | The WUR Mode element is present when dot11WUROptionImplemented is true, and the WUR Mode element is present in the Association Request frame that solicited this Association Response frame; otherwise it is not present. |
| 77 | Device ID | The Device ID element is optionally present when using FILS authentication; otherwise, it is not present. |
| 78 | MAAD | The MAAD element is optionally present when using FILS authentication; otherwise it is not present |
| Last | Vendor Specific | One or more Vendor Specific elements are optionally present. These elements follow all other elements. |

* Reassociation Request frame format

*Add new rows into Table 9-64 (Reassociation Request frame body) as shown below:*

|  |  |  |
| --- | --- | --- |
| * Reassociation Request frame body | | |
| Order | Information | Notes |
| ... | ... | ... |
| 61 | WUR Mode | The WUR Mode element is optionally present when dot11WUROptionImplemented is true; otherwise, it is not present. |
| 62 | Device ID | The Device ID element is optionally present when using FILS authentication; otherwise, it is not present. |
| 63 | MAAD | The MAAD element is optionally present when using FILS authentication; otherwise it is not present |
| Last | Vendor Specific | One or more Vendor Specific elements are optionally present. These elements follow all other elements. |

* Reassociation Response frame format

*Add new rows into Table 9-65 (Reassociation Response frame body) as shown below:*

|  |  |  |
| --- | --- | --- |
| * Reassociation Response frame body | | |
| Order | Information | Notes |
| ... | ... | ... |
| 79 | WUR Mode | The WUR Mode element is present when dot11WUROptionImplemented is true, and the WUR Mode element is present in the Reassociation Request frame that solicited this Reassociation Response frame; otherwise it is not present. |
| 80 | Device ID | The Device ID element is optionally present when using FILS authentication; otherwise, it is not present. |
| 81 | MAAD | The MAAD element is optionally present when using FILS authentication; otherwise it is not present |
| Last | Vendor Specific | One or more Vendor Specific elements are optionally present. These elements follow all other elements. |

* Elements
* General

*Add new rows into Table 9-128 (Element IDs) as shown below:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * Element IDs | | | | |
| Element | Element ID | Element ID Extension | Extensible | Fragmentable |
| ... | ... | ... | ... | ... |
| Anti-Clogging Token Container  (see 9.4.2.247 (Anti-Clogging Token Container element)) | 255 | 93 | No | No |
| Device ID (see 9.4.2.x (Device ID element) | 255 | <ANA> | No | No |
| MAAD (see 9.4.2.x MAAD element) | 255 | <ANA> | No | No |
| Reserved | 255 | 94–255 |  |  |
| NOTE—See 10.28.6 (Element parsing) on the parsing of elements. | | | | |

* Extended Capabilities element

*Add new rows into Table 9-190 (Extended Capabilities field) as shown below:*

|  |  |  |
| --- | --- | --- |
| * Extended Capabilities field | | |
| Bit | Information | Notes |
| ... | ... | ... |
| 89 | TWT Parameters Range Support | Set to 1 to indicate support for reception of a TWT Setup frame that contains two TWT elements (see 10.47.9 (TWT parameter ranges)); otherwise, set to 0. |
| <ANA> | Network Device ID Support | Set to 1 to indicate support for network-generated Device ID; otherwise, set to 0. |
| <ANA+1> | Client Device ID Support | Set to 1 to indicate support for client-device generated Device ID; otherwise, set to 0. |
| <ANA+2> | MAAD Capability | A STA sets MAAD Capability subfield to 1 to indicate support for MAAD and sets to 0 if MAAD is not supported. |
| 88, 90–*n* | Reserved |  |

9.4.2.296 Device ID element

*Add new subclauses after 9.4.2.295 (i.e., at the end of the 9.4.2 subclauses):*

The Device ID element contains a device identifier. The format of the Device ID element is shown in Figure 9-1002a (Device ID element format).

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Element ID | | Length | | Element ID Extension | | Device ID Type | | ID Blob  (optional) | Device ID TTL  (optional) | Device ID  (optional) |
| Octets: | 1 | | 1 | | 1 | | 1 | | variable | 2 | variable |
|  | |  | |  | |  | |

The Element ID, Length, and Element ID Extension fields are defined in 9.4.2.1 (General).

The Device ID Type field contains the code indicating the contents and status, per Table 9-bbb.

|  |  |  |  |
| --- | --- | --- | --- |
| Table 9-bbb—Device ID Type values | |  | |
| Value | Description | |
| 0 | Success. ID Blob, Device ID TTL and Device ID fields are not present. | |
| 1 | Network-generated Device ID provided. ID Blob field is present. Device ID TTL and Device ID fields are not present | |
| 2 | Client-device generated ID provided. ID Blob field is not present. Device ID TTL and Device ID fields are present. | |
| 255 | Unspecified failure. ID Blob, Device ID TTL and Device ID fields are not present. | |

The ID Blob field contains an opaque identifier from an AP in the ESS, and is present for network-generated Device ID.

The Device ID TTL field is present for client-device generated Device ID, and indicates how long the Device ID is going to remain valid using values defined in Table 9-aaa (Device ID TTL values).

NOTE—Device ID does not change during an ESS association even if its indicated TTL expires.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table 9-aaa—Device ID TTL values | |  | | |
| Value | Description | | |
| 0 | Duration of this ESS association | | |
| 1-65000 | Value times 10 minutes (e.g., 144 indicates one day) | | |
| 65001-65532 | Reserved | | |
| 65533 | Not specified | | |
| 65534 | Indefinitely | | |
| 65535 | Vendor specific duration indicated using mechanisms outside the scope of this standard. | | |
|  | | |

The Device ID field corresponds to the arbitrary identifier for the STA sending this element, and is present for client-device generated Device ID.

**9.4.2.297 MAAD element**

The MAAD element contains a MAAD MAC address. The format of the MAAD element is shown in Figure 9-y.

|  |  |  |  |
| --- | --- | --- | --- |
| Element ID | Length | Element ID Extension | MAAD MAC |

Octets 1 1 1 6

**Figure 9-y MAAD element**

The Element ID, Length, and Element ID Extension fields are defined in 9.4.2.1 (General).

The MAAD MAC field is a 48-bit MAC address.

*Add a new subclause at the end of clause 11 (MLME)*

**11.xx MAC Address Designation (MAAD) operation**

**11.xx.1 General**

To mitigate tracking and traffic analysis, a non-AP STA may randomly change its MAC address (see 4.5.4.10). For some services, however, it may be desirable to the user that the non-AP STA is identified by the AP and network services. MAAD operation enables a non-AP STA to use a random MAC address that is designated by the AP/ESS, and therefore the non-AP STA is identifiable by the AP/ESS whilst being unidentifiable to a third party.

A STA advertises support for MAAD by setting the MAAD Capability subfield to 1 in the Extended Capabilites element in Probe Response, Association Response and Reassociation Response frames.

Each time the non-AP STA associates to the AP/ESS, it receives a new MAAD MAC address during the RSN association. The non-AP STA may then use that MAAD MAC address as its TA the next time it probes or requests association to that same AP/ESS.

When the associating non-AP STA advertises support for MAAD, the AP shall allocate a new MAAD MAC address to the non-AP STA by including a MAAD KDE in message 3 of the 4-way handshake or, when using FILS authentication, including the MAAD element in the Association Response frame.

The non-AP STA should store that newly allocated MAAD MAC as an identifier for that AP/ESS. The non-AP STA then may use that allocated MAAD MAC address as its TA when it again associates or reassociates to that same AP or ESS. In so doing, the AP/ESS will identify the non-AP STA.

Note 1: Allocating a new MAAD MAC during each association ensures that the non-AP STA will use a different TA for each association and hence that non-AP STA is unidentifiable to a third party.

**11.xx.2 MAAD MAC address**

The MAAD MAC addressis a 48-bit address that is constructed from the locally administered address space (see 12.2.10). The non-AP STA may then store this address and use it as the TA in the next association request to that same AP.

An AP should generate the MAAD MAC addresses on a random basis such that a returning non-AP STA cannot be identified by a third party from the TA it is using. Allocating random 48 bit addresses should suffice but an AP may embed bits into the addresses in order to categorize or aid recognition. The generation of the MAAD MAC address is out-of–scope.

**11.xx.3 Stored MAAD MAC addresses**

A list of MAAD MACs and respective non-AP STAs shall be stored by the AP and used as an identifier for each non-AP STA. A non-AP STA may store the latest MAAD MAC received from a particular AP such that the next time the non-AP STA associates to that AP, the AP can identify the non-AP STA.

The AP may determine further information or IDs about an associated non-AP STA such as membership number, guest information, family member, subscription, etc. The gathering and determination of such IDs is out-of-scope.

**11.xx.3 Pre-Association with MAAD MAC address**

A non-AP STA that has been allocated a MAAD MAC address, may use that address when directly probing the AP or ESS that allocated that address such that the AP may identify the non-AP STA and note that the particular non-AP STA is within range of the WM.

When a non-AP STA sends an Association Request using an allocated MAAD MAC address as the TA, to the AP that allocated that address, then that AP may identify the non-AP STA before association is started or completed.

12.2.11 Device ID indication

*Add a new subclause after 12.2.10 (i.e., immediately before 12.3):*

A non-AP STA may opt-in to providing an idenfitier, or using an identifier provided by an AP, to an AP when establishing an ESS association using RSN.

An AP may provide an identifier to a non-AP STA and the non-AP STA may opt-in to providing that identifier to any AP in the same ESS to allow the network to recognize the same non-AP STA when it returns to the ESS even if it changes its MAC address.

A non-AP STA may provide its persistent or semi-persistent identifier to the AP when establishing an ESS association using RSN. For some use cases, this identifier might be the globally unique MAC address of the STA. For some other cases, it might be a random value generated for connections to the specific ESS. For the identifier to be useful, the value should be selected in a manner that is likely to result in a unique value between the STAs connected to the ESS at the same time.

The non-AP STA may send the Device ID element or the Device ID KDE if the AP indicates support for Device ID in the Extended Capabilities element. Otherwise, the non-AP STA shall not send the Device ID element or the Device ID KDE.

NOTE—A globally unique MAC address might be used to track the non-AP STA between ESSs and as such, might not be an appropriate choice for some use cases.

When using FILS authentication, the non-AP STA sends the identifier, if it has one and opts-in to using it, in the Association Request frame and the AP may send a new network-generated identifier in the Association Response frame. When using FT, the non-AP STA sends the identifier, if it has one and opts-in to using it, during the initial mobility domain association the EAPOL-Key msg 2/4 and the AP may send a new network-generated identifier in the EAPOL-Key msg 3/4; the identifier or a new identifier are not exchanged during the FT protocol reassociations within the same ESS. For other cases, the non-AP STA sends the identifier, if it has one and opts-in to using it, during the initial 4-way handshake in the EAPOL-Key msg 2/4 and the AP may send a new network-generated identifier in the EAPOL-Key msg 3/4. When the non-AP STA sends a network-generated identifier, it shall send the most recently received value from an AP in the ESS without modification.

Alternatively, the AP or non-AP STA may provide a status response of Success or Unspecified Failure in response to an identifier provided by its peer.

*Add a new subclause at the end of 12.2 subclauses*

**12.2.z MAAD MAC**

An AP may provide a MAAD MAC address to a non-AP STA and the non-AP STA may use this MAAD MAC address as its TA when it returns to the ESS so as to allow the network to recognize the same non-AP STA.

The provision of this MAAD MAC address is protected from third parties to limit the tracking capability to the APs in an ESS. The non-AP STA uses the MMAD MAC address as the TA when it next associates or addresses the ESS. Each time the non-AP STA associates to the ESS, it receives a new MAAD MAC address. A non-AP STA uses a different TA for every association to an ESS and hence cannot be identified by a third party by the TA it is using.

When using FILS authentication, the AP sends a new MAAD MAC address in the Association Response frame. When using FT, the AP sends a new MAAD MAC address in the EAPOL-Key message 3; the new MAAD MAC address is sent during the FT protocol reassociations within the same ESS. For other cases the AP sends a new MAAD MAC address in the EAPOL-Key message 3.

* EAPOL-Key frames

*Add a new row into Table 12-10 (KDE selectors) as shown below:*

|  |  |  |  |
| --- | --- | --- | --- |
| * KDE selectors | | | |
| OUI | | Data type | Meaning |
| 00-0F-AC | | 0 | Reserved |
| ... | | ... | ... |
| 00-0F-AC | | 15 | WIGTK KDE |
| 00-0F-AC | | <ANA> | Device ID KDE |
| 00-0F-AC | | <ANA+1> | MAAD KDE |
| 00-0F-AC | | 16–255 | Reserved |
| Other OUI or CID | | Any | Vendor specific |

*Add the following description of the new KDEs at the end of 12.7.2 (P3212 L55) as shown below:*

|  |
| --- |
| * WIGTK KDE |

The WIPN corresponds to the WIPN value that was used for computing the MIC in the last protected broadcast or group addressed WUR Wake-up frame and it is used by the receiver as the initial value for the BIP replay counter for the WIGTK.

The format of the Device ID KDE is shown in Figure 12-48a (Device ID KDE format).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Device ID Type | ID Blob  (optional) | Device ID TTL  (optional) | Device ID  (optional) |
| Octets: | 1 | variable | 2 | variable |

Figure 12-48a—Device ID KDE format

The Device ID Type field contains the code indicating the contents and status, per Table 9-bbb.

The ID Blob field contains an opaque identifier from an AP in the ESS.The Device ID TTL field indicates how long the Device ID is going to remain valid using values defined in Table 9-aaa (Device ID TTL values).

The Device ID corresponds to the arbitrary identifier for the STA sending this KDE.The format of the MAAD KDE is shown in Figure 12-48a (MAAD KDE format).

|  |
| --- |
| MAAD MAC |

Octets 6

Figure 12-48a—MAAD KDE format

The MAAD MAC field contains MAAD MAC address from an AP in the ESS.

* EAPOL-Key frame notation

*Modify 12.7.4 (P3215 L25) as shown below:*

OCI KDE is a KDE containing operating channel information

Device ID KDE is a KDE containing a device identifier

MAAD KDE is a KDE containing a MAAD MAC

RSNXE is described in 9.4.2.241 (RSN Extension element (RSNXE))

PMKID identifies the PMKSA selected by the Authenticator

“{a} or {b}” means that exactly one of either {a} or {b} is present as the {Key Data}

* 4-way handshake
* General

*Modify 12.7.6.1 as shown below:*

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} or {RSNE, Device ID KDE} or {RSNE, OCI KDE, Device ID KDE} or {RSNE, RSNXE, Device ID KDE} or {RSNE, OCI KDE, RSNXE, Device ID KDE})

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} or {RSNE, GTK[N], Device ID KDE} or   
{RSNE, GTK[N], OCI KDE, Device ID KDE} or {RSNE, GTK[N], RSNXE, Device ID KDE} or   
{RSNE, GTK[N], OCI KDE, RSNXE, Device ID KDE})

Message 4: Supplicant ® Authenticator: EAPOL-Key(1,1,0,0,P,0,0,0,MIC,{} or {Device ID KDE}).

* 4-way handshake message 2

*Modify 12.7.6.3 as shown below:*

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 if the Device ID KDE is included, 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.
* Additionally, may include a Device ID KDE.
* 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.
* 4-way handshake message 3

*Modify 12.7.6.4 as shown below:*

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 12-8 (Cipher suite key lengths)

Key Replay Counter = *n+1*

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(11ba), and if WUR frame protection is negotiated, the WIGTK 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.
* Additionally, may include a Device ID KDE.
* Additionally, may include a MAAD KDE.
* 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.
* 4-way handshake message 4

*Modify 12.7.6.5 (P3221 L17 and L32) as shown below:*

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 if the Device ID KDE is included, 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 =

— May include a Device ID KDE

— May include one or more vendor specific KDEs and/or Vendor Specific elements