IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Network generated Device ID | | | | |
| Date: 2022-01-24 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Jouni Malinen | Qualcomm, Inc. |  |  | jouni@qca.qualcomm.com |
|  |  |  |  |  |

Abstract

This document proposes a P802.11bh mechanism for a non-AP STA to be able to optionally provide the AP an arbitrary, network (ESS) generated identifier using existing frames and encryption functionality to protect the exchange from third parties. The arbitrary identifier is generated by the network (APs in the ESS) and the contents of that identifier is outside the scope of this standard. Though, an informative example following the design presented in 154r0 could be described in the amendment.

This document is based on 158r0 with the changes done to replace the non-AP STA generated identifier with a network generated one following the design presented in 154r0.

TODO:

* consider replacing Extended Capabilities bit with an RSNXE bit
* consider adding informative description of an example approach for the network (APs) to generate and update the blob (shared key, symmetric crypto, etc. as presented in 154r0)
* consider renaming some elements/fields/etc.

# Discussion

This mechanism assumes that RSN is used. If authentication is not needed, OWE can be used to allow this mechanism to be used with RSN. The non-AP STA is in control of when it opts-in to accepting and sending the identifier provided by an AP to other APs in the same ESS.

When using FILS authentication, the identifier is sent in the Association Request frame and a new identifier is received in Association Response frame. When using FT, the identifier is sent during the initial mobility domain association EAPOL-Key msg 2/4 and a new identifier is received in EAPOL-Key msg 3/4, but not during the FT protocol reassociations within the same ESS. For other cases, the identifier is sent during the initial 4-way handshake in EAPOL-Key msg 2/4 and a new identifier is received in EAPOL-Key msg 3/4. It is assumed that the identifier does not change during an ESS association and as such, there is no need to send a new identifier when reassociating within the ESS (which is also maintaining the same MAC address) or when rekeying PTK.

The new extended capability bit is used to advertise whether the STA supports this mechanism in the ESS. This allows the AP to not generate new identifiers unnecessarily if the non-AP STA is not going to be using them. This can also help with potential interoperability issues with deployed APs that might not be ready for the EAPOL-Key msg 2/4 to include an encrypted Key Data field.

The maximum length of the identifier blob is currently limited implicitly by the IE and KDE constraints to 254 octets. An explicit shorter limit could be considered, if desired.

# Proposed text changes

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

* Association Request frame format

*Add a new row 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. |
| Last | Vendor Specific | One or more Vendor Specific elements are optionally present. These elements follow all other elements. |

* Association Response frame format

*Add a new row 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. |
| Last | Vendor Specific | One or more Vendor Specific elements are optionally present. These elements follow all other elements. |

* Reassociation Request frame format

*Add a new row 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. |
| Last | Vendor Specific | One or more Vendor Specific elements are optionally present. These elements follow all other elements. |

* Reassociation Response frame format

*Add a new row 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. |
| Last | Vendor Specific | One or more Vendor Specific elements are optionally present. These elements follow all other elements. |

* Elements
* General

*Add a new row 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 |
| Reserved | 255 | 94–255 |  |  |
| NOTE—See 10.28.6 (Element parsing) on the parsing of elements. | | | | |

* Extended Capabilities element

*Add a new row 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> | Device ID Support | Set to 1 to indicate support for Device ID indication; otherwise, set to 0. |
| 88, 90–*n* | Reserved |  |

9.4.2.296 Device ID element

*Add a new subclause 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 | ID Blob |
| Octets: | 1 | | 1 | 1 | variable |
|  | |

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

The ID Blob field contains an opaque identifier from an AP in the ESS.

12.2.11 Device ID indication

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

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. Exchanges of this identifier information are protected from third parties to limit the tracking capability to the APs in an ESS.

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 sends a new 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 sends a new 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 sends a new identifier in the EAPOL-Key msg 3/4.

* 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 | | 16–255 | Reserved |
| Other OUI or CID | | Any | Vendor specific |

*Add the following description of the new KDE 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).

|  |  |
| --- | --- |
|  | ID Blob |
| Octets: | variable |

Figure 12-48a—Device ID KDE format

The ID Blob field contains an opaque identifier 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

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,{}).

* 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.
* 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.