IEEE P802.11  
Wireless LANs

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | CR for KEK from PASN | | | | | | Date: 2024-01-18 | | | | | | Author(s): | | | | | | Name | Affiliation | Address | Phone | email | | Po-Kai Huang | Intel |  |  | po-kai.huang@intel.com | | Ilan Peer |  |  |  | | Emily Qi |  |  |  | | Ido Ouzieli |  |  |  | | Jouni Malinen | Qualcomm |  |  |  | | Guoqing Li | Meta |  |  |  | | Davide Magrin |  |  |  | |

Abstract

This submission proposes CR for the following comments on P802.11-bh D2.0:

208, 209, 210, 256, 164, 165, 211, 235

**Revision History:**

R0: Initial version.

R1: Editorial change from “is set to 0/1” to “is 0/1”. Typo fix. Add CID 235 and 164 related to CID 210

R2: Add CID 165

R3: Revise CID 235, which is related to CID 211. Add discussion for CID 211. Add details of encryption of KEK for CID 211. Add “PASN” in front of the Encrypted Data element for CID 210.

R4: Revision on CID 211 to follow suggestion from Jouni to use a separate AKM for key wrap algorithm.

R5: Revision on CID 211 to use empty array for AAD of AES-SIV. Add coauthors.

R6: Update discussion of CID 211 for AAD to justify for using empty array and resolution number in resolution box.

R7: Clarified AAD description for AES-SIV. Simpler proposed resolution text for CID 211 and 235.

R8: Accepted changes in R7; CID 211 and 235 are now ready for motion.

Interpretation of a Motion to Adopt

A motion to approve this submission means that the editing instructions and any changed or added material are actioned in the TGbh D2.0 Draft. This introduction is not part of the adopted material.

***Editing instructions formatted like this are intended to be copied into the TGbh D2.0 Draft. (i.e. they are instructions to the 802.11 editor on how to merge the text with the baseline documents).***

***TGbh Editor: Editing instructions preceded by “TGbh Editor” are instructions to the TGbh editor to modify existing material in the TGbh draft. As a result of adopting the changes, the TGbh editor will execute the instructions rather than copy them to the TGbh Draft.***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CID** | **Commenter** | **Clause** | **P.L** | **Comment** | **Proposed Change** | **Resolution** |
| 208 | Po-Kai Huang | 12.13.7 | 39.1 | This condition of not deriving KEK is not correct. When the peer does not support both device ID or IRM, then KEK is also not needed, otherwise, the key location of TK for both sides will be wrong. A better solution seems to just introduce an additional capability bit say "KEK in PASN" then KEK is dervied if both sides set this bit to 1 and otherwise do not derive KEK. This will likley simply the design and prevent complicated condition check to avoid interop issue. | Commenter will submit a contributon for the proposal since description of the change for this in a comment resolution box is difficult. The general direction is to introduce new capability bit in RSNXE called KEK in PASN and derive KEK based on the capability bit set to 1 on both sides rather than feature support, which is likely to have interop issues since the draft also gets it wrong. Then we can have condition like if support IRM or device ID, then set this capability bit to 1. | Revised –  Agree in principle with the commenter.  TGbh editor to make the changes shown in 11-24/0044r3 under all headings that include CID 208 |
| 209 | Po-Kai Huang | 12.13.7 | 39.30 | This condition of not deriving KEK is not correct. When the peer does not support both device ID or IRM, then KEK is also not needed, otherwise, the key location of TK for both sides will be wrong. A better solution seems to just introduce an additional capability bit say "KEK in PASN" then KEK is dervied if both sides set this bit to 1 and otherwise do not derive KEK. This will likley simply the design and prevent complicated condition check to avoid interop issue. | Commenter will submit a contributon for the proposal since description of the change for this in a comment resolution box is difficult. The general direction is to introduce new capability bit in RSNXE called KEK in PASN and derive KEK based on the capability bit set to 1 on both sides rather than feature support, which is likely to have interop issues since the draft also gets it wrong. Then we can have condition like if support IRM or device ID, then set this capability bit to 1. | Revised –  Agree in principle with the commenter.  TGbh editor to make the changes shown in 11-24/0044r3 under all headings that include CID 208 |
| 210 | Po-Kai Huang | 12.2.12.3 | 34.27 | Currently encryption of KEK is independely for each element. This is not a scalable approach. To iillusrtate, the usage of KEK in 4-way handshake is to encrypt an entire Key Data field with multiple KDE only once rather than do each KDE independently, which is not saclable. see "If the Encrypted Key Data subfield (of the Key Information field) is 1, the entire Key Data field shall be encrypted." As a result, to align with the usage of KEK in 4-way, we should define a Encrypted Data element and define device ID subelement and IRM subelement to be included in the encrypted data element to be encrypted by KEK. This approach then will aligns with the processes of using KEK in 4-way. This will also align inclusion of vendor specific subelement for whatever important information that needs to be encrypted. | Commenter will submit a contributon for the proposal since description of the change for this in a comment resolution box is difficult. The general direction is to introduce a new element called encrypted data element and define sublement like device ID subelement and IRM subelement and vendor specific subelement to be included in the new element for the content to be encrypted by KEK to align with the operation in 4-way handshake. After this, some corresponding changes in 12.2.12.1 and 12.2.12.2 to mention subelement in the right place. | Revised –  Agree in principle with the commenter.  TGbh editor to make the changes shown in 11-24/0044r3 under all headings that include CID 210 |
| 256 | Emily Qi | 12.2.12.3 | 34.37 | Currently encryption of KEK is for each element.  This is not a scalable approach. We should define a scalable approach so that any new elements or vendor specific elements can be encrypted. | to align with the usage of KEK in 4-way, we should define an Encrypted Data element and define device ID subelement and IRM subelement to be included in the encrypted data element to be encrypted by KEK. | Revised –  Agree in principle with the commenter.  TGbh editor to make the changes shown in 11-24/0044r3 under all headings that include CID 210 |
| 164 | Mark RISON | 12.2.12.3 | 34.29 | "When using PASN authentication, the Device ID element shall be encrypted in PASN frame 2 (if present) and then IRM element shall be encrypted in PASN frame 3 (if present) with the negotiated key wrap algorithm (see Table 12-11-Integrity and key wrap algorithms). To encrypt a Device ID element in PASN frame 2 or an IRM element in PASN frame 3, KEK shall be used, as derived as part of PTK (see 12.13.7 (PTKSA derivation with PASN authentication)), with the negotiated key wrap algorithm (see Table 12-11 (Integrity and key wrap algorithms))." seems to say most things twice, and laboriously | Change to "When using PASN authentication, the Device ID element shall be encrypted in PASN frame 2 (if present) and then IRM element shall be encrypted in PASN frame 3 (if present), using the KEK (see 12.13.7), using the negotiated key wrap algorithm (see Table 12-11-Integrity and key wrap algorithms)." | Revised –  Agree in principle with the commenter. We revise the clause to use a general element that includes subelement and the entire field of the element is encrypted. We also delete the redundant reference of 12-11 and split the sentence into 2.  TGbh editor to make the changes shown in 11-24/0044r3 under all headings that include CID 210 |
| 165 | Mark RISON | 12.2.12.3 | 34.29 | "When using PASN authentication, the Device ID element shall be encrypted in PASN frame 2 (if present) and then IRM element shall be encrypted in PASN frame 3 (if present) with the negotiated key wrap algorithm (see Table 12-11-Integrity and key wrap algorithms). To encrypt a Device ID element in PASN frame 2 or an IRM element in PASN frame 3, KEK shall be used, as derived as part of PTK (see 12.13.7 (PTKSA derivation with PASN authentication)), with the negotiated key wrap algorithm (see Table 12-11 (Integrity and key wrap algorithms))." -- it's not clear how you encrypt an element (as opposed to the payload of an element) | Clarify | Revised –  Agree in principle with the commenter. We revise the clause to use a general element that includes subelement and clarify that the entire field of the element (i.e. payload) is encrypted.  TGbh editor to make the changes shown in 11-24/0044r3 under all headings that include CID 210 |
| 211 | Po-Kai Huang | 12.7.3 | 35.47 | AES-SIV is mandated to be used as the KEK algorithm for PASN. However, AES-SIV is not the most common key wrap algorithm used for key wrap in 4-way. The most common algorithm is NIST AES Key Wrap. As a minimum, we should not mandate AES-SIV when the corresponding base AKM does not even use AES-SIV at all. Further, we note that AES-SIV has the benefits to do AAD, but since we already have KCK to do MIC and verificaiton, it does not make sense to have additional AAD, which is anyhow not defined in 11bh D2.0. | Revise Key wrap algorithm box as "As defined by Base AKMP in Table 12-11 if Base AKMP is not PASN AKMP. NIST AES Key Wrap if Base AKMP is PASN AKMP." Revise KEK\_bits as "As defined by Base AKMP in Table 12-11 if Base AKMP is not PASN AKMP. 128 if Base AKMP is PASN AKMP" | Revised –  Incorporate the changes shown in <this doc> under all headings that include CID 211. This allows both AES-SIV and AES key wrapping to be used with PASN. In addition, the AAD details (it is not used) is described for AES-SIV. |
| 235 | Okan Mutgan | 12.2.12.3 | 34.27 | Section 12.2.12.3 Encryption of Device ID IE and IRM IE in PASN needs further details. | Add encryption details in this section. | Revised -  Incorporate the changes shown in <this doc> under all headings that include CID 211 |

## Discussion:

None

*TGbh editor: Change Clause C.3 as follows (track change on):*

**C.3 MIB detail**

Insert the following entries to the end of the "dot11StationConfigEntry” of the “dot11StationConfig TABLE” as follows:

…,

dot11KEKPASNActivated TruthValue,(#208)

|  |  |
| --- | --- |
| dot11DeviceIDActivated TruthValue, dot11IRMActivated TruthValue |  |

}

***Insert the following elements at the end of the dot11StationConfigTable element definitions:***

dot11KEKPASNActivated OBJECT-TYPE(#208)

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

“This is a control variable. It is written by an external management

entity or the SME. Changes take effect as soon as practical in the implementation.

This attribute, when true, indicates support of deriving KEK in PASN. ”

DEFVAL { false }

::= { dot11StationConfigEntry <ANA>}

dot11DeviceIDActivated OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

“This is a control variable. It is written by an external management

entity or the SME. Changes take effect as soon as practical in the implementation.

This attribute, when true, indicates that the STA might send a device ID. ”

DEFVAL { false }

::= { dot11StationConfigEntry <ANA>}

dot11IRMActivated OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current DESCRIPTION

"This is a control variable. It is written by an external management

entity or the SME. Changes take effect as soon as practical in the implementation. This attribute, when true at a non-AP STA, indicates that the STA might send an IRM. This attribute, when true at an AP indicates that the AP supports IRM."

DEFVAL { false }

::= { dot11StationConfigEntry <ANA> }

*TGbh editor: Change Clause 9.4.2.240 as follows (track change on):*

**9.4.2.240 RSNXE**

***Insert the following new rows in Table 9-371 (Extended RSN Capabilities field) (header row shown for convenience***).

**Table 9-371—Extended RSN Capabilities field**

|  |  |  |
| --- | --- | --- |
| **Bit** | **Information** | **Notes** |
| <ANA> | Device ID Active | A STA sets the Device ID Active field to 1 when dot11DeviceIDActivated is true to indicate that the device ID mechanism is active. Otherwise, the STA sets the Device ID Active field to 0. |
| <ANA> | IRM Active | A STA sets IRM Active field to 1 when dot11IRMActivated is true to indicate that the IRM mechanism is active. Otherwise, the STA sets the IRM Active field to 0. |
| <ANA> | KEK in PASN | The field is set to 1 when dot11KEKPASNActivated is true to indicate support of deriving KEK in PASN. Otherwise, the field is set to 0.(#208) |

*TGbh editor: Modify 9.4.2.1 as follows (track change on):*

**9.4.2.1 General**

*Insert the following new rows in Table 9-130 (Element IDs) (header row shown for convenience) as appropriate*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element | Element ID | Element ID Extension | Extensible | Fragmentable |
| Device ID (see 9.4.2.311 (Device ID element)) | 255 | <ANA> | No | No |
| IRM (see 9.4.2.312 (IRM element)) | 255 | <ANA> | No | No |
| PASN Encrypted Data element (see 9.4.2.314(PASN Encrypted Data element)) | 255 | <ANA> | Subelements | Yes(#210) |

*TGbh editor: Add 9.4.2.314 as follows (track change on): (#210)*

**9.4.2.314 PASN Encrypted Data element**

The PASN Encrypted Data element contains an Encrypted Data field to be encrypted by KEK. The format of the PASN Encrypted Data element is shown in Figure 9-xxxx (PASN Encrypted Data element format).

|  |  |  |  |
| --- | --- | --- | --- |
| Element ID | Length | Element ID Extension | Encrypted Data |

Octets: 1 1 1 variable

**Figure 9-xxx—PASN Encrypted Data element format**

The Element ID, Length, and Element ID Extension fields are defined in 9.4.2.1 (General). The Encrypted Data field contains one or more subelements. The subelement format is defined in [9.4.3](#bookmark125) [(Subelements)](#bookmark125). The Subelement ID field values for the defined subelements of the PASN Encrypted Data element are shown in [Table 9-xxx (Subelement IDs for Encrypted Data field of the PASN Encrypted Data element)](#bookmark184).

**Table 9-xxx—Subelement IDs for Encrypted Data field of the PASN Encrypted Data element**

|  |  |  |
| --- | --- | --- |
| **Subelement ID** | **Name** | **Extensible** |
| 0 | Device ID | No |
| 1 | IRM | No |
| 2–220 | Reserved |  |
| 221 | Vendor Specific | Vendor defined |
| 222–255 | Reserved |  |

The format of the Device ID subelement is shown in Figure 9-xxx (Device ID subelement format).

|  |  |  |  |
| --- | --- | --- | --- |
| Subelement ID | Length | Device ID Status | Device ID |

Octets: 1 1 1 variable

**Figure 9-xxx—Device ID subelement format**

The Subelement ID field is defined in Table 9-xxx (Subelement IDs for Encrypted Data field of the PASN Encrypted Data element).

The Length field is defined in 9.4.3 (Subelements).

The Device ID status field and the Device ID field are defined in 9.4.2.311 (Device ID element).

The format of the IRM subelement is shown in Figure 9-xxx (IRM subelement format).

|  |  |  |  |
| --- | --- | --- | --- |
| Subelement ID | Length | IRM Status | IRM |

Octets: 1 1 2 variable

**Figure 9-1001m—IRM subelement format**

The Subelement ID field is defined in Table 9-xxx (Subelement IDs for Encrypted Data field of the PASN Encrypted Data element).

The Length field is defined in 9.4.3 (Subelements).

The IRM status field and the IRM field are defined in 9.4.2.312 (IRM element).

The Vendor Specific subelements have the same format as their corresponding elements (see 9.4.2.24 (Vendor Specific element)). Zero or more Vendor Specific subelements are included in the Encrypted Data field.

*TGbh editor: Change Clause 9.3.3.11 as follows (track change on):*

* **Authentication frame format**

…(existing texts)….

|  |  |  |  |
| --- | --- | --- | --- |
| * **Presence of fields and elements in Authentication frames** | | | |
| **Authentication algorithm** | **Authentication transaction sequence number** | **Status Code(#3326)** | **Presence of fields and elements  from order 4 onward** |
| PASN Authentication(11az) | 1 | Reserved | RSNE is present.  RSNXE is present if any subfield of the Extended RSN Capabilities field in this element, except the Field Length subfield, is nonzero.  PASN Parameters element is present.  Timeout Interval element may be present.  Wrapped Data element is present if wrapped data format in PASN Parameters element is nonzero and not reserved.  Fragment element may be present if any of the elements are fragmented. |
| PASN Authentication(11az) | 2 | Status | RSNE is present and PASN Parameters element is present if Status Code field is 0.  RSNXE is present if any subfield of the Extended RSN Capabilities field in this element, except the Field Length subfield, is nonzero.  Timeout Interval element may be present.  Wrapped data element is present if wrapped data format in PASN Parameters element is nonzero and not reserved and Status Code field is 0.  PASN Encrypted Data element maybe present.(#210)  MIC element is present.  Fragment element may be present if any of the elements are fragmented and Status Code field is 0. |
| PASN Authentication(11az) | 3 | Status | ASN Parameters element is present if Status Code field is 0.  Wrapped data element is present if wrapped data format in PASN Parameters element is nonzero and not reserved; and Status Code field is 0.  MIC element is present.  PASN Encrypted Data element maybe present.(#210)  Fragment element may be present if any of the elements are fragmented and Status Code field is 0. |

*TGbh editor: Change Clause 12.13.7 as follows (track change on):*

**12.13.7 PTKSA derivation with PASN authentication**

….(existing texts)….

-When dot11KEKPASNActivated is false or when dot11KEKPASNActivated is true and the KEK in PASN field in the RSNXE from the peer is 0 (#208), PTK is composed of the Key Confirmation Key (KCK), Temporal Key (TK) and the Key Derivation Key (KDK) which are derived as follows:

KCK = L(PTK, 0, 256)

KCK is the first 256 bits of the PTK.

TK = L(PTK, 256, TK\_Length\_Bits)

TK is the transient key whose length is the same as a key for the pairwise cipher in RSNE provided by the AP in the second PASN frame This length is 16 octets for all ciphers, except for the ciphers 00-0F-AC:9 and 00-0F-AC:10 for which it is 32 octets.

KDK = L(PTK, 256 + TK\_Length\_Bits, KDK\_bits)

The KDK is of bit length KDK\_bits which has the value 256 if a KDK is derived (see 12.7.1.3 (Pairwise Key Hierarchy)) or 0 otherwise.

KDK shall be derived if dot11SecureLTFImplemented is true and the peer STA has indicated Secure HE-LTF support capability in its advertised Extended Capabilities.

The Key ID in the PTKSA (see 12.6.1.1.6 (PTKSA)) resulting from PASN authentication shall be 3 0.

***Insert the following text as shown.***

-When dot11KEKPASNActivated is true and the KEK in PASN field in the RSNXE from the peer is 1(#208), PTK is composed of the Key Confirmation Key (KCK), Key Encryption Key (KEK), Temporal Key (TK) and the Key Derivation Key (KDK) which are derived as follows (see Table 12-11 (Integrity and key wrap algorithms)):

KCK = L(PTK, 0, 256)

KCK is the first 256 bits of the PTK.

KEK = (PTK, 256, KEK\_bits)

KEK is used to provide encryption for the Encrypted Data field in the Encrypted Data element(#210) in PASN frames, as defined in

12.13.3.2 PASN frame construction and processing. Its length is defined in Table 12-11 (Integrity and key wrap algorithms)

TK = L(PTK, 256 + KEK\_bits, TK\_Length\_Bits)

TK is the transient key whose length is the same as a key for the pairwise cipher in RSNE provided by the AP in the second PASN frame. This length is 16 octets for all ciphers, except for the ciphers 00-0F-AC:9 and 00-0F-AC:10 for which it is 32 octets.

KDK = L(PTK, 256 + KEK\_bits + TK\_Length\_Bits, KDK\_bits)

The KDK is of bit length KDK\_bits which has the value 256 if a KDK is derived (see 12.7.1.3 (Pairwise Key Hierarchy)) or 0 otherwise.

KDK shall be derived if dot11SecureLTFImplemented is true and the peer STA has indicated Secure HE-LTF support capability in its advertised Extended Capabilities.

The Key ID in the PTKSA (see 12.6.1.1.6 (PTKSA)) resulting from PASN authentication shall be 3 0.

….(existing texts)….

*TGbh editor: Change Clause 12.2.12.3 as follows (track change on):*

**12.2.13 Encryption of the Encrypted Data field in PASN Encrypted Data element in PASN(#210)**

When using PASN authentication, the Encrypted Data field of the PASN Encrypted Data element shall be encrypted in PASN frame 2 (if present) and in PASN frame 3 (if present).(#210)

To encrypt the Encrypted Data field of a PASN Encrypted Data element(#210), KEK shall be used, as derived as part of PTK (see 12.13.7 (PTKSA derivation with PASN authentication)), with the negotiated key wrap algorithm (see Table 12-11 (Integrity and key wrap algorithms)).

If the size of the Encrypted data field is larger than 254, then the Encrypted data field shall be encrypted first, then perform element fragmentation as defined in 10.28.11 (Element fragmentation). (#210)

*TGbh editor: Change Clause 12.13.3 as follows (track change on):*

**12.13.3 Key establishment with PSN authentication 12.13.3.2 PASN frame construction and processing**

***Add the following text as shown at the end of the list that begins: “The first PASN authentication frame (see 9.3.3.11) of the exchange is constructed as follows:”***

— If dot11DeviceIDActivated is true, including a Device ID element containing a device identifier as

defined in 9.4.2.311 (Device ID element), if required per the procedure in 12.2.12.1 (Device ID mechanism).

***Add the following text as shown in the list that begins: “***— Derives the PTKSA; see 12.13.7.***”***

* If dot11RSNAOperatingChannelValidationActivated is true, including an OCI Element containing

an OCI element as defined in 9.4.2.236 (OCI element), if dot11RSNAOperatingChannelValidationActivated is true.

* If dot11DeviceIDActivated is true, including a PASN Encrypted Data element and a Device ID subelement containing a device identifier as defined in 9.4.2.311 (Device ID element) in the PASN Encrypted Data element, if required per the procedure in 12.2.12.1 (Device ID mechanism). The PASN Encrypted Data element shall be encrypted as defined in 12.2.12.3 (Encryption Encrytped Data element in PASN). (#210)
* A MIC element (9.4.2.118) with MIC computed as specified in 12.13.8.1.

***Add the following text as shown in the list that begins: “Otherwise the STA begins the constructions of the third PASN frame as follows:”***

* If dot11RSNAOperatingChannelValidationActivated is true, including an OCI Element containing an OCI element as defined in 9.4.2.236 (OCI element).
* If dot11DeviceIDActivated is true, including a Device ID element containing a device identifier as defined in 9.4.2.311 (Device ID element).
* If dot11IRMActivated is true, including a PASN Encrypted Data element and a IRM subelement containing an IRM as defined in 9.4.2.312 (IRM element) in the PASN Encrypted Data element, if the STA so chooses, per the procedure in 12.2.12.2 (Identifiable random MAC address (IRM) operation). The PASN Encrypted Data element shall be encrypted as defined in 12.2.12.3 (Encryption of PASN Encrypted Data element in PASN). (#210)
* A MIC element (9.4.2.117) with MIC computed as specified in 12.13.8.2.

***TGbh editor: Change Clause 12.2.12.1 as follows (track change on):***

**12.2.12.1 Device ID mechanism**

An AP that has dot11DeviceIDActivated equal to true advertises support of the device ID mechanism by setting the Device ID Active field to 1 in the Extended RSN Capabilities field (see 9.4.2.240 (RSNXE)) in Beacon and Probe Response frames.

A non-AP STA that has dot11DeviceIDActivated equal to true, indicates activation of the device ID mechanism by setting the Device ID Active field to 1 in the Extended RSN Capabilities field in (Re)Association Request frames or the first PASN frame sent to any AP in an ESS that has dot11DeviceIDActivated equal to true.

An AP that includes the PASN AKMP as part of the RSNE included in Beacon and Probe Response frames, i.e., when dot11PASNActivated is true, and has dot11DeviceIDActivated equal to true shall set dot11KEKPASNActivated to true.(#208)

A non-AP STA that has dot11DeviceIDActivated equal to true and intends to use PASN, i.e., when dot11PASNActivated is true, shall set dot11KEKPASNActivated to true. (#208)

….(existing texts)….

A STA shall not send a frame containing a device ID (sub)element to any STA unless the receiving STA sets the Device ID Active field to 1 in the Extended RSN Capabilities field. (#210)

….(existing texts)….

An AP shall provide a device ID when required by the procedures described below:

1. When using PASN authentication, in the Device ID subelement in the second PASN frame. (#210)
2. When using FILS authentication, in the Device ID element in the Association Response frame.
3. When not using PASN or FILS authentication, in the Device ID KDE in message 3 of the 4 way

handshake.

….(existing texts)….

When an AP with dot11DeviceIDActivated equal to true receives a non-AP STA Identity frame from a nonAP STA with dot11DeviceIDActivated equal to true and the received device ID is recognized, the AP shall perform one of the following actions:

1. With the Device ID field not present (indicating the current device ID is maintained) and set the

Device ID Status field of the Device ID KDE or Device ID (sub)element to 0 to indicate that the AP recognizes the non-AP STA in the appropriate frame. (#210)

1. Assign a new device ID value in the Device ID field and set the Device ID Status field of the

Device ID KDE or Device ID (sub)element to 0 in the appropriate frame. (#210)

When an AP with dot11DeviceIDActivated equal to true receives a first PASN frame containing a device ID which is recognized, the AP shall assign a new device ID value to the non-AP STA, via setting a new device ID in the Device ID field with the Device ID Status field of the Device ID subelement set to 0 to indicate that the AP recognizes the non-AP STA in the second PASN frame. (#210)

When a non-AP STA receives a frame that contains a Device ID Status field in the Device ID KDE or Device ID (sub)element equal to 0 it may proceed with the assumption that the shared identity state with the AP or ESS (as per the concepts of 12.2.10) is now bound to the TA field in the Association Request frame most recently transmitted by the non-AP STA. (#210)

If an AP sets Device ID (sub)element or Device ID KDE with the Device ID Status field set to 1 indicating “Not Recognized”, then the AP may also provide in that same Device ID (sub)element or Device ID KDE a new device ID, thus establishing a new shared identity. An AP may set a Device ID Status field to 1 indicating “Not Recognized” if the AP cannot unequivocally identify the non-AP STA shared identity state. (#210)

When a non-AP STA receives a frame that contains a Device ID Status field in a Device ID KDE or Device ID (sub)element equal to 1, it shall assume that no shared identity state exists with the AP or ESS (as per the concepts of 12.2.10). (#210)

….(existing texts)….

***TGbh editor: Change Clause 12.2.12.2 as follows (track change on):***

**12.2.12.2 Identifiable random MAC address (IRM) operation**

A non-AP STA that has dot11IRMActivated equal to true indicates activation of the IRM mechanism by setting the IRM Active field to 1 in the Extended RSN Capabilities field in (Re)Association Request frames or the first PASN frame sent to any AP in an ESS that has dot11IRMActivated equal to true.

An AP that has dot11IRMActivated equal to true and that receives a (Re)Association Request frame or the first PASN frame that includes an Extended RSN Capabilities field with the IRM Active field equal to 1 shall do one of the following:

* include an Extended RSN Capabilities element in the (Re)Association Response frame with theIRM Active field set to 1.
* include an Extended RSN Capabilities element in the second PASN frame with the IRM Activefield set to 1.

An AP that includes the PASN AKMP as part of the RSNE included in Beacon and Probe Response frames, i.e., when dot11PASNActivated is true, and has dot11IRMActivated equal to true shall set dot11KEKPASNActivated to true. (#208)

A non-AP STA that has dot11IRMActivated equal to true and intends to use PASN, i.e., when dot11PASNActivated is true, shall set dot11KEKPASNActivated to true. (#208)

….(existing texts)….

When associating to an AP that advertises support for IRM, the non-AP STA may provide a new IRM to the AP by including an IRM KDE in message 4 of the 4-way handshake or, when using FILS authentication, including the IRM element in the Association Request frame. When using PASN, the non-AP STA may provide a new IRM to the AP by including the IRM subelement in the third PASN frame. (#210)

….(existing texts)….

**Discussion for CID 211:**

During offline discussions, two clear issues of the existing texts on mandating AES-SIV 256 becomes clear. First, the key size is limited to the smallest 256 one used when the corresponding SHA is SHA-256 and KCK size is 128. The key size is not suitable when the corresponding SHA is SHA-384 or SHA-512 and corresponding KCK size is 192 and 256. Second, AES-SIV requires definition of AAD, which is not even defined in the current 11bh draft.

Combined with the consideration that the key wrap algorithm does not make sense to be inconsistent with the corresponding key wrap algorithm used by KEK in 4-way for the corresponding AKM, we propose to align the key wrap algorithm and key size with the underlying Base AKMP. We also note that this is nothing new. The existing PASN aligns the key derivation function with the underlying Base AKMP as well.

*KDF-HASH-NNN is the key derivation function defined in 12.7.1.6.2 (Key derivation function (KDF)) using the hash algorithm defined for the Base AKMP*

There is a question on what to do when the Base AKMP is PASN. Then in that case, we create another AKM number to have the version of AES-SIV.

Finally, we add the encryption details by creating the counterpart of the existing 4-way handshake language. For AAD, we note that everything in framebody is already protected by MIC in PASN frame 2 and frame 3, so we simply say that the AAD is empty.

*TGbh editor: Change Clause 12.7.3 as follows (track change on):*

***Insert a new row in Table 9-190 (AKM suite selectors) as shown below.***

**Table 9-190—AKM suite selectors**(#211)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **OUI** | **Suite type** | **Meaning** | | | **Authentication algorithm numbers(see 9.4.1.1 (AuthenticationAlgorithm Number field))** | **Cipher suite selector restriction** |
| **Authentication type** | **Key management type** | **Key derivation type** |
| **00-0F-AC** | **<ANA>** | **PASN with defined key wrap** | **N/A** | **N/A** | **N/A** | **N/A** |

|  |
| --- |
|  |

*TGbh editor: Change Clause 12.7.3 as follows (track change on):*

**12.7.3 EAPOL-Key PDU construction and processing**(#211)

***Modify the following row in Table 12-11 (Integrity and key wrap algorithms) as shown below.***

**Table 12-11—Integrity and key wrap algorithms**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AKM** | **Integrity algorithm** | **KCK\_bits** | **Size of MIC (octets)** | **Key wrap algorithm** | **KEK\_bits** | **KCK2\_bits** | **KEK2\_bits** |
| 00-0F-AC:21 | See NOTE | N/A | N/A | As defined by Base AKMP in Table 12-11 if Base AKMP is not 00-0F-AC:21. NIST AES Key Wrap if Base AKMP is 00-0FAC:21. | As defined by Base AKMP in Table 12-11 if Base AKMP is not 00-0F-AC:21. 128 if Base AKMP is 00-0FAC:21. | N/A | N/A |
| 00-0F-AC:<ANA> | See NOTE | N/A | N/A | AES-SIV-256 | 256 | N/A | N/A |

*TGbh editor: Change Clause 12.13.2 as follows (track change on):*

**12.13.2 Discovery of a PASN capable AP**

An AP indicates it is capable of performing PASN authentication by including the PASN AKMP as part of the RSNE included in Beacon and Probe Response frames. When PASN AKMP is advertised, the AP shall also include at least one additional AKMP in the RSNE unless it allows PTKSA derivation without authentication using the ephemeral keys exchanged during PASN authentication. When PASN AKMP is advertised, the AP with dot11KEKPASNActivated equal to true shall also include 00-0F-AC:<ANA> if AES-SIV is supported as the key wrap algorithm for KEK. (#211)

*TGbh editor: Change Clause 12.13.3.2 as follows (track change on):*

**12.13.3.2 PASN frame construction and processing**

If non-AP STA chooses to initiate PASN authentication, it first selects the following authentication parameters:

* Base AKMP from among AKMPs advertised by the AP if RSNA authentication is desired.

Otherwise, if dot11NoAuthPASNActivated is true, Base AKMP chosen is the PASN AKMP or PASN with defined key wrap AKMP(#211), indicating that PTKSA is to be established without mutual authentication, that is, without a corresponding PMKSA.

* Pairwise cipher suite to use for the PTKSA that is being setup.
* Finite cyclic group from the dot11RSNAConfigDLCGroupTable that is at least of the security strength provided by the Base AKMP and cipher suites.

……

Upon receiving the first PASN frame, the AP:

……

* Verifies the public key as specified in 5.6.2.3 of NIST SP 800-56A R2. If verification fails, the processing status is set to INVALID\_PUBLIC\_KEY. Verifies that a PMKSA named via a PMKID in the RSNE exists for the specified Base AKMP, or the Base AKMP is set to PASN AKMP or PASN with defined key wrap AKMP (#211) or Base AKMP data exists in the frame to allow a PMK to be established. If Base AKMP is equal to PASN AKMP or PASN with defined key wrap AKMP (#211), verifies that dot11NoAuthPASNActivated is set to true. Otherwise processing status is set to REFUSED.

…

Upon receiving the second PASN frame, the non-AP STA:

….

* Verifies that a PMKSA named via a PMKID in the RSNE exists for the specified Base AKMP, or

the Base AKMP is set to PASN AKMP or PASN with defined key wrap AKMP (#211) or Base AKMP data exists in the frame to allow a PMK to be established. If Base AKMP is equal to PASN AKMP or PASN with defined key wrap AKMP (#211), verifies that dot11NoAuthPASNActivated is set to true.

…

*TGbh editor: Change Clause 12.13.7 as follows (track change on):*

**12.13.7 PTKSA derivation with PASN authentication**

For PTKSA key derivation, the inputs to the PRF are the PMK of the PMKSA, a constant label, and a concatenation of non-AP STA’s MAC address, AP’s BSSID, and the DH shared secret from the ephemeral exchange.

PTK = KDF-HASH-NNN (PMK, “PASN PTK Derivation”, SPA || BSSID || DHss)

where

|  |  |
| --- | --- |
| PMK | is the pairwise master key for the Base AKMP if the AKMP is other than PASN AKMP or PASN with defined key wrap AKMP (#211); see 9.4.2.23.3 (AKM suites). Otherwise, if the Base AKMP is PASN AKMP or PASN with defined key wrap AKMP (#211), that is, the PASN PTKSA is being setup without mutual authentication in a non-RSN, the PMK shall be set to the string “PMKz” padded with 28 0s.  NOTE—The PMK for the derivation can come from a cached PMKSA for the AKMP or from the PMKSA established with PASN by tunneling Wrapped Data or Authentication frames. |

……

*TGbh editor: Change Clause 12.2.12.3 as follows (track change on):*

**12.2.12.3 Encryption of Device ID IE and IRM IE in PASN**

… KEK shall be used, as derived as part of PTK (see 12.13.7 (PTKSA derivation with PASN authentication)), with the negotiated key wrap algorithm (see Table 12-11 (Integrity and key wrap algorithms)).

If the Encrypted Data field uses the NIST AES key wrap, then the Encrypted Data field shall be padded before encrypting if the length of the Encrypted Data field is nonzero and less than 16 octets, or if it is not a multiple of 8 octets. The padding consists of appending a single octet 0xdd followed by zero or more 0x00 octets. When processing a received PASN Encrypted Data element, the receiver shall ignore this trailing padding. (#211)

If the Encrypted Data field uses an AEAD cipher, the Encrypted Data field shall not be padded and the AAD for the encipherment operation shall not be used and the number of AAD components is zero. (#211)