IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 802.11bi D0.4 CR for some CIDs on 12.14.8 | | | | |
| Date: 2024-07-15 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Okan Mutgan | Nokia |  |  | okan.mutgan@nokia.com |
|  |  |  |  |  |

Abstract

This document proposes resolutions and discussions for CID1041 and CID1042 on 802.11bi D0.4:

R0. Initial Version.

R1. Changed the resolution for CID1042.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Page | Line | Comment | Proposed Change | Resolution |
| 1041 | 85 | 14 | "12.14.8.3.4 PTKSA derivation with EDPKE authentication defines to derive KEK to encrypt (Re)Association Request/Response frame body.  802.11bh draft also defines to derive KEK. To be compatible with 802.11bh draft's KEK derivation, integrate 802.11bh draft's KEK derivation in 802.11bi draft." | As in comment. | REVISED. |
| 1042 | 84 | 21 | "Enhanced Data Privacy Key Exchange subsection defines dot11EDPEncryptionOfTheFrameBodyFieldOfTheReAssociation  RequestResponseFrameSupportActivated to encrypt (Re)Association Request/Response frame body.  dot11EDPEncryptionOfTheFrameBodyFieldOfTheReAssociation  RequestResponseFrameSupportActivated contains ""ReAssociation"", as if this is intended to be used for Re-Association Req/Resp when roaming. The intention should also cover when non-AP STA associates again (no roaming)." | "Change  dot11EDPEncryptionOfTheFrameBodyFieldOfTheReAssociation  RequestResponseFrameSupportActivated  to  dot11EDPEncryptionOfTheFrameBodyFieldOfTheAssociation  RequestResponseFrameSupportActivated" | REVISED |

**Discussion**

**CID1041**

802.11bi D0.4 uses EDPKE (based on PASN - SAE) to derive KEK to encrypt Association Request/Response.

802.11bh D5.0 also defines KEK derivation for PASN.

The same KEK can be used in 802.11bi draft.

Specifically, 802.11bh D5.0 defines **dot11KEKPASNActivated**, which can be used in 802.11bi as follows:

dot11EDPKEActivated **true**

dot11EDPEncryptionOfTheFrameBodyFieldOfTheReAssociation RequestResponseFrameSupportActivated is **true**

**dot11KEKPASNActivated is true [*add this so that 802.1bi can use the already defined KEK*]**

[KEK is generated so that Assoc Req/Resp can be encrypted]

**CID1042**dot11EDPEncryptionOfTheFrameBodyFieldOfTheReAssociationRequestResponseFrameSupportActivated does not seem to cover Association Request/Response, rather, it seems to cover only ReAssociation Request/Response. Also, the name can be shorthened.

The following can be used:

**dot11EDPEncryptionOfTheReassociationAndAssociationFrameSupportActivated.**

**Proposed Changes**

**CID1041 & CID1042**

*Modify the following sentences in 12.14.8 Enhanced Data Privacy Key Exchange:*

**12.14.8 Enhanced Data Privacy Key Exchange**

**12.14.8.1 General**

If dot11EDPKEActivated is true, then ~~dot11EDPEncryptionOfTheFrameBodyFieldOfTheReAssociation~~

~~RequestResponseFrameSupportActivated~~ dot11EDPEncryptionOfTheReassociationAndAssociationFrameSupportActivated (#1042) and dot11KEKPASNActivated ~~is~~ are true, and the KEK In PASN field in the RSNXE from the peer is 1. (#1041)

Enhanced Data Privacy Key Exchange (EDPKE) is an RSNA authentication protocol that uses the PASN

procedures (see 12.12 (Preassociation security negotiation)) with the following differences:

— SAE AKMP 00-0F-AC:8 or 00-0F-AC:24 can be used as the Base AKMP.

• Use of other authentication protocols as the Base AKMP is TBD

— When there is no Base AKMP, EDKPE is not used.

— The three Authentication frames have the Authentication Algorithm Number field set to  
 <ANA> (EPASN Authentication).

~~— A KEK is generated.~~ (#1041)

— The generated PTK is used as the initial PTK once associated.

**12.14.8.2 Discovery of an EDPKE capable AP**

An AP indicates it is capable of performing EDPKE authentication by including the EDPKE AKMP as part of the RSNE included in Beacon and Probe Response frames. When EDPKE AKMP is advertised, the AP shall also include at least one additional AKMP in the RSNE.

**12.14.8.3 Key establishment with EDPKE authentication**

**12.14.8.3.1 Overview**

This subclause defines the procedures for establishing a PTKSA and the corresponding shared keys between a EDPKE capable STA and AP. The same procedures as specified in 12.12.3.1 (Overview) are used except with the following modifications:

— The three Authentication frames have the Authentication Algorithm Number field set to  
 <ANA> (EDPKE Authentication).

— EDPKE AKMP is used instead of PASN AKMP.

— RSNE indicates EDPKE instead of PASN.

**12.14.8.3.2 EDPKE Frame Construction and Processing**

The same procedures as specified in 12.12.3.2 (PASN Frame Construction and Processing) are used except

with the following modifications:

— The three Authentication frames have the Authentication Algorithm Number field set to <ANA>  
 (EDPKE Authentication).

— EDPKE AKMP is used instead of PASN AKMP.

— RNSE indicates EDPKE instead of PASN.

— The PTK is generated as specified in 12.14.8.3.4 (PTKSA derivation with EDPKE authentication).

**12.14.8.3.3 EDPKE authentication with SAE**

The same procedures as specified in 12.13.5 (PASN authentication with SAE) are used.

**12.14.8.3.4 PTKSA derivation with EDPKE authentication**

The same procedures as specified in 12.13.7 (PTKSA derivation with PASN authentication) are used ~~except~~

~~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:~~

~~KCK = L(PTK, 0, KCK\_bits)~~

~~KEK = L(PTK, KCK\_bits, KEK\_bits)~~

~~TK = L(PTK, KCK\_bits+KEK\_bits, TK\_bits)~~

~~KDK = L(PTK, KCK\_bits+KEK\_bits+TK\_bits, KDK\_bits)~~

~~The values of KCK\_bits and KEK\_bits are AKM suite dependent and are listed in Table 12-11 (Integrity~~

~~and key wrap algorithms). The value of TK\_bits is cipher-suite dependent and is defined in Table 12-8~~

~~(Cipher suite key lengths). If a KDK is derived, the value of KDK\_bits is equal to the value of PMK\_bits;~~

~~otherwise the value of KDK\_bits shall be 0.~~

~~The NNN described in 12.13.7 (PTKSA derivation with PASN authentication) is the Bits required for KCK,~~

~~KEK, TK, and KDK depending on the pairwise cipher and whether a KDK is derived.~~ (#1041)

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

*Modify the following row in Table 9-373:*

**Table 9-373—Extended RSN Capabilities field**

|  |  |  |
| --- | --- | --- |
| Bit | Information | Notes |
| … |  |  |
| <ANA> | Encryption of the Frame Body Field of the (Re)Association Request/Response Frame  Support | A EDP STA sets the Encryption of the  Frame Body Field of the (Re)Association  Request/Response Frame Support subfield  To if ~~dot11EDPEncryptionOfTheFrameBodyFiel~~  ~~dOfTheReAssociationRequestResponseFrameSupportActivated~~ dot11EDPEncryptionOfTheReassociationAndAssociationFrameSupportActivated (#1042) is true. Otherwise, this subfield is set to 0. See 12.14.5 (Encryption of the Frame Body Field of the (Re)Association Request/ Response Frame). |
| … |  |  |