IEEE P802.11
Wireless LANs

|  |
| --- |
| Proposed Spec Text for Extended PASN for 11bi |
| Date: 2023-01-15 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Duncan Ho | Qualcomm Technologies, Inc. |  | +1-858-845-3214 | dho@qti.qualcomm.com |
| Jouni Malinen |  |  |  |
| Philip Hawkes |  |  |  |
| George Cherian |  |  |  |

**Introduction:**

We propose the initial spec text in this document for Requirement #4 in contribution “11-21-1848-16-00bi-requirements-document”:

**Requirement #4:** 11bi shall define a mechanism for a CPE Client and CPE AP to establish keys from an Authentication exchange which can then be used to protect the (Re)Association Request/Response.

**Revisions:**

* Rev 0: Initial version of the document.
* Rev 1: fixed some editorials and limit EPASN to use SAE (and removed support of FILS and FT).

**Discussion:**

We propose to extend the Preassociation security negotiation (PASN) in P802.11az D7.0 to provide keys for protecting the (Re)Association Request and Response frames.

**High level summary of the changes:**

* Define a new 802.11 authentication method called “Extended preassociation security negotiation (EPASN)”, which is closely based on PASN with the following modifications:
	+ Add KEK (256-bit) to the PTK generated by PASN
	+ Use the KEK to protect the GTK/IGTK/BIGTK (sent from AP to STA)
	+ The PTK generated by the EPASN is used to protect the (Re)Association Request and Response
	+ Allow the PTK generated by EPASN to be used during the time the STA is associated with the AP

**Proposed spec text:**

The baseline for this text is P802.11az D7.0 (with Word Change Tracking turned ON).

To editor: since the 11az text also contains changes against 802.11 REVme, please follow only the directions highlighted in yellow below and apply only the changes shown by Word tracking in this document.

3.2 Definitions specific to IEEE 802.11

***Insert the new definition into 3.2 in alphabetical order:***

**EPASN:** Extendedpreassociation security negotiation.

4.5.4.2 Authentication

***Modify 4.5.4.2 as follows:***

***Change the following paragraphs as shown:***

IEEE Std 802.11™-2020 defines ~~five~~six IEEE 802.11 authentication methods: Open System authentication, Shared Key authentication, FT authentication, (IEEE Std 802.11™-2020) simultaneous authentication of equals (SAE), ~~and~~ FILS authentication (IEEE Std 802.11™-2020), preassociation security negotiation (PASN) authentication and extended preassociation security negotiation (EPASN) authentication. Open System authentication admits any STA to the DS. Shared Key authentication relies on WEP to demonstrate knowledge of a WEP encryption key. FT authentication relies on keys derived during the initial mobility domain association to authenticate the stations as defined in Clause 13 (Fast BSS transition). SAE authentication uses finite field cryptography to prove knowledge of a shared password. FILS authentication allows for faster connection to the network for FILS non-AP STAs by providing authentication, association, and key confirmation information in an efficient number of frame exchanges (see 4.10.3.6). (E)PASN authentication allows for the protection of Management frames without association by establishing a PTKSA using authentication frames. The IEEE 802.11 authentication mechanism also allows definition of new authentication methods, or any combination of these authentication methods.

An RSNA might support one or more of SAE authentication, FILS authentication, or (E)PASN authentication both (IEEE Std 802.11™-2020) methods. An RSNA also supports authentication based on IEEE Std 802.1X-2010, or preshared keys (PSKs) after Open System authentication. IEEE 802.1X authentication utilizes the EAP to authenticate STAs and the AS with one another. This standard does not specify an EAP method that is mandatory to implement. See 12.6.5 (RSNA policy selection in an IBSS) for a description of the IEEE 802.1X authentication and PSK usage within an IEEE 802.11 IBSS.

***Insert the following paragraphs at the end of 4.5.4.2:***

(E)PASN authentication is used in an RSN for an infrastructure BSS when it is based on a PMKSA established by another RSN authentication protocol. Otherwise, it does not guarantee mutual authentication, and can be used as a non-RSN protocol in an infrastructure BSS.

6.3.5 Authenticate

***Modify 6.3.5.2, 6.3.5.3, 6.3.5.4, and 6.3.5.5 as follows:***

6.3.5.2 MLME-AUTHENTICATE.request

6.3.5.2.2 Semantics of the service primitive

***Change the primitive parameters as follows in the .request:***

The primitive parameters are as follows:

 MLME-AUTHENTICATE.request (

………………………….

Content of FILS Authentication frame,

Content of PASN Authentication frame,

Content of EPASN Authentication frame,

VendorSpecificInfo

)

***Insert a new row in the following unnumbered table:***

|  |  |  |  |
| --- | --- | --- | --- |
| Name  | Type  | Valid range  | Description |
| Content of PASN Authentication frame | Sequence of elements and fields | As defined in [12.12.3.2](#H12o13o3o2) PASN Frame Construction and Processing. [9.4.2.24](#H09o4o2o24) (RSNE), [9.4.2.241](#H09o4o2o241) (RSNXE), [9.4.2.187](#H09o4o2o187) (Wrapped Data element), [9.4.2.303](#H09o4o2o303) (PASN Parameters element), 9.4.2.48 (Timeout Interval element) | The set of elements and fields to be included in PASN Authentication frames. Present if AuthenticationType indicates PASN authentication and dot11PASNActivated is true, otherwise not present. |
| Content of EPASN Authentication frame | Sequence of elements and fields | As defined in [12.14.x.3.2](#H12o13o3o2) EPASN Frame Construction and Processing. [9.4.2.24](#H09o4o2o24) (RSNE), [9.4.2.241](#H09o4o2o241) (RSNXE), [9.4.2.187](#H09o4o2o187) (Wrapped Data element), [9.4.2.303](#H09o4o2o303) (PASN Parameters element), 9.4.2.48 (Timeout Interval element) | The set of elements and fields to be included in EPASN Authentication frames. Present if AuthenticationType indicates EPASN authentication and dot11EPASNActivated is true, otherwise not present. |
| VendorSpecificInfo  |  A set ofelements |  As defined in 9.4.2.25(Vendor Specific element) |  Zero or more elements. |

6.3.5.3 MLME-AUTHENTICATE.confirm

6.3.5.3.2 Semantics of the service primitive

***Change the primitive parameters as follows in the .confirm:***

MLME-AUTHENTICATE.confirm(

………………………….

Content of FILS Authentication frame,

Content of PASN Authentication frame,

Content of EPASN Authentication frame,

VendorSpecificInfo

)

***Insert the following entry into the unnumbered table in this subclause maintaining the primitive order above :***

|  |  |  |  |
| --- | --- | --- | --- |
| Name  | Type  | Valid range  | Description |
| … | … | … | … |
| AuthenticationType | Enumeration | OPEN\_SYSTEM,SHARED\_KEYFAST\_BSS\_TRANSITION,SAE, FILS\_SHAREDKEY\_WITHOUT\_PFS,FILS\_SHARED\_KEY\_WITH\_PFS,FILS\_PUBLIC\_KEY,PASN | Specifies the type of authenticationalgorithm that was used during theauthentication process. This valuematches the AuthenticationTypeparameter specified in the correspondingMLME-AUTHENTICATE.requestprimitive. |
| … | … | … | … |
| Content of PASN Authentication frame | Sequence of elements and fields | As defined in [12.12.3.2](#H12o13o3o2) PASN Frame Construction and Processing, [9.4.2.241](#H09o4o2o241) (RSNXE), [9.4.2.24](#H09o4o2o24) (RSNE), [9.4.2.187](#H09o4o2o187) (Wrapped Data element), [[[9.4.2.303](#H09o4o2o303)](#H09o4o2o303)](#H09o4o2o301) (PASN Parameters element), 9.4.2.48 (Timeout Interval element) | The set of elements and fields to be included in PASN Authentication frames. Present if AuthenticationType indicates PASN authentication and dot11PASNActivated is true, otherwise not present. |
| Content of EPASN Authentication frame | Sequence of elements and fields | As defined in [12.14.x.3.2](#H12o13o3o2) EPASN Frame Construction and Processing, [9.4.2.241](#H09o4o2o241) (RSNXE), [9.4.2.24](#H09o4o2o24) (RSNE), [9.4.2.187](#H09o4o2o187) (Wrapped Data element), [9.4.2.303](#H09o4o2o303) (PASN Parameters element), 9.4.2.48 (Timeout Interval element) | The set of elements and fields to be included in EPASN Authentication frames. Present if AuthenticationType indicates EPASN authentication and dot11EPASNActivated is true, otherwise not present. |

6.3.5.4 MLME-AUTHENTICATE.indication

6.3.5.4.2 Semantics of the service primitive

***Change the primitive parameters as follows in the .indication:***

MLME-AUTHENTICATE.indication(

………………………….

Content of FILS Authentication frame,

Content of PASN Authentication frame,

Content of EPASN Authentication frame,

VendorSpecificInfo

)

***Insert the following entry into the unnumbered table in this subclause maintaining the primitive order above :***

|  |  |  |  |
| --- | --- | --- | --- |
| Name  | Type  | Valid range  | Description |
| … | … | … | … |
| AuthenticationType | Enumeration | OPEN\_SYSTEM,SHARED\_KEYFAST\_BSS\_TRANSITION,SAE, FILS\_SHAREDKEY\_WITHOUT\_PFS,FILS\_SHARED\_KEY\_WITH\_PFS,FILS\_PUBLIC\_KEY,PASN | Specifies the type of authenticationalgorithm that was used during theauthentication process. This valuematches the AuthenticationTypeparameter specified in the correspondingMLME-AUTHENTICATE.requestprimitive. |
| … | … | … | … |
| Content of PASN Authentication frame | Sequence of elements and fields | As defined in [12.12.3.2](#H12o13o3o2) PASN Frame Construction and Processing. [9.4.2.24](#H09o4o2o24) (RSNE), [9.4.2.241](#H09o4o2o241) (RSNXE), [9.4.2.187](#H09o4o2o187) (Wrapped Data element), [[[9.4.2.303](#H09o4o2o303)](#H09o4o2o303)](#H09o4o2o301) (PASN Parameters element), 9.4.2.48 (Timeout Interval element) | The set of elements and fields to be included in PASN Authentication frames. Present if AuthenticationType indicates PASN authentication and dot11PASNActivated is true, otherwise not present. |
| Content of EPASN Authentication frame | Sequence of elements and fields | As defined in [12.14.x.3.2](#H12o13o3o2) EPASN Frame Construction and Processing. [9.4.2.24](#H09o4o2o24) (RSNE), [9.4.2.241](#H09o4o2o241) (RSNXE), [9.4.2.187](#H09o4o2o187) (Wrapped Data element), [9.4.2.303](#H09o4o2o303) (PASN Parameters element), 9.4.2.48 (Timeout Interval element) | The set of elements and fields to be included in EPASN Authentication frames. Present if AuthenticationType indicates EPASN authentication and dot11EPASNActivated is true, otherwise not present. |

6.3.5.5 MLME-AUTHENTICATE.response

6.3.5.5.2 Semantics of the service primitive

***Change the primitive parameters as follows in the .response***

 MLME-AUTHENTICATE.response(

………………………….

Content of FILS Authentication frame,

Content of PASN Authentication frame,

Content of EPASN Authentication frame,

VendorSpecificInfo

)

***Insert the following entry into the unnumbered table in this subclause maintaining the primitive order above :***

|  |  |  |  |
| --- | --- | --- | --- |
| Name  | Type  | Valid range  | Description |
| … | … | … | … |
| AuthenticationType | Enumeration | OPEN\_SYSTEM,SHARED\_KEYFAST\_BSS\_TRANSITION,SAE, FILS\_SHAREDKEY\_WITHOUT\_PFS,FILS\_SHARED\_KEY\_WITH\_PFS,FILS\_PUBLIC\_KEY,PASN | Specifies the type of authenticationalgorithm that was used during theauthentication process. This valuematches the AuthenticationTypeparameter specified in the correspondingMLME-AUTHENTICATE.requestprimitive. |
| … | … | … | … |
| Content of PASN Authentication frame | Sequence of elements and fields | As defined in [12.12.3.2](#H12o13o3o2) PASN Frame Construction and Processing. [9.4.2.24](#H09o4o2o24) (RSNE), [9.4.2.187](#H09o4o2o187) (Wrapped Data element), [[[9.4.2.303](#H09o4o2o303)](#H09o4o2o303)](#H09o4o2o301) (PASN Parameters element), 9.4.2.48 (Timeout Interval element)  | The set of elements and fields to be included in PASN Authentication frames. Present if AuthenticationType indicates PASN authentication and dot11PASNActivated is true, otherwise not present. |
| Content of EPASN Authentication frame | Sequence of elements and fields | As defined in [12.14.x.3.2](#H12o13o3o2) EPASN Frame Construction and Processing. [9.4.2.24](#H09o4o2o24) (RSNE), [9.4.2.187](#H09o4o2o187) (Wrapped Data element), [9.4.2.303](#H09o4o2o303) (PASN Parameters element), 9.4.2.48 (Timeout Interval element)  | The set of elements and fields to be included in EPASN Authentication frames. Present if AuthenticationType indicates EPASN authentication and dot11EPASNActivated is true, otherwise not present. |

9.3.3.11 Authentication frame format

***Modify 9.3.3.11 as follows:***

***Insert the new row at the end of table 9-40:***

1. Table 9-40—Authentication frame body

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| 41 | PASN Parameters |  A PASN Parameters element is present only in certain Authentication frames as defined in Table [9-41](#T09o41) (Presence of fields and elements in Authentication frames). |

***Insert the new rows at the end of table 9-41:***

1. Table 9-41—Presence of fields and elements in Authentication frames *(continued):*

|  |  |  |  |
| --- | --- | --- | --- |
| Authentication algorithm | **Authentication transaction sequence number** | **Status code** | **Presence of fields 4 onwards** |
| EPASN authentication | 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. |
| EPASN authentication  | 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.MIC element is presentFragment element may be present if any of the elements are fragmented and Status Code field is 0.  |
| EPASN authentication | 3 | Status | PASN 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 presentFragment element may be present if any of the elements are fragmented and Status Code field is 0.  |

9.4 Management and Extension frame body components

9.4.1 Fields that are not elements

9.4.1.1 Authentication Algorithm Number field

***Insert the new algorithm number <ANA> for EPASN authentication before the line, “vendor specific use”:***

Authentication algorithm number = 1: Shared Key

Authentication algorithm number = 2: Fast BSS Transition

Authentication algorithm number = 3: Simultaneous Authentication of Equals (SAE)

Authentication algorithm number = 4: FILS Shared Key authentication without PFS

Authentication algorithm number = 5: FILS Shared Key authentication with PFS

Authentication algorithm number = 6: FILS Public Key authentication

Authentication algorithm number = 7: PASN authentication

Authentication algorithm number = <ANA>: EPASN authentication

Authentication algorithm number = 65 535: vendor specific use

9.4.2.24 RSNE

9.4.2.24.3 AKM suites

***Insert the following new row (EPASN) into Table 9-151 (AKM suite selectors):***

1. Table 9-151—AKM suite selectors

|  |  |  |
| --- | --- | --- |
| **OUI** | **Suite type** | **Meaning** |
| **Authentication type** | **Key management type** | **Key derivation****type** | **Authentication****algorithm****numbers (see****9.4.1.1****(Authentication****Algorithm****Number****field))** |
| 00-0F-AC | 21 | PASN-1 | PASN | PASN key management defined in [[12.12](#H12o12)](#H12o13) (Pre Association Security Negotiation) | Defined in [12.12.3](#H12o13o3) (Key establishment with PASN authentication) |
| 00-0F-AC | <ANA> | EPASN | EPASN | EPASN key management defined in [12.1](#H12o12)4.x (Extended preassociation security negotiation) | Defined in [12.14.x.3](#H12o13o3) (Key establishment with EPASN Authentication) |

9.4.2.187 ~~FILS~~ Wrapped Data element

***Modify 9.4.2.187 as follows:***

The ~~FILS~~ Wrapped Data element is used for the STA and AP to communicate data used by the ~~FILS~~ RSNA protocols ~~authentication algorithm~~. The format of the ~~FILS~~ Wrapped Data element is defined in Figure [9-664](#F09o664) (~~FILS~~ Wrapped Data element format).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Element ID | Length | Element IDExtension | ~~FILS~~ Wrapped Data |
| Octets | 1 | 1 | 1 | variable |

1. Figure 9-664—~~FILS~~ Wrapped Data element format

The Element ID, Length, and Element ID Extension fields are defined in [9.4.2.1](#H09o4o2o1) (General).

The ~~FILS~~ Wrapped Data field is the data used by the FILS authentication algorithm (see [12.11](#H12o11) (Authentication for FILS)), PASN authentication algorithm (see [12.12](#H12o12) (Preassociation Security Negotiation)), and EPASN authentication algorithm (see 12.14.x (Extended Preassociation Security Negotiation)).

11.3.4.2 Authentication—originating STA

***Change 11.3.4.2 Authentication—originating STA as follows:***

***Modify 11.3.4.2 as follows:***

**…**

c) For any authentication protocol that is not PASN, if the authentication was successful within the AuthenticateFailureTimeout, the state for the indicated STA shall be set to State 2 if it was in State 1 or State 1a; the state shall remain unchanged if it was other than State 1 or State 1a.

d) For PASN authentication, if the authentication was successful within the AuthenticateFailureTimeout, the state for the indicated STA shall be set to State 1a if it was State 1; PASN authentication is disallowed in states other than State 1.

The MLME shall issue an MLME-AUTHENTICATE.confirm primitive to inform the SME of the result of the authentication.

***Insert the following paragraph after “4)”:***

5) For PASN authentication, the authentication mechanism described in [12.12](#H12o12) (Preassociation Security Negotiation).

6) For EPASN authentication, the authentication mechanism described in 12.14.x (Extended Preassociation Security Negotiation).

11.3.4.3 Authentication—destination STA

***Modify 11.3.4.3 as follows:***

***Change 11.3.4.3 Authentication—destination STA as follows:***

**…**

f) Upon receipt of an MLME-AUTHENTICATE.response primitive, if the ResultCode is not

SUCCESS, the MLME shall transmit an Authentication frame with the corresponding status code, as defined in 9.4.1.9 (Status Code field), and the state for the originating STA shall be left

unchanged. The Authentication frame is constructed using the appropriate procedure in 12.3.3.2

(Open System authentication), 12.3.3.3 (Shared Key authentication), 13.5 (FT protocol), ~~or~~ 13.6 (FT resource request protocol) or [12.12](#H12o12) (Preassociation Security Negotiation) or 12.14.x (Extended Preassociation Security Negotiation).

g) Upon receipt of an MLME-AUTHENTICATE.response primitive, if the ResultCode is SUCCESS, the MLME shall transmit an Authentication frame that is constructed using the appropriate procedure in 12.3.3.2 (Open System authentication), 12.3.3.3 (Shared Key authentication), 13.5 (FT protocol), ~~or~~ 13.6 (FT resource request protocol), or [12.12](#H12o12) (Preassociation Security Negotiation) with a status code of SUCCESS.~~, and~~ ~~the~~

The state for the originating STA shall be set to State 2 if it was in State 1 or state 1a when PASN authentication procedure was not used. . The state for the originating STA shall be set to State 1a if it was in State 1 and PASN authentication procedure was used. PASN authentication shall be disallowed in states other than State 1.

12. Security

12.2 Framework

12.2.4 RSNA establishment

***Insert the following paragraph after “g)”:***

g) If an RSNA uses PASN authentication, an RSNA capable STA establishes an RSNA as described in 12.12 (Preassociation Security Negotiation).

h) If an RSNA uses EPASN authentication, an RSNA capable STA establishes an RSNA as described in 12.14.x (Extended Preassociation Security Negotiation).

12.6 RSNA security association management

12.6.1 Security associations

12.6.1.1 Security association definitions

12.6.1.1.1 General

***Change the following sentence:***

***Modify 12.6.1.1.1 as follows:***

PTKSA: A result of a successful 4-way handshake, FT 4-way handshake, FT authentication sequence, FILS authentication, PASN authentication, or EPASN authentication.

12.6.1.1.6 PTKSA

*Change 12.6.1.1.6 PTKSA as follows*

***Modify 12.6.1.1.6 as follows:***

The PTKSA results from a successful 4-way handshake, FT 4-way handshake, FT protocol, FT resource request protocol, ~~or~~ FILS authentication, PASN authentication, or EPASN authentication. This security association is also bidirectional.

PTKSAs, except those established using PASN authentication, have the same lifetime as the PMKSA or PMK-R1 security Association, whichever comes first. PTKSAs for PASN authentication have a minimum of the lifetime of the PMKSA used and the timeout negotiated, if any, during PASN authentication.

Because the PTKSA is tied to the PMKSA or to a PMK-R1 security association, it only has the additional information from the 4-way handshake, FT Protocol authentication, FILS authentication, PASN authentication, or EPASN authentication.

12.6.7 RSNA policy selection in an MBSS

***Insert the following line at the end of 12.6.7.***

***Modify 12.6.7 as follows:***

When establishing an RSNA in an MBSS, PASN authentication and EPASN authentication shall not be used.

***Insert the following new section 12.14.x under section 12.14 Client Privacy Enhancement:***

12.14 Client Privacy Enhancement

### 12.14.x Extended preassociation security negotiation

#### 12.14.x.1 General

If dot11EPASNActivated is true, then dot11EDPEncryptionOfTheFrameBodyFieldOfTheReAssociation RequestResponseFrameSupportActivated is true.

Extended Pre-Association Security Negotiation (EPASN) is an RSNA authentication protocol that uses the PASN procedures (see 12.12 (Preassociation security negotiation)) with the following differences:

* Only SAE AKMP 00-0F-AC:8 or 00-0F-AC:24 can be used as the Base AKMP when PMK caching is not used.
* The three Authentication frames have the Authentication Algorithm Number field set to <ANA> (EPASN Authentication).
* A KEK is generated.
* The generated PTK is used as the initial PTK once associated.

#### 12.14.x.2 Discovery of an EPASN capable AP

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

#### 12.14.x.3 Key establishment with EPASN authentication

12.14.x.3.1 Overview

This subclause defines the procedures for establishing a PTKSA and the corresponding shared keys between a EPASN 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> (EPASN Authentication).
* EPASN AKMP is used instead of PASN AKMP.
* RSNE indicates EPASN instead of PASN.

12.14.x.3.2 EPASN 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> (EPASN Authentication).
* EPASN AKMP is used instead of PASN AKMP.
* RNSE indicates EPASN instead of PASN.
* The PTK is generated as specified in 12.14.x.3.4 (PTKSA derivation with EPASN authentication).

12.14.x.3.3 EPASN authentication with SAE

The same procedures as specified in 12.12.5 (PASN authentication with SAE) are used.

12.14.x.3.4 PTKSA derivation with EPASN authentication

The same procedures as specified in 12.12.7 (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 12-8 (Integrity and key-wrap algorithms). The value of TK\_bits is cipher-suite dependent and is defined in Table 12-5 (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 12.12.7 (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.

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

# Annex C

**(normative)**

1. ASN.1 encoding of the MAC and PHY MIB

## C. 3 MIB detail

***…***

***Modify “Dot11StationConfigEntry” as follows:***

Dot11StationConfigEntry ::=

SEQUENCE{

…….

dot11S1GOptionImplemented TruthValue,

dot11PASNActivated TruthValue,

dot11NoAuthPASNActivated TruthValue,

dot11EPASNActivated TruthValue
 }

***Insert the following object before the “End of ..” as shown below:***

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- \* dot11StationConfig TABLE

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*…….*

dot11EPASNActivated 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 for the next MLME-START.request primitive or MLME JOIN.request primitive. This attribute indicates whether or not EPASN authentication is enabled."

DEFVAL {false}

::= { dot11StationConfigEntry <ANA> }

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- \* End of dot11StationConfig TABLE

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#