IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11bi D1.0 12.16.5 comments | | | | |
| Date: 2025-03-25 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | Email |
| Po-Kai Huang | Intel |  |  | po-kai.huang@intel.com |

Abstract

This submission resolves the following CIDs:

649, 965, 980

Revisions:

* Rev 0: Initial version of the document.
* Rev 1: Revision based on the discussion during the meeting
* Rev 2: Revision for CID 965 and add CID 649
* Rev 3: Revision for CID 965.
* Rev 4: Revision for CID 649.
* Rev 5: Revision for CID 965 based on the discussion during the teleconference. Add CID 980.
* Rev 6: Update typo in the resolution of 980

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 TGbi D1.0 Draft. This introduction is not part of the adopted material.

Editing instructions formatted like this are intended to be copied into the TGbi D1.0 Draft. (i.e. they are instructions to the 802.11 editor on how to merge the text with the baseline documents). TGbi Editor: Editing instructions preceded by “TGbi Editor” are instructions to the TGbi editor to modify existing material in the TGbi draft. As a result of adopting the changes, the TGbi editor will execute the instructions rather than copy them to the TGbi Draft.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **Clause** | **P.L** | **Comment** | **Proposed Change** | **Resolution** |
| 649 | 110,28 | 12.6.1.2.2 | "The last step is key management. The authentication process, whether SAE authentication, or FILS authentication, or IEEE 802.1X authentication utilizing Authentication frames; or IEEE 802.1X authentication utilizing Data frames post association,; or the OWE exchange utilizing association frames creates cryptographic keys shared between the cryptographic endpoints--the AP and STA, or the IEEE 802.1X AS and the STA, when using SAE/FILS/OWE or IEEE Std 802.1X, respec-tively." -- this is barely comprehensible, but the mixture of commas and semicolons makes it even worse | As it says in the comment | Revised –  Agree to fix the mixture of commas and semicolons by using or exclusively. Also, tweak the sentence of the paragraph.  TGbi editor to make the changes shown in the latest version of 11-25/0536 under all headings that include CID 965 |
| 965 | 123.56 | 12.16.5 | It is not described how the EAPOL exchange using Authentication frames is completed. Is Status Code 0 used in all cases? Even with EAP-Failure? Is the non-AP STA support to upper layer information (i.e., EAP-Success or EAP-Failure) to determine when authentication has been completed? If so, that needs to be specified. If not, there needs to be lower layer information (e.g., one new Status Code to indicate EAP exchange continues and Status Code 0 to indicate that EAP completed successfully and new Status Code to indicate that EAP resulted in failure). | Describe the exact rules based on which a non-AP STA or a non-AP MLD concludes that authentication succeeded or failed and can move to the next step (i.e., association in the success case). | Revised –  Agree in principle with the commenter.  Note that when EAPOL PDU is exchanged with the Data frame, if the authentication fails, then there is a deauthentication frame with reason code “802\_1\_X\_AUTH\_FAILED”. We add this reason code to the status code when the 802.1X authentication fails. This happens when the EAPOL PDU is EAP Failure.  We also add the status code “802\_1\_X\_AUTH\_SUCCESS” to indicate the case when the 802.1X authentication is successful. This happens when the EAPOL PDU is EAP Success.  ***2. Extensible Authentication Protocol (EAP)***  *The EAP authentication exchange proceeds as follows:  …..*   * *[4] The conversation continues until the authenticator cannot authenticate the peer (unacceptable Responses to one or more Requests), in which case the authenticator implementation MUST transmit an EAP Failure (Code 4). Alternatively, the authentication conversation can continue until the authenticator determines that successful authentication has occurred, in which case the authenticator MUST transmit an EAP Success (Code 3).*   TGbi editor to make the changes shown in the latest version of 11-25/0536 under all headings that include CID 965 |
| 980 | 4.10.2 | 25.20 | Clauses 4 and 5 of the baseline need to be enhanced to talk about how Authentication frames that are carrying EAPOL PDUs get their payload routed to the PAE and then Authenticator/Supplicant. It isn't sufficient to just say "Data frames or Authentication frames", when these are discussed in the context of the [Un]Controlled Port (which is a data plane concept). Clause 4 needs some architectural work to enable this link from Management frame (Authentication frame) to the PAE. This might need to ripple into the MLME SAP, it's not clear. | Some architecture needs to be created for this new pathway to the PAE, and description added (probably separately from the EAPOL Data frame text). | Revised –  Agree in principle to update clause 4. Clause 5 is about MSDU, so it is not related to the Authentication frame exchange. Essentailly, 802.11 just extracts the EAPOL PDU and foreward to PAE directly without going through the data exchange. PAE is defined in 802.1X-2020 to handle EAPOL PDU processing.  *5.4 PAE requirements*  *A PAE shall*  *a) Encode, decode, address, and validate EAPOL PDUs as specified in Clause 11.*  TGbi editor to make the changes shown in the latest version of 11-25/536 under all headings that include CID 980 |

***Discussion:***

***Proposal:***

**TGbi Editor: *Modify 4.3.8 Robust security network association (RSNA) as follows:***

**4.3.8 Robust security network association (RSNA)**

(…existing texts…)

An RSNA might rely on components external to the IEEE 802.11 architecture.

The first component is an IEEE 802.1X port access entity (PAE). A PAE is present on each STA or each MLD in an RSNA to process the EAPOL PDUs(#980) and control the forwarding of data to and from the medium access control (MAC). An AP or an AP MLD always implements the Authenticator PAE and Extensible Authentication Protocol (EAP) Authenticator roles, and a non-AP STA or a non-AP MLD always implements the Supplicant PAE and EAP peer roles. In an IBSS or PBSS, each STA implements both the Authenticator PAE and Supplicant PAE roles and both EAP Authenticator and EAP peer roles.

A second component is the Authentication Server (AS). The AS authenticates the elements of the RSNA itself, i.e., the STAs or MLDs provide material that the RSNA elements use to authenticate each other. The AS communicates through the IEEE 802.1X Authenticator with the IEEE 802.1X Supplicant on each STAor on each MLD, enabling the STA or the MLD to be authenticated to the AS and vice versa. An RSNA depends upon the use of an EAP method that supports mutual authentication of the AS and the STA or mutual authentication of the AS and the MLD, such as those that meet the requirements in IETF RFC 4017. In certain applications, the AS might be integrated into the same physical device as the AP or the AP MLD, or into a STA in an IBSS or PBSS.

In some applications, there is no need for a PAE or AS, and a STA and AP, or a non-AP MLD and AP MLD, or two IBSS STAs, or two mesh STAs in an MBSS, might authenticate each other using a password.

(…existing texts…)

**TGbi Editor: *Modify 4.10 as follows:***

* IEEE Std 802.11 and IEEE Std 802.1X-2020
* IEEE 802.11 usage of IEEE Std 802.1X-2020

***Change the first paragraph as follows:***

IEEE Std 802.11 depends upon IEEE Std 802.1X-2020 to control the flow of MAC service data units (MSDUs) between the DS and STAs by use of the IEEE 802.1X Controlled/Uncontrolled Port model. IEEE 802.1X EAPOL PDUs are transmitted either in one or more IEEE 802.11 Data frames and passed via the IEEE 802.1X Uncontrolled Port or in one or more Authentication frames(#980). The IEEE 802.1X Controlled Port is blocked from passing general data traffic between two STAs until an IEEE 802.1X authentication procedure completes successfully over the IEEE 802.1X Uncontrolled Port or over the Authentication frames exchange carrying EAPOL PDUs(#980). It is the responsibility of both the Supplicant and the Authenticator to implement port blocking. Each association between a pair of STAs creates a unique pair of IEEE 802.1X Ports, and authentication takes place relative to those ports alone or through exchange of Authentication frames carrying EAPOL PDUs.(#980)

* Infrastructure functional model overview
* AKM operations with AS

***Change item b) of the first paragraph as follows (not all lines are shown):***

The following AKM operations are carried out when an IEEE 802.1X AS is used:

* A STA discovers the AP’s security policy through passively monitoring Beacon frames or through active probing. If IEEE 802.1X authentication is used, the EAP authentication process starts when the Authenticator sends the EAP-Request or the Supplicant sends the EAPOL-Start PDU (in one or more EAPOL-Start frames or in an Authentication frame(#962)). EAP messages pass between the Supplicant and AS via the Authenticator and Supplicant’s Uncontrolled Ports as described in 12.7 (Keys and key distribution). EAP messages pass between the Supplicant and AS via the Authentication frames exchange carrying EAPOL PDUs as described in 12.16.5 (IEEE 802.1X authentication utilizing Authentication frames).(#980)

(…existing texts)

The following is achieved by a 4-way handshake or FT 4-way handshake utilizing (#1836)EAPOL-Key PDUs initiated by the Authenticator or an IEEE 802.1X Authentication frame exchange followed by an encrypted (Re)Association Request/Response frame exchange: (#980)

* Confirm that a live peer holds the PMK.
* Confirm that the PMK is current.
* In the case of fast BSS transition, derive PMK-R0s and PMK-R1s.
* Derive a fresh pairwise transient key (PTK) from the PMK or, in the case of fast BSS transition, from the PMK-R1, the derived PTK including the key derivation key (KDK) if WUR frame protection is negotiated.(11ba)
* If WUR frame protection is negotiated, derive a fresh WTK from the KDK.(11ba)
* Install the pairwise encryption and integrity keys, and if WUR frame protection is negotiated, the WTK.(11ba)
* Transport the group (#1349)keys and sequence number from Authenticator to Supplicant and install the (#1349)group keys and sequence number in the STA and, if not already installed, in the AP.
* Verify that the RSN capabilities negotiated are valid as defined in 9.4.2.23.4 (RSN capabilities).
* Confirm the cipher suite selection.

Installing the PTK, and where applicable the (#1349)group keys GTK causes the MAC to encrypt and decrypt all subsequent MSDUs irrespective of their path through the (#1909)Controlled or Uncontrolled Ports. (11ba)Installing the WTK when WUR frame protection is negotiated (#1349)causes the MAC to integrity protect subsequent individually addressed WUR Wake-up frames at the AP or to validate subsequent individually addressed WUR Wake-up frames at the non-AP STA.

Upon successful completion of the 4-way handshake or the encrypted (Re)Association Request/Response frame exchange (#980), the Authenticator and Supplicant have authenticated each other; and the IEEE 802.1X Controlled Ports are unblocked to permit general data traffic (#1349)as described in 12.7 (Keys and key distribution).

(…existing texts)

**TGbi Editor: *Modify 6.4 as follows:***

**6.4 Table of MLME SAP interfaces**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service Name | MLME-XXX | Type | References | Comments |
| … | …. | … | …. | …. |
| EAPOL | EAPOL | 5 | 12.7.2 (EAPOL-Key frames), 12.16.5 (IEEE 802.1X authentication utilizing Authentication frames)(#980) |  |

**TGbi Editor: *Instruction: Modify 12.6.1.2.2 as follows***

* Security association in an ESS

***Change item d) of the second paragraph (not all shown) as follows:***

* The last step is key management. The authentication process, whether SAE authentication utilizing Authentication frames, or~~or~~ FILS authentication utilizing Authentication frames, or IEEE 802.1X authentication utilizing Authentication frames, or IEEE 802.1X authentication utilizing Data frames post association, or the OWE exchange utilizing association frames creates cryptographic keys shared between the cryptographic endpoints—the AP and STA, or the IEEE 802.1X AS and the STA, when using SAE/FILS/OWE or IEEE Std 802.1X, respectively. When using IEEE Std 802.1X, the AS transfers these keys to the AP, and if encryption of (Re)Association Request/Response frames is not used, the AP and STA uses one of the key confirmation handshakes, e.g., the 4-way handshake or FT 4-way handshake, to complete security association establishment. When using SAE authentication or OWE there is no AS and therefore no key transfer; if encryption of (Re)Association Request/Response frames is not used, the 4-way handshake is performed directly between the AP and STA. The key confirmation handshake indicates when the link has been secured by the keys and is ready to allow normal data traffic and protected robust Management frames. When FILS authentication is performed or if encryption of (Re)Association Request/Response frames is used, the key confirmation is performed using association frames. Hence, no additional handshake is necessary.(#649)

**TGbi Editor: *Instruction: Modify 12.6.1.1 as follows***

* Security association definitions
* General

***Change second paragraph (not all lines shown) by modifying a sub-bullet and adding a sub-bullet as follows:***

A security association is a set of policy(ies) and key(s) used to protect information. The information in the security association is stored by each party of the security association, needs to be consistent among all parties, and needs to have an identity. The identity is a compact name of the key and other bits of security association information to fit into a table index or an MPDU. The following types of security associations are supported by an RSNA STA:

* PTKSA: A result of a successful 4-way handshake, FT 4-way handshake, FT authentication sequence, FILS authentication, ~~or~~ PASN authentication, EDPKE authentication, or authentication followed by the encrypted (Re)Association Request/Response frame exchange.(#649)
* PGTKSA: A result of a successful group key handshake, the Reassociation Response frame of the fast BSS transition protocol, the encrypted Reassociation Response frame specified in 12.16.6.2 (MLO procedure), or successful FILS authentication.

**TGbi Editor: *Instruction: Modify 12.16.5 as follows***

* IEEE 802.1X authentication utilizing Authentication frames

If an AP sets the IEEE 802.1X Authentication Utilizing Authentication Frame Support field in the RSNXE that it transmits to 1, then a non-AP STA (originator) with dot11EDPIEEE8021XAuthenticationUtilizingAuthenticationFrameActivated equal to true may signal its Supplicant to authenticate with the AP (responder) using IEEE Std 802.1X-2020 utilizing Authentication frames.

If any AP affiliated with an AP MLD sets the IEEE 802.1X Authentication Utilizing Authentication Frame Support field in the RSNXE that it transmits to 1, then a non-AP MLD (originator) with dot11EDPIEEE8021XAuthenticationUtilizingAuthenticationFrameActivated equal to true may signal its Supplicant to authenticate with the AP MLD (responder) using IEEE Std 802.1X-2020 utilizing Authentication frames by transmitting the Authentication frames to the AP through a non-AP STA affiliated with the non-AP MLD.

When the originator is a non-AP MLD and the responder is an AP MLD, the RA field of an Authentication frame in response to an Authentication frame from the peer shall be set to the TA field of the Authentication frame from the peer.

If an originator chooses to initiate IEEE 802.1X authentication utilizing Authentication frames, it first selects an IEEE 802.1X AKM that is supported by the responder.

The originator then shall construct(#676) the first Authentication frame of the exchange as follows:

* Authentication Algorithm Number field is set to 8 (IEEE 802.1X authentication).
* Authentication Transaction Sequence Number field is set to 1.
* The Encapsulation field carries an EAPOL PDU from the PAE(#980).
* Include the AKM Suite Selector element indicating the selected IEEE 802.1X AKM.

The originator sends the first Authentication frame to the responder.

Upon receiving the first Authentication frame, the responder shall(#676):

* Validate(#676) that the AKM indicated in AKM Suite Selector element is an IEEE 802.1X AKM. Otherwise, processing status is set to STATUS\_INVALID\_AKMP.(#965)
* Validate(#676) that the selected IEEE 802.1X AKM indicated in AKM Suite Selector element is supported. Otherwise processing status is set to STATUS\_INVALID\_AKMP.
* If the validation is successful, extract(#676)(#965) an EAPOL PDU from the Encapsulation field, and forward the EAPOL PDU to the PAE.(#980)

The responder then shall construct(#676) the second Authentication frame of the exchange as follows:

* Authentication Algorithm Number field is set to 8 (IEEE 802.1X authentication).
* Authentication Transaction Sequence Number field is set to 2.
* Status Code field indicates the processing status.
* The Encapsulation Length field indicates 0 if the status is set to STATUS\_INVALID\_AKMP.
* The Encapsulation field (if present) carries an EAPOL PDU from the PAE.(#980)
* Include(#676) the AKM Suite Selector element indicating the same IEEE 802.1X AKM indicated in the first Authentication frame.

Once the processing is complete, the responder sends the second Authentication frame to the originator. If the processing status returned in the frame was not SUCCESS, the responder shall terminate the authentication.

Upon receiving the second Authentication frame, the originator shall(#676):

* Validate that the Status Code field is SUCCESS. Otherwise, the originator shall terminate the authentication. (#965)
* Validate(#676) that the AKM indicated in AKM Suite Selector element is the same as the one indicated in the first Authentication frame. Otherwise,(#676) processing status is set to STATUS\_INVALID\_AKMP.
* If the validation is successful, extract(#676) an EAPOL PDU from the Encapsulation field (if present), (#965) and forward the EAPOL PDU to the PAE.(#980)

The originator then shall construct(#676) the third Authentication of the exchange as follows:

* Authentication Algorithm Number field is set to 8 (IEEE 802.1X authentication).
* Authentication Transaction Sequence Number field is set to 3.
* Status Code field indicates the processing status.
* The Encapsulation Length field indicates 0 if the status is set to STATUS\_INVALID\_AKMP.
* The Encapsulation field (if present) carries an EAPOL PDU(#965) from the PAE.(#980)

Once the processing is complete, the originator sends the third Authentication frame to the responder. If the processing status returned in the frame was not SUCCESS, the originator shall terminate the authentication.

* Upon receiving the Authentication frame with Authentication Transaction Sequence Number field set to a value that(#848) is larger than or equal to 3, the originator or the responder:Extract(#676) an EAPOL PDU (if present) from the Encapsulation field, and forward the EAPOL PDU to the PAE.(#980)
* Validate that the Status Code field is SUCCESS or 802\_1\_X\_AUTH\_SUCCESS. Otherwise, the originator shall terminate the authentication after forwarding the EAPOL PDU (if present) to the PAE. (#965)

If needed by the EAP method, the(#965) originator or the responder then shall construct(#676) the Authentication frame of the exchange in response to the Authentication frame with Authentication Transaction Sequence Number field set to a value that(#848) is larger than or equal to 3, as follows:

* Authentication Algorithm Number field is set to 8 (IEEE 802.1X authentication).
* Authentication Transaction Sequence Number field is set to the value of the Authentication Transaction Sequence Number field of the Authentication frame being responded to(#848, #Ed) +1.
* (#965)
* The Encapsulation field (if present) carries an EAPOL PDU from the PAE.(#980)
* (#965)
* If the EAPOL PDU is EAP Success, the Status Code field is set to 802\_1\_X\_AUTH\_SUCCESS. (#965)
* If the EAPOL PDU is EAP Failure, the Status Code field is set to 802\_1\_X\_AUTH\_FAILED. (#965)

NOTE —The number of Authentication frame exchanges depends on the EAP method in use.(#Ed)

Once the processing is complete, the originator or the responder sends the Authentication frame in response to the Authentication frame with Authentication Transaction Sequence Number field set to a value that is larger than or equal to 3,(#848) to its peer (if needed by the EAP method). If the processing status returned in the frame was not SUCCESS or 802\_1\_X\_AUTH\_SUCCESS, the originator or the responder shall terminate the authentication. (#965)

**TGbi Editor: *Instruction: Modify 9.4.1.9 as follows***

* Status Code field

***Insert the following new rows to Table 9-80 (Status codes) while maintaining the numerical order and updating the reserved range (not all lines shown):***

* ANA assignment and update are done
* Status codes

|  |  |  |
| --- | --- | --- |
| Status code | Name | Meaning |
| … |  |  |
| 145 | SUCCESS\_SIMILAR\_EPOCH | The request to join or create a group EDP(#1012) epoch is successful but the epoch parameters are not exactly those(#22) requested. |
| 146 | SUCCESS\_ALREADY\_EXISTING\_EPOCH(#192) | The STA has successfully joined the requested group EDP(#1012) epoch. The EDP group already exists, no new EDP group is created.(#192) |
| 147 | FAILURE\_MAX\_NUM\_EPOCH\_REACHED(#931) | Failure to create a group EDP(#1012) epoch because the maximum number of group EDP(#1012) epochs at the AP has been reached. |
| 148 | SUCCESS\_AID\_LIST\_PARTIALLY\_STORED | The AID List is too large and the CPE non-AP MLD has stored it only partially. |
| 149 | FAILURE\_AID\_LIST\_NOT\_STORED | No AID value has been stored. |
| 150 | FAILURE\_AID\_STORAGE\_TOO\_SMALL | The request to join or create a group EDP(#1012) epoch has failed, because the AID storage of the non-AP MLD is too small. |
| 151 | NO\_ASSIGNED\_AID | The non-AP MLD has no AID value for the current group EDP(#1012) epoch. |
| <ANA> | 802\_1\_X\_AUTH\_FAILED | IEEE 802.1X authentication failed.(#965) |
| <ANA> | 802\_1\_X\_AUTH\_SUCCESS | The EAPOL PDU carried in the Authentication frame is EAP SUCCESS. (#965) |