802.11bi Draft Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proposed spec texts for 802.1X authentication utilizing authentication frame | | | | |
| Date: 2023-01-15 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Po-Kai Huang | Intel |  |  | po-kai.huang@intel.com |
| Ido Ouzieli |  |  |  |
| Johannes Berg |  |  |  |
| Ilan Peer |  |  |  |
| Robert Stacey |  |  |  |

Abstract

This submission proposes spec text based on the following passed requirement.

*11bi shall define a mechanism for a CPE Client and CPE AP to carry 802.1X EAPOL PDUs in Authentication frames to perform IEEE 802.1X authentication.*

Revision History:

* Rev 0: Initial version of the document
* Rev 1: Editorial revision based on comments received offline.
* Rev 2: Revision based on discussion during Jan IEEE meeting to use field length and field for encapsulation rather than wrapped data element and directly have capability bit in RSNXE.
* Rev 3: Editorial revision.
* Rev 4: Revision based on the suggestion from Kurt. Changes are marked with green.

***Editing instructions formatted like this are intended to be copied into the TGbi 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 or insert 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.***

**TGbi Editor: *Instruction: Insert new rows in Table 9-363 in 9.4.2.241 RSNXE as shown below***

9.4.2.241 RSNXE

(…existing texts…)

|  |  |  |
| --- | --- | --- |
| * **Extended RSN Capabilities field** | | |
| **Bit** | **Information** | **Notes** |
| <ANA> | IEEE 802.1X Authentication Utilizing Authentication Frame Support | A EDP STA sets the IEEE 802.1X Authentication Utilizing Authentication Frame Support subfield to 1 if dot11EDPIEEE8021XAuthenticationUtilizinAuthenticationFrameActivated is true. Otherwise, this subfield is set to 0. |

…(existing texts)….

**TGbi Editor: *Instruction: Modify 9.3.3.11 Authentication frame format as shown below (track change on)***

* Authentication frame format

(…existing texts….)

***Insert the new rows in Table 9-68—Authentication frame body:***

|  |  |  |
| --- | --- | --- |
| * **Authentication frame body** | | |
| **Order** | **Information** | **Notes** |
| 1 | Authentication algorithm number |  |
| 2 | Authentication transaction sequence number |  |
| 3 | Status code | The status code information is reserved in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 4 | Finite Cyclic Group | An unsigned integer indicating a finite cyclic group as described in **9.4.1.xx Length of Encapsulation field**. This is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 5 | Anti-Clogging Token | A random bit string used for anti-clogging purposes as described in 12.4.6 (Anti-clogging tokens). This is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 6 | Send-Confirm | A binary encoding of an integer used for anti-replay purposes as described in 12.4.7.4 (Encoding and decoding of SAE Confirm messages). This is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 7 | Scalar | An unsigned integer encoded as described in 12.4.7.3 (Encoding and decoding of SAE Commit messages). This is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 8 | FFE(#312) | An element in a finite field encoded as described in 12.4.7.3 (Encoding and decoding of SAE Commit messages). This is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 9 | Confirm | An unsigned integer encoded as described in 12.4.7.4 (Encoding and decoding of SAE Confirm messages). This is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 10 | Length of Encapsulation field | The Length of Encapsulation field indicates the number of octets of the Encapsulation field. This is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames |
| 11 | Encapsulation field | The field is used to carry EAPOL PDU. This is present only when the Length of Encapsulation field indicates a non-zero value. |
| 12 | Challenge text | A Challenge Text element is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 13 | RSN | An RSNE is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 14 | Mobility Domain | An MDE is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 15 | Fast BSS Transition | An FTE is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 16 | Timeout Interval (reassociation deadline) | A TIE containing the reassociation deadline interval is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 17 | RIC | A resource information container, containing a variable number of elements, is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 18 | Multi-band | The Multi-band element is optionally present if dot11MultibandImplemented is true. |
| 19 | Neighbor Report | One or more Neighbor Report elements are present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 20 | FILS Nonce | The FILS Nonce element is present in FILS Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 21 | FILS Session | The FILS Session element is present in FILS Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 22 | Wrapped Data | The Wrapped Data element is present in FILS Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 23 | Association Delay Info | The Association Delay Info element is present in FILS Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 24 | Password Identifier | The Password Identifier element is optionally present in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 25 | Rejected Groups | The Rejected Groups element is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 26 | Anti-Clogging Token Container | The Anti-Clogging Token Container element is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| 27(M67) | AKM Suite Selector | The AKM Suite Selector element is present only in certain Authentication frames as defined in Presence of fields and elements in Authentication frames . |
| Last | Vendor Specific | One or more Vendor Specific elements are optionally present. These elements follow all other elements. |

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

|  |  |  |  |
| --- | --- | --- | --- |
| * Presence of fields and elements in Authentication frames | | | |
| Authentication algorithm | Authentication transaction sequence number | Status code | Presence of fields and elements  from order 4 onward |
| IEEE 802.1X authentication | 1 | Reserved | The Length of Encapsulation field is present.  The Encapsulation field is present only when the Length of Encapsulation field indicates a non-zero value.  The AKM Suite Selector element is present. |
| IEEE 802.1X authentication | 2 | NOT\_SUCCESS | The Length of Encapsulation field is present.  The Encapsulation field is present only when the Length of Encapsulation field indicates a non-zero value. |
| IEEE 802.1X authentication | 2 | SUCCESS | The Length of Encapsulation field is present.  The Encapsulation field is present only when the Length of Encapsulation field indicates a non-zero value.  The AKM Suite Selector element is present. |
| IEEE 802.1X authentication | 3 | NOT\_SUCCESS | The Length of Encapsulation field is present.  The Encapsulation field is present only when the Length of Encapsulation field indicates a non-zero value. |
| IEEE 802.1X authentication | 3 | SUCCESS | The Length of Encapsulation field is present.  The Encapsulation field is present only when the Length of Encapsulation field indicates a non-zero value. |
| IEEE 802.1X authentication | >3 | Status | The Length of Encapsulation field is present.  The Encapsulation field is present only when the Length of Encapsulation field indicates a non-zero value. |

**TGbi Editor: *Instruction: Insert the following subclauses as the last subclause of 9.4.1***

**9.4.1.xx Length of Encapsulation field**

The Length of Encapsulation field indicates the number of octets of the Encapsulation field.

|  |  |
| --- | --- |
|  | Length of Encapsulation field |
| Octets: | 2 |
| * **Length of Encapsulation field format** | |

**9.4.1.xx Encapsulation field**

The Encapsulation field carries the EAPOL PDU.

|  |  |
| --- | --- |
|  | Encapsulation field |
| Octets: | variable |
| * **Encapsulation field format** | |

**TGbi Editor: *Instruction: Insert 12.13.2 IEEE 802.1X authentication utilizing Authentication frames as shown below***

**12.13.2 IEEE 802.1X authentication utilizing Authentication frames**

If an AP sets the IEEE 802.1X Authentication Utilizing Authentication Frame Support subfield of the CPE Capabilities Information subfield in the RSNXE that it transmits to 1, then a non-AP STA (authentication originator) that supports IEEE 802.1X Authentication Utilizing Authentication Frame may signal its Supplicant to authenticate with the AP (authentication responder) using IEEE Std 802.1X-2010 utilizing Authentication frames.

If any AP affiliated with an AP MLD sets the IEEE 802.1X Authentication Utilizing Authentication Frame Support subfield of the CPE Capabilities Information subfield in the RSNXE that it transmits to 1, then a non-AP MLD (authentication originator) that supports IEEE 802.1X Authentication Utilizing Authentication Frame may signal its Supplicant to authenticate with the AP MLD (authentication responder) using IEEE Std 802.1X-2010 utilizing Authentication frames by transmitting the Authentication frames to the AP through a non-AP STA affiliated with the non-AP MLD.

When the authentication originator is non-AP MLD and the authentication responder is 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 authentication originator chooses to initiate 802.1X authentication utilizing Authentication frames, it first selects one 802.1X AKM that is supported by the authentication responder.

The authentication originator then constructs the first Authentication frame of the exchange as follows:

* Authentication Algorithm Number field set to 8 (IEEE 802.1X authentication);
* Authentication Transaction Sequence Number field set to 1;
* The Length of Encapsulation field indicates the length of the Encapsulation field;
* The encapsulation field carries EAPOL PDU;
* Including the AKM Suite Selector element indicating the selected 802.1X AKM;

The authentication originator sends the first Authentication frame to the authentication responder.

Upon receiving the first Authentication frame, the authentication responder:

* Validates that the selected 802.1X AKM indicated in AKM Suite Selector element is supported. Otherwise processing status is set to STATUS\_INVALID\_AKMP
* Extracts EAPOL PDU from the Encapsulation field, and processes it according to the behavior  
  described in a later subclause specific to the AKMP;

The authentication responder then constructs the second Authentication frame of the exchange as follows:

* Authentication Algorithm Number field set to 8 (IEEE 802.1X authentication);
* Authentication Transaction Sequence Number field set to 2;
* Status code indicating the processing status
* The Length of Encapsulation field indicates the length of the Encapsulation field. The Length of Encapsulation field indicates 0 if the status is set to STATUS\_INVALID\_AKMP;
* The encapsulation field (if present) carries EAPOL PDU;
* Includes the AKM Suite Selector element indicating the selected 802.1X AKM indicated in the first Authentication frame

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

Upon receiving the second Authentication frame, the authentication originator:

* Validates that 802.1X AKM indicated in AKM Suite Selector element is the same as the one indicated in the first Authentication frame. Otherwise processing status is set to STATUS\_INVALID\_AKMP
* Extracts EAPOL PDU from the Encapsulation field, and processes it according to the behavior  
  described in a later subclause specific to the AKMP;

The authentication originator then constructs the third Authentication of the exchange as follows:

* Authentication Algorithm Number field set to 8 (IEEE 802.1X authentication);
* Authentication Transaction Sequence Number field set to 3;
* Status code indicating the processing status
* The Length of Encapsulation field indicates the length of the Encapsulation field. The Length of Encapsulation field indicates 0 if the status is set to STATUS\_INVALID\_AKMP;

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

Upon receiving the Authentication frame with Authentication Transaction Sequence Number field set to X, where X is larger than or equal to 3, the authentication originator or the authentication responder:

* Extracts EAPOL PDU from the Encapsulation field, and processes it according to the behavior  
  described in a later subclause specific to the AKMP;

The authentication originator or the authentication responder then constructs the Authentication frame of the exchange in response to the Authentication frame with Authentication Transaction Sequence Number field set to X (if needed by the EAP method) as follows:

* Authentication Algorithm Number field set to 8 (IEEE 802.1X authentication);
* Authentication Transaction Sequence Number field set to X+1;
* Status code indicating the processing status
* The Length of Encapsulation field indicates the length of the Encapsulation field.
* The encapsulation field (if present) carries EAPOL PDU;

Once the processing is complete, the authentication originator or the authentication responder sends the Authentication frame to its peer (if needed by the EAP method). If the processing status returned in the frame was not SUCCESS, the authentication originator or the authentication responder shall terminate the authentication.

**TGbi Editor: *Instruction: Insert the following definitions to 3.2 Definitions specific to IEEE 802.11 (maintaining alphabetical order):***

**3.2 Definitions specific to IEEE 802.11**

**EAPOL-Start Authentication frame:** An Authentication frame that carries all or part of an IEEE 802.1X Extensible Authentication Protocol (EAP) over local area network (LAN) (EAPOL) protocol data unit (PDU) of type EAPOL-Start.(#238)(#1438)

**TGbi Editor: *Instruction: Modify 4.2.5 Interaction with other IEEE 802*® *layers as shown below (track change on)***

* **Interaction with other IEEE 802® layers**

IEEE Std 802.11 is required to appear to higher layers [logical link control (LLC) sublayer] as a general-purpose IEEE 802 LAN. This requires that the IEEE 802.11 network handle STA mobility within the MAC sublayer. To meet reliability assumptions (that LLC makes about lower layers), it is necessary for IEEE Std 802.11 to incorporate functionality that is untraditional for MAC sublayers.

In a robust security network association (RSNA), IEEE Std 802.11 provides functions to protect Data frames, IEEE Std 802.1X-2010 provides authentication and a Controlled Port, and IEEE Std 802.11 and IEEE Std 802.1X-2010 collaborate to provide key management. All STAs in an RSNA have a corresponding IEEE 802.1X entity that handles these services. This standard defines how an RSNA utilizes IEEE Std 802.1X-2010 to access these -services. Within IEEE Std 802.11, EAPOL PDUs are carried as MSDUs within one or more Data frames or as MMPDU within one or more Authentication frames, as described in Clause 12 of IEEE Std 802.1X-2010. Within this standard, Data frames used for this purpose are generally referred to as *EAPOL-Key frames*, *EAPOL-Key request frames*, and *EAPOL-Start frames*. Authentication frames used for this purpose are generally referred to as *EAPOL-Start Authentication frames*

**TGbi Editor: *Instruction: Modify 4.5.3.3 Association as shown below (track change on)***

**4.5.3.3 Association**

…(existing texts)….

Within a robust security network (RSN), association is handled differently. In an RSNA, the IEEE 802.1X Port determines when to allow data traffic across an IEEE 802.11 link. A single IEEE 802.1X Port maps to one association, and each association maps to an IEEE 802.1X Port. An IEEE 802.1X Port consists of an IEEE 802.1X Controlled Port and an IEEE 802.1X Uncontrolled Port. 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 Authentication frame exchanges carrying EAPOL PDU (if using IEEE 802.1X authentication utilizing Authentication frame) and the IEEE 802.1X Uncontrolled Port. Once the AKM completes successfully, data protection is enabled to prevent unauthorized access, and the IEEE 802.1X Controlled Port unblocks to allow protected data traffic. IEEE 802.1X Supplicants and Authenticators exchange protocol information via the IEEE 802.1X Uncontrolled Port or Authentication frames carrying EAPOL PDU. It is expected that most other protocol exchanges use the IEEE 802.1X Controlled Ports. However, a given protocol might need to bypass the authorization function and make use of the IEEE 802.1X Uncontrolled Port.

…(existing texts)….

802.1X authentication procedure includes 4-way and EAPOL PDU exchange

Method 1: 4-way and EAPOL PDU exchange goes through uncontrolled port

Method 2: 4-way goes through uncontrolled port, and EAPOL PDU goes through authentication frame exchange

**TGbi Editor: *Instruction: Modify 4.5.4.2 Authentication as shown below (track change on)***

* **Authentication**

IEEE 802.11 authentication operates at the link level between IEEE 802.11 STAs. IEEE Std 802.11 does not provide either end-to-end (MSDU origin to MSDU destination) or user-to-user authentication.

IEEE Std 802.11 attempts to control LAN access via the authentication service. IEEE 802.11 authentication is a station service(#1296). This service might be used by all STAs to establish their identity to STAs with which they communicate, in both ESSs and IBSSs. If a mutually acceptable level of authentication has not been established between two STAs, an association is not established.

IEEE Std 802.11 defines six IEEE 802.11 authentication methods: Open System authentication, Shared Key authentication, FT authentication, simultaneous authentication of equals (SAE), IEEE 802.1X authentication, and FILS 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 **Error! Reference source not found.**). The IEEE 802.11 authentication mechanism also allows definition of new authentication methods.

An RSNA might support SAE authentication, and/or IEEE 802.1X authentication, and/or FILS authentication. An RSNA also supports authentication based on IEEE Std 802.1X-2010, or preshared keys (PSKs) after Open System authentication. 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.

In an RSNA, IEEE 802.1X Supplicants and Authenticators exchange protocol information via the IEEE 802.1X Uncontrolled Port or Authentication frames carrying EAPOL PDU. 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 Authentication frame exchanges carrying EAPOL PDU (if using IEEE 802.1X authentication utilizing Authentication frame) and the IEEE 802.1X Uncontrolled Port.

SAE authentication, IEEE 802.1X authentication, and Open System IEEE 802.11 authentication are used by STAs in an RSN for an infrastructure BSS. FILS authentication can be used by FILS STAs in an RSN for an infrastructure BSS. SAE authentication, Open System IEEE 802.11 authentication, or no IEEE 802.11 authentication is used in an RSN for an IBSS. SAE authentication is used for an MBSS. In an RSN, Shared Key IEEE 802.11 authentication is not used. In an RSN for DMG BSS, Open System IEEE 802.11 authentication is not used (12.2.4 (RSNA establishment)).

A STA might be authenticated with many other STAs at the same time.

Because the IEEE 802.1X authentication process could be time-consuming (depending on the authentication protocol in use), the authentication service can be invoked independently of the association service.

This type of preauthentication is typically done by a STA while it is already associated with an AP (with which it previously authenticated). IEEE Std 802.11 does not require that STAs preauthenticate with APs. However, authentication is required before an association establishment is complete.

If the authentication is left until reassociation time, this might impact the speed with which a STA reassociates between APs, limiting BSS-transition mobility performance. The use of preauthentication takes the authentication service overhead out of the time-critical reassociation process.

SAE authentication is performed prior to association and a STA can take advantage of the fact that it can be IEEE 802.11 authenticated to many APs simultaneously by completing the SAE protocol with any number of APs while still being associated to another AP. RSNA security can be established after association using the resulting shared key.

**TGbi Editor: *Instruction: Modify 4.10.2 IEEE 802.11 usage of IEEE Std 802.1X-2010 as shown below (track change on)***

* **IEEE 802.11 usage of IEEE Std 802.1X-2010**

IEEE Std 802.11 depends upon IEEE Std 802.1X-2010 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 may be transmitted in one or more IEEE 802.11 Data frames and passed via the IEEE 802.1X Uncontrolled Port or may be transmitted in one or more IEEE 802.11 Authentication frames and passed between the Supplicant and the Authenticator. 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 Authentication frame exchanges carrying EAPOL PDU (if using IEEE 802.1X authentication utilizing Authentication frame) and the IEEE 802.1X Uncontrolled Port. 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.

IEEE Std 802.11 depends upon IEEE Std 802.1X-2010 and various IEEE 802.11 protocols and handshakes, described in Clause 12 (Security) and Clause 13 (Fast BSS transition), to establish and change cryptographic keys. Keys are established after authentication has completed. Keys might change for a variety of reasons, including expiration of an IEEE 802.1X authentication timer, key compromise, danger of compromise, or policy.

**TGbi Editor: *Instruction: Modify 4.10.3.2 AKM operations with AS as shown below (track change on)***

* **AKM operations with AS**

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

* Prior to any use of IEEE Std 802.1X-2010, IEEE Std 802.11 assumes that the Authenticator and AS have established a secure channel. The security of the channel between the Authenticator and the AS is outside the scope of this standard.

Authentication credentials are distributed to the Supplicant and AS prior to association.

* A STA discovers the AP’s security policy through passively monitoring Beacon frames or through active probing(#1349). If IEEE 802.1X authentication is used, the EAP authentication process starts when the Authenticator sends the EAP-Request(#1349) or (#1836)the Supplicant sends the EAPOL-Start PDU (in one or more EAPOL-Start frames or EAPOL-Start Authentication frames). EAP messages pass between the Supplicant and AS via the Authenticator and Supplicant’s Uncontrolled Ports (#1349)as described in 12.7 (Keys and key distribution) or via the Authentication frame exchanges carrying EAPOL PDUs between the Authenticator and the Supplicant as described in 12.13.2 (IEEE 802.1X authentication utilizing Authentication frames).
* The Supplicant and AS authenticate each other and generate a PMK. The PMK is sent from the AS to the Authenticator over the secure channel (#1349)as described in 12.7 (Keys and key distribution).

A 4-way handshake or FT 4-way handshake utilizing (#1836)EAPOL-Key PDUs is initiated by the Authenticator to do the following:

* 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 9.4.2.24.4 (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, 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).

If the Authenticator later changes (#1349)one or more group keys, it sends the new group keys and sequence number to the Supplicant using the group key handshake to allow the Supplicant to continue to receive group addressed frames and, optionally, to transmit and receive individually addressed frames. (#1836)EAPOL-Key PDUs are used to carry out this exchange as described in 12.7 (Keys and key distribution).

**TGbi Editor: *Instruction: Modify 9.4.1.1 Authentication Algorithm Number field as shown below: (track change on)***

* Management and Extension frame body components
* Fields that are not elements
* Authentication Algorithm Number field

The Authentication Algorithm Number field indicates a single authentication algorithm. The length of the Authentication Algorithm Number field is 2 octets. The Authentication Algorithm Number field is shown in Authentication Algorithm Number field format. The following values are defined for authentication algorithm number:

Authentication algorithm number = 0: Open System

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 = 8: IEEE 802.1X authentication

Authentication algorithm number = 65 535: vendor specific use

NOTE—The use of this value implies that a Vendor Specific element is included with more information.

All other values of authentication algorithm number are reserved.

|  |  |
| --- | --- |
|  | Authentication Algorithm Number |
| Octets: | 2 |
| * Authentication Algorithm Number field format | |

**TGbi Editor: *Instruction: Modify 12.2.2 Security methods as shown below (track change on)***

* **Security methods**

Pre-RSNA security comprises the following algorithms and procedures:

* WEP, described in **Error! Reference source not found.**
* IEEE 802.11 entity authentication, described in **Error! Reference source not found.**

RSNA security comprises the following algorithms and procedures:

* TKIP, described in **Error! Reference source not found.**

(#1433)NOTE—TKIP is not however considered a robust security network mechanism.

* CCMP, described in **Error! Reference source not found.**
* GCMP, described in **Error! Reference source not found.**
* BIP, described in **Error! Reference source not found.**
* RSNA establishment and termination procedures, including use of IEEE 802.1X authentication, described in **Error! Reference source not found.** and 12.13.2 (IEEE 802.1X authentication utilizing Authentication frames), SAE authentication described in **Error! Reference source not found.**, and OWE described in IETF RFC 8110(#1084).
* Key management procedures, described in **Error! Reference source not found.**

The RSN operations in a CMMG BSS shall be the same as the RSN operations in a DMG BSS.

**TGbi Editor: *Instruction: Modify 12.2.4 RSNA establishment as shown below (track change on)***

* **RSNA establishment**

(#1084)An SME establishes an RSNA in one of seven ways:

* If an RSNA uses authentication negotiated over IEEE Std 802.1X or FILS authentication in an infrastructure BSS, an RSNA STA’s SME establishes an RSNA as follows:
* It identifies the AP as an RSNA AP from the AP’s Beacon, DMG Beacon, Announce, Information Response, or Probe Response frames.
* It shall invoke Open System or IEEE 802.1X authentication or FILS authentication if the STA is a non-DMG STA.
* It negotiates cipher suites during the association process, as described in **Error! Reference source not found.** and **Error! Reference source not found.**.
* It uses IEEE Std 802.1X-2010 to authenticate if 802.1X authentication is not performed before association, as described in **RSNA establishment in an infrastructure BSS** and **Error! Reference source not found.**, or uses FILS authentication to authenticate as described in **Error! Reference source not found.**.
* It establishes one or more temporal keys by executing a key management algorithm, using the protocol defined by **Error! Reference source not found.** or 13 (Fast BSS transition).
* It protects the data link by programming the negotiated cipher suites and the established temporal key into the MAC and then invoking protection. See, for example, **Error! Reference source not found.** for a description of the RSNA data protection -mechanisms.
* If the STAs negotiate management frame protection, the SME programs the TK and pairwise cipher suite into the MAC for protection of individually addressed robust Management frames. It also installs the IGTK and IPN for protection of group addressed robust Management frames.
* If beacon protection is enabled, the SME programs the BIGTK and BIPN into the MAC for the protection of Beacon frames.
* (11ba)If the STAs negotiate WUR frame protection, the SME programs the WTK and WTPN into the MAC for protection of individually addressed WUR Wake-up frames, and programs the WIGTK and WIPN into the MAC for the protection of broadcast and group addressed WUR Wake-up frames.

(…existing texts…)

**TGbi Editor: *Instruction: Modify 12.6.1.2.2 Security association in an ESS as shown below (track change on)***

* **Security association in an ESS**

In an ESS there are two cases:

* Initial contact between the STA and the ESS
* (#1600)BSS transition by the STA within the ESS

A STA and AP establish an initial security association via the following steps:

* The STA selects an authorized ESS by selecting among APs that advertise an appropriate SSID.
* The STA then performs IEEE 802.11 authentication followed by association to the chosen AP. Confirmation of security parameters takes place during association. (#1084)A non-DMG STA performing authentication with IEEE 802.1X after using Open System authentication or 802.1X authentication utilizing Authentication frames. A STA performing password-based authentication can use SAE authentication. A STA performing FILS uses FILS authentication. A non-DMG STA executing the OWE exchange uses the Open System authentication.

NOTE 1—It is possible for more than one PMKSA to exist. As an example, a second PMKSA might come into existence through PMKSA caching. A STA might leave the ESS and flush its cache. Before its PMKSA expires in the AP’s cache, the STA returns to the ESS and establishes a second PMKSA from the AP’s perspective.

NOTE 2—An attack altering the security parameters is detected by the key derivation procedure.

NOTE 3—IEEE 802.11 Open System authentication provides no security, but is included to maintain backward compatibility with the IEEE 802.11 state machine (see 11.3 (STA authentication and association)).

* SAE authentication, 802.1X authentication, and FILS authentication provides mutual authentication and derivation of a PMK. If Open System authentication is chosen instead, the Authenticator or the Supplicant initiates IEEE 802.1X authentication after (re)association. The EAP method used by IEEE Std 802.1X-2010 needs to support mutual authentication, as the STA needs assurance that the AP is a legitimate AP.

NOTE 4—Prior to the completion of IEEE 802.1X authentication and the installation of keys, the IEEE 802.1X Controlled Port in the AP blocks all Data frames. The IEEE 802.1X Controlled Port returns to the unauthorized state and blocks all Data frames before invocation of an MLME-DELETEKEYS.request primitive. The IEEE 802.1X Uncontrolled Port allows IEEE 802.1X frames to pass between the Supplicant and Authenticator. The Supplicant and Authenticator might also use Authentication frames to pass EAPOL PDUs.

(#1794)Supplicants without a Controlled Port compromise RSN security and shall not be used.

(#1794)NOTE 5—Although IEEE Std 802.1X-2010 does not require a Supplicant Controlled Port, this standard relies on the fact that the Supplicant has a Controlled Port in order to provide the needed level of security.

(#1084)NOTE 6—A secure IEEE 802.1X network cannot support promiscuous association, e.g., an unauthenticated operation of IEEE Std 802.11. A trust relationship is needed between the STA and the AS of the targeted SSID prior to association and secure operation, in order for the association to be trustworthy. Without some sort of prior relationship between the ESS and the STA, an attacker could deploy a rogue AP that would not be detected by the STA.

(#1084)NOTE 7—OWE networks are promiscuous and do not require any prior relationship between the ESS and STA. There is a guarantee of confidentiality of the communication between the STA and AP but there is no authentication. OWE networks are therefore not suited to situations where detection of a rouge access point is necessary or where networks require definitive identification of associated STAs.

* 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, (#1084)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 (#1084)SAE/FILS/OWE or IEEE Std 802.1X, respectively. When using IEEE Std 802.1X, the AS transfers these keys to the AP, and 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(#1084) there is no AS and therefore no key transfer; 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, the key confirmation is performed using association frames. Hence, no additional handshake is necessary.

When FT is not enabled, a STA BSS transitioning(#1600) within an ESS establishes a new PMKSA by one of the five schemes:

* In the case of (re)association followed by (#1084)IEEE 802.1X authentication, PSK authentication or the OWE exchange, the STA repeats the same actions as for an initial contact association, but its Supplicant also deletes the PTKSA when  it  roams  from  the  old  AP.  The  Supplicant  also  deletes  the PTKSA when it disassociates/deauthenticates from all BSSIDs in the ESS.
* In the case of SAE authentication or 802.1X authentication followed by (re)association, the STA repeats the same actions as for initial contact association, but the non-AP STA also deletes the PTKSA when it roams from the old AP. Note that a STA can take advantage of the fact that it can perform SAE authentication or 802.1X authentication to multiple APs while maintaining a single association with one AP, and then use any of the PMKSAs created during authentication to effect a fast BSS transition.
* A STA (AP) can cache PMKSAs for APs (STAs) in the ESS to which it has previously performed a full IEEE 802.1X authentication, SAE authentication or OWE exchange(#1084). If a STA wishes to roam to an AP for which it has cached one or more PMKSAs, it can include one or more PMKIDs in the RSNE of its (Re)Association Request frame. An AP that has retained the PMK for one or more of the PMKIDs can proceed with the 4-way handshake (see **Error! Reference source not found.**). If the STA fails to send a PMKID when the negotiated AKM uses IEEE 802.1X authentication, the STA and AP need to perform a full IEEE 802.1X authentication.
* A STA already associated with the ESS can request its IEEE 802.1X Supplicant to authenticate with a new AP before associating to that new AP. The normal operation of the DS via the old AP provides the communication between the STA and the new AP. The SME delays reassociation with the new AP until IEEE 802.1X authentication completes via the DS. If IEEE 802.1X authentication completes successfully, then PMKSAs shared between the new AP and the STA are cached, thereby enabling the possible usage of reassociation without requiring a subsequent full IEEE 802.1X authentication procedure.
* In the case of FILS, the STA may repeat the same actions as an initial contact and authentication, The STA may also use a cached PMKSA to authenticate (see **Error! Reference source not found.** and **Error! Reference source not found.**). A STA already associated with the ESS can initiate FILS authentication to multiple other APs while associated.

The MLME-DELETEKEYS.request primitive deletes the temporal key(s) established for the security association so that they cannot be used to protect subsequent IEEE 802.11 traffic. An SME uses this primitive when it deletes a PTKSA, GTKSA, IGTKSA, (11ba)BIGTKSA, WIGTKSA or TPKSA(#205).

**TGbi Editor: *Instruction: Modify 12.6.9 RSN management of the IEEE 802.1X Controlled Port as shown below (track change on)***

**12.6.9 RSN management of the IEEE 802.1X Controlled Port**

(#2128)In an RSN, this standard (#1794)relies on the fact that IEEE 802.1X Supplicants and Authenticators exchange protocol information via the IEEE 802.1X (#1909)Uncontrolled Port or Authentication frames carrying EAPOL PDU. The IEEE 802.1X Controlled Port is blocked from passing general data traffic between the STAs until an IEEE 802.1X authentication procedure completes successfully over the Authentication frame exchanges carrying EAPOL PDU (if using IEEE 802.1X authentication utilizing Authentication frame) and the IEEE 802.1X Uncontrolled Port. The security of an RSNA depends on this (#1794)being true.

In an infrastructure BSS or PBSS, if the STA associates with the AP or PCP, the STA indicates the IEEE 802.11 link is available by invoking the MLME-ASSOCIATE.confirm or MLME-REASSOCIATE.confirm primitive. This signals the Supplicant that the MAC has transitioned from the disabled to enabled state. At this point, the Supplicant’s Controlled Port is blocked, and communication of all non-IEEE-802.1X MSDUs sent or received via the port is not authorized.

In an infrastructure BSS or PBSS, if the AP or PCP associates with a STA, the AP or PCP indicates that the IEEE 802.11 link is available by invoking the MLME-ASSOCIATE.indication or MLME-REASSOCIATE.indication primitive. At this point the Authenticator’s Controlled Port corresponding to the STA’s association is blocked, and communication of all non-IEEE-802.1X MSDUs sent or received via the Controlled Port is not authorized.

In an IBSS the STA shall block all IEEE 802.1X ports at initialization. Communication of all non-IEEE 802.1X MSDUs sent or received via the Controlled Port is not authorized.

In a PBSS, if a STA chooses not to associate with the PCP, the STA shall block all IEEE 802.1X ports at initialization. Communication of all non-IEEE-802.1X MSDUs sent or received via the Controlled Port is not authorized.

(#1794)NOTE 1—IEEE Std 802.1X maintains each Controlled Port in a blocked state until the IEEE 802.1X state variables portValid and keyDone both become true. (#1794)This means that the IEEE 802.1X Controlled Port discards MSDUs sent across the IEEE 802.11 channel prior to the installation of cryptographic keys into the MAC. This protects (#1795)the upper layers from forged MSDUs written to the channel while it is still being -initialized.

(#2128)NOTE 2—This means that Data frames other than those containing EAPOL PDUs are discarded when received before the initial 4-way handshake (see **Error! Reference source not found.**) completes, and that unprotected Data frames (other than those containing retransmissions of the third message of the initial 4-way handshake (see **Error! Reference source not found.**) are discarded when received after the initial 4-way handshake completes.

The MAC does not distinguish between MSDUs for the Controlled Port, and MSDUs for the Uncontrolled Port. In other words, EAPOL-Start frames and EAPOL-Key frames are encrypted (#2128)and decrypted after invocation of the MLME-SETPROTECTION.request primitive.

(#2128)NOTE 3—An Authenticator might retransmit the third message of the 4-way handshake (see **Error! Reference source not found.**). In the initial 4-way handshake this third message (and the fourth message sent in response) will be unprotected and in a rekeying 4-way handshake the third (and the fourth) message will be protected with the old key. If the ends of the link do not both support extended Key IDs for individually addressed frames, the mechanism by which a Supplicant might accept and respond to this retransmission of the third message even though the MLME-SETKEYS.request and MLME-SETPROTECTION.request primitives have already been invoked is outside the scope of this standard.

(#1794)NOTE 4—IEEE Std 802.1X-2010 does not block the Controlled Port when authentication is triggered through IEEE 802.1X reauthentication. During reauthentication, an existing RSNA can protect all MSDUs exchanged between the STAs. Blocking MSDUs is not required during reauthentication over an RSNA.

(#2128)NOTE 5—EAPOL PDUs are not delivered to the Controlled Port (see IEEE Std 802.1X). This means that an AP does not forward EAPOL PDUs received from a STA to any other STA in the BSS or ESS, to the portal, to the attached bridge port, or to a local higher layer other than the PAE, and that a PCP does not forward EAPOL PDUs received from a STA to any other STA in the BSS or to a local higher layer other than the PAE.

(#2128)EAPOL PDUs shall be carried in individually addressed MSDUs or individually addressed Authentication frames .

If management frame protection is enabled, an Authenticator shall provide the IGTK in the 4-way handshake and the group key handshake unless the group management cipher suite selector for the BSS is 00-0F-AC:7.(#1924)

**TGbi Editor: *Instruction: Modify 12.6.10.1 General as shown below*** ***(track change on)***

* **RSNA establishment in an infrastructure BSS(#1084)**
* **General**

When establishing an RSNA in a non-FT environment or during an FT initial mobility domain association, a STA shall use IEEE 802.11 SAE authentication, FILS authentication, 802.1X authentication, or Open System authentication prior to (re)association.

SAE authentication is initiated when a STA’s MLME-SCAN.confirm primitive finds another AP within the ESS of which the STA is a member that advertises support for SAE in its RSNE.

FILS authentication is initiated when a STA’s MLME-SCAN.confirm primitive finds an AP that advertises support for FILS authentication in its RSNE.(#1084)OWE is initiated when a STA’s MLME-SCAN.confirm primitive finds another AP within the ESS of which the STA is a member that advertises support for OWE in its RSNE.

IEEE 802.1X authentication is initiated by any one of the following mechanisms:

* If a STA negotiates to use IEEE 802.1X authentication during (re)association, the STA’s management entity may respond to the MLME-ASSOCIATE.confirm (or indication) or MLME-REASSOCIATE.confirm (or indication) primitive by requesting the Supplicant (or Authenticator) to initiate IEEE 802.1X authentication if 802.1X authentication is not performed before (re)association. Thus, in this case, authentication is driven by the STA’s decision to associate and the AP’s decision to accept the association.
* If a STA’s MLME-SCAN.confirm primitive finds another AP within the ESS of which the STA is a member, a STA may signal its Supplicant to use IEEE Std 802.1X-2010 to preauthenticate with that AP. (see 12.6.10.2 Preauthentication and RSNA key management)

NOTE—A (#1600)BSS transitioning STA’s IEEE 802.1X Supplicant can initiate preauthentication by (#1836)sending an EAPOL-Start PDU (in one or more EAPOL-Start frames) via its old AP, through the DS, to a new AP.

* If a STA’s MLME-SCAN.confirm primitive finds another AP within the ESS of which the STA is a member that advertises support for 802.1X authentication in its RSNXE, a STA may signal its Supplicant to use IEEE Std 802.1X-2010 to authenticate with that AP. (see 12.13.2 IEEE 802.1X authentication utilizing Authentication frames)
* If a STA receives an IEEE 802.1X message, it delivers this to its Supplicant or Authenticator, which may initiate a new IEEE 802.1X authentication.