### IEEE P802.11 Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proposed Draft Specification for ML General, MLD Authentication, MLD Association, and ML Setup | | | | |
| Date: 2020-08-26 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Po-Kai Huang | Intel Corporation | 2200 Mission College Blvd, Santa Clara, CA 950542200 |  | po-kai.huang@intel.com |
| Robert Stacey |  |  |  |  |
| Daniel Bravo |  |  |  |  |
| Ido Ouzieli |  |  |  |  |
| Danny Alexander |  |  |  |  |
| Laurent Cariou |  |  |  |  |

Abstract

We propose the draft specification skeleton for MLD to help the creation of TGbe draft D0.1. This document proposes texts for the motions and SPs listed in the following pages.

Revisions:

* Rev 0: Initial version of the document.
* Rev 1: Revision based on the feedback received offline.

Multi-link device (MLD): A device that has more than one affiliated STA and has one MAC SAP to LLC, which includes one MAC data service.

NOTE 1 – The device can be logical.

NOTE 2 – It is TBD for a MLD to have only one STA.

NOTE 3 – Whether the WM MAC address of each STA affiliated with the MLD is the same or different is TBD.

[Motion 23, [5] and [108]]

AP multi-link device (AP MLD): A MLD, where each STA affiliated with the MLD is an AP.

Non-AP multi-link device (non-AP MLD): A MLD, where each STA affiliated with the MLD is a non-AP STA.

[Motion 24, [5] and [108]]

A MLD has a MAC address that singly identifies the MLD management entity.

For example, the MAC address can be used in multi-link setup between a non-AP MLD and an AP MLD.

[Motion 40, [5] and [108]]

[Motion 111, #SP0611-28, [15] and [115]]

802.11be supports that if different affiliated APs of an AP MLD have different MAC addresses, then different affiliated non-AP STAs of a non-AP MLD with more than one affiliated STA have different MAC addresses.

[Motion 112, #SP38, [15] and [115]]

The value of the RA/TA fields sent over-the-air in the MAC header of a frame is the MAC address of the STA affiliated with the MLD corresponding to that link.

[Motion 108, [24] and [116]]

The MAC address of each affiliated AP within an AP MLD shall be different from each other unless the affiliated APs cannot perform simultaneous TX/RX operation (e.g., due to near band in-device interference), in which case the MAC address properties are TBD.

NOTE – It is TBD whether we allow the operation of an AP MLD without simultaneous TX/RX operation.

[Motion 109, [24] and [116]]

802.11be defines a multi-link setup signaling exchange executed over one link initiated by a non-AP MLD with an AP MLD as follows:

* Capability for one or more links can be exchanged during the multi-link setup.
* The AP MLD serves as the interface to the DS for the non-AP MLD after successful multi-link setup.

NOTE 1 – The link identification is TBD.

NOTE 2 – Details for non-infrastructure mode of operation TBD.

[Motion 25, [5] and [120]]

802.11be supports the following:

* Existing frames are reused for discovering APs that are affiliated with AP MLD.
* Association Request and Association Response frames are reused for multi-link setup.
* NOTE: After association, new signaling to query AP link specific parameters or AP MLD parameters by using Protected Management Frames (PMF) encrypted Management frames is TBD.

[Motion 115, #SP76, [12] and [121]]

802.11be shall define a mechanism to teardown an existing multi-link setup agreement.

[Motion 70, [24] and [122]]

802.11be supports the following:

* Reuse disassociation frame for multi-link teardown.
* Reuse authentication frame for multi-link SAE exchange and multi-link Open System authentication.

[Motion 115, #SP88, [12] and [123]]

Between two MLDs, 802.11be supports using the MLD MAC addresses to derive PMK under SAE method and PTK in 802.11be SFD.

[Motion 112, #SP40, [15] and [124]]

TGbe shall define a multi-link resetup mechanism to resetup with another AP MLD or changing configuration of existing multi-link setup with an AP MLD.

* Reassociation Request/Response frame is used for this purpose.

[Motion 115, #SP86, [12] and [125]]

When a non-AP MLD that has multi-link setup with current AP MLD sends a Reassociation Request frame to a new AP MLD, AP MLD MAC address of the current AP MLD is used in Current AP Address field of the frame.

[Motion 115, #SP87, [12] and [125]]

When a STA of a non-AP MLD that has multi-link setup with current AP MLD sends a Reassociation Request frame to a new AP that is not affiliated with an AP MLD, AP MLD MAC address of the current AP MLD is used in Current AP Address field of the frame.

* Note: Only the STA that sends the Reassociation Request frame can associate with the new AP.

[Motion 115, #SP94, [12] and [125]]

802.11be supports the following:

* An AP that is part of an AP MLD that supports SAE authentication shall include the MLD address in beacon and probe response frames it transmits.
* EHT MLD shall indicate its MLD MAC address during authentication request/response exchange.

[Motion 115, #SP89, [12] and [123]]

An EHT MLD shall indicate its MLD MAC address during ML setup.

[Motion 112, #SP32, [15] and [128]]

**Straw poll #192**

Do you support the following?

* After a successful multi-link (re)setup between a non-AP MLD and an AP MLD, the non-AP MLD is in associated state and is (re)associated with the AP MLD, and the non-AP MLD to the AP MLD mapping is provided to the DS by the AP MLD
  + For each setup link, the corresponding non-AP STA affiliated with the non-AP MLD is in the same associated state as the non-AP MLD and is associated with the corresponding AP affiliated with the AP MLD, without providing the corresponding non-AP STA to the corresponding AP mapping to the DS, and enables the functionalities between a non-AP STA and its associated AP unless the functionalities have been extended to MLD level and specified otherwise.
* After a successful multi-link tear down between a non-AP MLD and an AP MLD, the non-AP MLD is in unassociated state and is disassociated with the AP MLD
  + All the non-AP STAs affiliated with the non-AP MLD are in the unassociated state
* NOTE - A non-AP MLD needs to perform successful multi-link (re)setup with AP MLD before the non-AP MLD is allowed to send/receive MSDU(s) via the AP MLD to the DS ***[#SP192]***

[20/0669r1 (MLD transition, Po-Kai Huang, Intel), SP, Y/N/A: 48/2/31]

**Straw poll #196**

**Do you support to add the following in 11be SFD in R1:**

* Define ML transition\* as follows:
  1. A non-AP MLD movement from being associated with one AP MLD in one ESS to be reassociated with another AP MLD within the same ESS.
  2. A non-AP MLD movement from being associated with one AP MLD in one ESS to become a non-AP STA that is reassociated with an AP within the same ESS.
  3. A non-AP STA movement from being associated with one AP in one ESS to become a non-AP MLD that is reassociated with an AP MLD with the same ESS.
* Define fast ML transition\* as follows:
  1. A ML transition\* that establishes the state necessary for data connectivity before the reassociation rather than after the reassociation.
* Note\* – the name can be changed

**Straw poll #197**

**Do you support to reuse existing frame exchange of over-the-air fast BSS transition (FT) for fast ML transition in R1?**

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 TGbe 0.1 Draft. This introduction is not part of the adopted material.

***Editing instructions formatted like this are intended to be copied into the TGbe D0.1 Draft (i.e. they are instructions to the 802.11 editor on how to merge the text with the baseline documents).***

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

**Discussion:** *None.*

**Propose:**

**Part I:**

* Definitions, acronyms, and abbreviations
* Definitions specific to IEEE 802.11

***TGbe editor: Insert the following definitions (maintaining alphabetical order) except green tag:***

**access point (AP) multi-link device (MLD):** A MLD, where each station (STA) affiliated with the MLD is an AP.(#Motion 24)

**multi-link device (MLD):** A device that is a logical entity and has more than one affiliated STA and has one medium access control (MAC) service access point (SAP) to logical link control (LLC), which includes one MAC data service. (#Motion 23)

**non-access point (non-AP) multi-link device (MLD):** A MLD, where each station (STA) affiliated with the MLD is a non-AP STA. (#Motion 24)

* Abbreviations and acronyms

***TGbe editor: Insert the following acronym definitions (maintaining alphabetical order) except green tag:***

ML multi-link (#Motion 23)

MLD multi-link device (#Motion 23)

MLDME multi-link device management entity (Motion 111, #SP0611-28)

***TGbe editor: Insert clause 33.3.1 as follows except green tag:***

33. Extreme High Throughput (EHT) MAC specification

**33.3 Multi-link operation**

**33.3.1 Multi-link device addressing**

A MLD has a MLD MAC address that singly identifies the MLD management entity. (#Motion 111, #SP0611-28)

The MAC address of each AP affiliated with an AP MLD shall be different from each other unless the affiliated APs cannot perform simultaneous TX/RX operation (e.g., due to near band in-device interference), in which case the MAC address properties are TBD. (#Motion 109)

NOTE – It is TBD whether we allow the operation of an AP MLD without simultaneous TX/RX operation. (#Motion 109)

If each AP affiliated with an AP MLD has different MAC addresses, then each non-AP STA affiliated with an associated non-AP MLD shall have different MAC addresses. (#Motion 112, #SP38)

The value of the Address 2 field (TA) field in the MAC header of a frame sent over-the-air shall be the MAC address of the transmitting STA affiliated with the MLD corresponding to that link except the Individual/Group bit, which is set to 1 when the TA field value is a bandwidth signaling TA and set to 0 otherwise.(# Motion 108)

The value of the Address 1 (RA) field in the MAC header of an individually addressed frame sent over-the-air shall be the MAC address of the receiving STA affiliated with the MLD corresponding to that link. (# Motion 108)

**33.3.2 Multi-link (re)setup procedure**

Before a non-AP MLD performs multi-link (re)setup with an AP MLD, the non-AP MLD and AP MLD shall follow MLD authentication as described in 11.3 (STA/MLD authentication and association). (#Motion 115 #SP88)

For a non-AP MLD to perform multi-link (re)setup with an AP MLD, the non-AP MLD and the AP MLD shall exchange (Re)Association Request/Response frames and shall follow MLD (re)association procedure as described in 11.3 (STA/MLD authentication and association). (#Motion 25, #SP192, #Motion 115 #SP76, #Motion 115 #SP86)

In the (Re)Association Requeust frame, the non-AP MLD indicates the links that are requested for setup as described in 33.3.x.x (Usage and Rules of ML element in the context of multi-link setup). (Motion 122, #SP133)

After successful multi-link (re)setup between a non-AP MLD and an AP MLD, the non-AP MLD and the AP MLD setup links for multi-link operation, and the non-AP MLD is (re)associated with the AP MLD. (#SP192, Motion 25)

For each setup link, the corresponding non-AP STA affiliated with the non-AP MLD is in the same associated state as the non-AP MLD and is associated with the corresponding AP affiliated with the AP MLD, without providing the corresponding non-AP STA to the corresponding AP mapping to the DS, and enables the functionalities between a non-AP STA and its associated AP unless the functionalities have been extended to MLD level and specified otherwise.(#SP192)

**33.3.3 Multi-link tear down procedure**

To tear down the setup links between a non-AP MLD and an AP MLD, one of the non-AP STA or AP affiliated with the non-AP MLD or the AP MLD, respectively, shall send disassociation frame to the AP or non-AP STA affiliated with the AP MLD or non-AP MLD, respectively, on the corresponding link that is enabled and shall follow MLD disassociation procedure as described in 11.3 (STA/MLD authentication and association). (#Motion 70, #Motion 115 #SP88)

After multi-link teardown, all the non-AP STAs affiliated with the non-AP MLD are in the same unassociated state as the non-AP MLD. (#SP192)

***TGbe editor: Modify 4.5.3.1 as follows except green tag:***

* Connectivity-related services(11ak)(#120)
* General

The primary purpose of a MAC sublayer is to transfer MSDUs between MAC sublayer entities. The information required for the distribution system service(#120) to operate is provided by the association services. Before an MSDU can be handled by the distribution system service(#120), a STA or a MLD is “associated.” (#SP192)

To understand the concept of association, it is necessary first to understand the concept of mobility.

***TGbe editor: Modify 4.5.3.2 as follows except green tag:***

* Mobility types (#SP196, #197)

The three transition types of significance to this standard that describe the mobility of STAs within a net-work are as follows:

* ***No-transition:*** In this type, two subclasses that are usually indistinguishable are identified:
* Static—no motion.
* Local movement—movement within the PHY range of the communicating STAs, i.e., movement within a basic service area (BSA).
* ***BSS-transition:*** This type is defined as a STA movement from one BSS in one ESS to another BSS within the same ESS. A fast BSS transition is a BSS transition that establishes the state necessary for data connectivity before the reassociation rather than after the reassociation.
* ***ESS-transition:*** This type is defined as STA movement from a BSS in one ESS to a BSS in a different ESS. This case is supported only in the sense that the STA might move. Maintenance of upper-layer connections cannot be guaranteed by IEEE Std 802.11; in fact, disruption of service is likely to occur.

1. ***ML-***transition: This type is defined as described below.
2. A non-AP MLD movement from being associated with one AP MLD in one ESS to be reassociated with another AP MLD within the same ESS.
3. A non-AP MLD movement from being associated with one AP MLD in one ESS to become a non-AP STA that is reassociated with an AP within the same ESS.
4. A non-AP STA movement from being associated with one AP in one ESS to become a non-AP MLD that is reassociated with an AP MLD with the same ESS.

A fast ML transition is a ML transition that establishes the state necessary for data connectivity before the reassociation rather than after the reassociation.

The FT protocol provides a mechanism for a STA to perform a BSS transition between access points (APs) in a robust security network (RSN) or when quality-of-service (QoS) admission control is enabled in the ESS.

The over-the-air FT protocol also provides a mechanism for a non-AP MLD to perform a ML transition in a robust security network (RSN).

The different association services support the different categories of mobility.

***TGbe editor: Modify 4.5.3.3 as follows except green tag:***

* Association(#SP192, #Motion 25, #Motion 115 #SP76, #SP196)

To deliver an MSDU within an ESS via the DS, the DS(11ak)(#120) needs to know which AP or AP MLD within the ESS to deliver the MSDU, so that the MSDU might ultimately be delivered to the addressed IEEE 802.11 STA or MLD.(11ak) This information is provided to the DS by the concept of association. Association is nec-essary, but not sufficient, to support BSS/ML-transition mobility. Association is sufficient to support no-tran-sition mobility. Association is one of the services in the DSS.

Before a STA or a non-AP MLD is allowed to send an MSDU via an AP or an AP MLD, respectively, it first becomes associated with the AP or the AP MLD, respectively.For a non-GLK STA that is not affiliated with a MLD, the(11ak) act of becoming associated with an AP invokes the association service (STA association), which provides the STA to AP mapping to the DS. For a non-AP MLD, the act of becoming associated with an AP MLD invokes the association service (MLD association), which provides the non-AP MLD to AP MLD mapping to the DS. (see 33.3.2 (Multi-link (re)setup procedure)) How the information pro-vided by the association service is stored and managed within the DS is not specified by this standard.

(11ak)For a GLK STA, the act of becoming associated invokes the association service, which establishes a general link between two instances of the IEEE 802.1D Internal Sublayer Service. This link provides a point-to-point link between the two Internal Sublayer Service SAPs. The GLK AP STA and the GLK non-AP STA each coordinate with higher layer services and each other to create the point-to-point link. The higher layer services create or enable the Internal Sublayer Service SAPs, inform the GLK convergence function of the mapping of the Internal Sublayer Service SAPs, and inform the network routing protocol of the existence of the general link. The GLK AP and the GLK non-AP STA each establish a service\_access\_point\_identifier for each general link, for their respective (#4335)MS SAPs. This process allows for the establishment of a point-to-point link suitable for use in an IEEE 802.1Q network.

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 (STA association) or multiple IEEE 802.11 links (MLD association). 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 or two MLDs until an IEEE 802.1X authentication procedure completes successfully over 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. 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.

NOTE—See IEEE Std 802.1X-2010 for a discussion of Controlled Port and Uncontrolled Port.

At any given instant, a STA or a non-AP MLD is associated with no more than one AP or AP MLD, respectively. This allows the DS to determine a unique answer to the question, “Which AP or AP MLD is serving STA X or non-AP MLD X, respectively?” Once an STA or MLD association is com-pleted, a STA or a non-AP MLD, respectively, can make full use of a DS (via the AP or the AP MLD, respectively) to communicate. STA or MLD association is always initiated by the non-AP STA or the non-AP MLD, respectively, not the AP or AP MLD, respectively.

An AP or an AP MLD might be associated with many STAs or non-AP MLDs, respectively, at the same time. A STA or a non-AP MLD learns what APs or AP MLDs, respectively, are present and what operational capabilities are available from each of those APs or APs affiliated with AP MLDs, respectively, and then invokes the association service to establish a STA or a MLD, respectively, association. A FILS STA is able to discover, authenticate and associate with the AP with a(#19) reduced number of frame transmissions.(11ai) For details of how a STA learns about what APs are present, see 11.1.4 (Acquiring synchronization, scanning).

***TGbe editor: Modify 4.5.3.4 as follows:***

* Reassociation(#SP192, #Motion 115 #SP86, #SP196, #SP197)

Association is sufficient for no-transition MSDU delivery between IEEE 802.11 STAs or MLDs. Additional functionality is needed to support BSS/ML-transition mobility. The additional required functionality is provided by the reassociation service. Reassociation is one of the services in the DSS.The reassociation service is invoked to “move” a current STA association (see 4.5.3.3 (Association)) of a non-AP STA from one AP to another or a current MLD association of a non-AP MLD from one AP MLD to another or a current STA association of a non-AP STA with an AP to another MLD association of a non-AP MLD with an AP MLD or a current MLD association of a non-AP MLD with an AP MLD to another STA association of a non-AP STA with an AP (see 33.3.2 (Multi-link (re)setup procedure)). In an ESS with a DS, the reassociation service informs the DS of the current mapping between AP and STA or between AP MLD and non-AP MLD. For a general link in an IEEE 802.1Q network, the reassociation service informs higher layer services how the link is reconfigured, commonly, with which BSS the GLK non-AP STA is associated. The higher layer services will then destroy, disable, or maintain the existing Internal Sublayer Service SAPs, create or enable new Internal Sublayer Service SAPs, inform the GLK convergence function of the reconfigured general link mapping of the Internal Sublayer Service SAPs, and inform the network routing protocol of the updated general link. The GLK AP and GLK non-AP STA each then establish or maintain a service\_access\_point\_identifier for the reconfigured general link, for their respective (#4335)MS SAPs. Reassociation also enables changing association attributes of an established association while the (#2274)non-AP STA or non-AP MLD remains associated with the same AP or the same AP MLD, respectively. Reassociation is always initiated by the non-AP STA or the non-AP MLD.(11ak)

Only the fast BSS/ML transition facility can move an RSNA during reassociation. Therefore, if FT is not used, the old RSNA is deleted and a new RSNA is constructed.    

***TGbe editor: Modify 4.5.3.5 as follows:***

* Disassociation(#SP192, Motion 115 #SP88)

The disassociation service is invoked when an existing association is to be terminated. Disassociation is one of the services in the DSS.

For a non-GLK STA that is not affiliated with a MLD, the act of becoming disassociated invokes the disassociation service, which voids any existing STA to AP mapping known to the DS, for the disassociating STA. For a non-AP MLD, the act of becoming disassociated invokes the disassociation service, which voids any existing non-AP MLD to AP MLD mapping known to the DS, for the disassociating non-AP MLD. (see 33.3.3 (Multi-link tear down procedure))

How the information provided by the disassociation service is managed within the DS is not specified by this standard. For a general link, disassociation removes or disables the corresponding Internal Sublayer Service SAPs that were configured for the general link. The IEEE 802.1Q bridge uses this information to update bridging for the GLK non-AP STA.(11ak)

(11ak)For an IEEE 802.11 link that is the basis of a general link, the disassociation service informs higher layer services of the IEEE 802.1Q network that the general link has been destroyed. The GLK AP and the GLK non-AP STA each coordinate with higher layer services and each other to destroy the point-to-point link. The higher layer services destroy or disable the Internal Sublayer Service SAPs, inform the GLK convergence function of the deletion of the mapping of the Internal Sublayer Service SAPs, and inform the network routing protocol of the destruction of the general link, for their respective (#4335)MS SAPs. This process destroys the previously existing point-to-point link that was suitable for use in an IEEE 802.1Q network.

The disassociation service can be invoked by either party in an STA association (non-AP STA or AP see 4.5.3.3 (Association)) or a MLD association (non-AP MLD or AP MLD). Disassociation is a notification, not a request. Disassociation cannot be refused by the receiving STA or the receiving MLD except when management frame protection is negotiated and the message integrity check fails.

An AP or an AP MLD can disassociate STAs or non-AP MLDs, respectively, to enable the AP or AP MLD to be removed from a network for service or for other reasons. STAs or MLDs attempt to disassociate when they leave a network. However, the MAC protocol does not depend on STAs or MLDs invoking the disassociation service. (MAC management is designed to accommodate loss of communication with an associated STA or an associated MLD.)

***TGbe editor: Modify clause 9.4.1.5 as follows:***

* Current AP Address field (Motion 115 #SP87, Motion 115 #SP94)

For BSS transition, the Current AP Address field is the MAC address of the AP with which the STA is currently associated. For ML transition, if the current association is between a non-AP MLD and an AP MLD, then the Current AP Address field is the MLD MAC address of the AP MLD with which the non-AP MLD is currently associated. For ML transition, if the current association is between a non-AP STA and an AP, then the Current AP Address field is the MAC address of the AP with which the STA is currently associated. The length of the Current AP Address field is 6 octets. The Current AP Address field is shown(#243) in Figure 9-87 (Current AP Address field format(#2607)).

|  |  |
| --- | --- |
|  | Current AP Address |
| Octets: | 6 |
| * Current AP Address field format(#2607) | |

**Part 2:**

***TGbe editor: Modify clause 12.3.3 as follows:***

* Pre-RSNA authentication (Motion 115 #SP88)
* Overview

In clause 12.3.3 (Pre-RSNA authentication), the reference of a “STA” means that the “STA” is not affiliated with a MLD unless specified otherwise.

In clause 12.3.3 (Pre-RSNA authentication), when referring to MLD authentication, “SME” is the entity that manages the MLD, i.e., “MLDME”.

In an infrastructure BSS or between an AP MLD and a non-AP MLD, a non-DMG STA or the non-AP MLD, respectively, shall complete an IEEE 802.11 authentication exchange prior to association. A DMG STA not in an IBSS shall complete an IEEE 802.11 authentication exchange prior to association when an authentication algorithm other than the Open System authentication algorithm is requested. A DMG STA shall not perform an IEEE 802.11 authentication exchange using the Open System authentication algorithm. An IEEE 802.11 authentication exchange is optional in an IBSS.

All Authentication frames shall be individually addressed, as IEEE 802.11 authentication is performed between pairs of STAs or MLDs, i.e., group addressed authentication is not allowed. Deauthentication frames are advisory and may be sent as group addressed frames.

Shared Key authentication is obsolete and support for this mode might be subject to removal in a future revision of the standard. Shared Key authentication is distinct from FILS Shared Key authentication.(#4735)

* Open System authentication
* General

Open System authentication is a null authentication algorithm.

Any non-DMG STA or non-AP MLD requesting Open System authentication can be authenticated if dot11AuthenticationAlgorithmsTable at the peer STA or AP MLD, respectively, includes an entry with dot11AuthenticationAlgorithm equal to openSystem and dot11AuthenticationAlgorithmActivated equal to true.

A STA or AP MLD may decline to authenticate with another requesting STA or requesting non-AP MLD, respectively. Open System authentication is the default authentication algorithm for a pre-RSNA STA or MLD.

Open System authentication utilizes a two-message authentication transaction sequence. The first message asserts identity and requests authentication. The second message returns the authentication result. If the result is “successful,” the STAs or MLDs shall be declared mutually authenticated.

In the description in 12.3.3.2.2 (Open System authentication (first frame)) and 12.3.3.2.3 (Open System authentication (final frame)), the STA or non-AP MLD initiating the authentication exchange is referred to as the *requester*, and the STA or the AP MLD, respectively, to which the initial frame in the exchange is addressed is referred to as the *responder*.

NOTE – Open system authentication between MLDs is done before multi-link (re)setup (see 33.3.2 (Multi-link (re)setup procedure) and 11.3 (STA/MLD authentication and association)).

The specific items in each of the messages described in the following subclauses are defined in 9.3.3.11 (Authentication frame format), Table 9-42 (Authentication frame body), and Table 9-43 (Presence of fields and elements in Authentication frames).

***TGbe editor: Modify clause 12.4 as follows:***

* Authentication using a password (Motion 115 #SP88, Motion 112 #SP40)
* SAE overview

In clause 12.4 (Authentication using a password), the reference of a “STA” means that the “STA” is not affiliated with a MLD unless specified otherwise.

In clause 12.4 (Authentication using a password), when referring to MLD authentication, the reference of “SME” means the entity that manages the MLD, i.e., “MLDME”.

STAs, both AP STAs and non-AP STAs, may authenticate each other by proving possession of a password. MLDs, both AP MLDs and non-AP MLDs, may authenticate each other by proving possession of a password.

Authentication protocols that employ passwords need to be resistant to off-line dictionary attacks

Simultaneous authentication of equals (SAE) is a variant of *Dragonfly*, a password-authenticated key exchange based on a zero-knowledge proof. SAE is used by STAs or MLDs to authenticate with a password; it has the following security properties:

* The successful termination of the protocol results in a PMK shared between the two STAs or the two MLDs.
* An attacker is unable to determine either the password or the resulting PMK by passively observing an exchange or by interposing itself into the exchange by faithfully relaying messages between the two STAs or the two MLDs.
* An attacker is unable to determine either the password or the resulting shared key by modifying, forging, or replaying frames to an honest, uncorrupted STA or MLD.
* An attacker is unable to make more than one guess at the password per attack. This implies that the attacker cannot make one attack and then go offline and make repeated guesses at the password until successful. In other words, SAE is resistant to dictionary attack.
* Compromise of a PMK from a previous run of the protocol does not provide any advantage to an adversary attempting to determine the password or the shared key from any other instance.
* Compromise of the password does not provide any advantage to an adversary in attempting to determine the PMK from the previous instance.

Unlike other authentication protocols SAE does not have a notion of an “Initiator” and “Responder” or of a “Supplicant” and “Authenticator.” The parties to the exchange are equals, with each side being able to initiate the protocol. (#2222)Each side may initiate the protocol simultaneously such that each side views itself as the “initiator” for a particular run of the protocol. This is necessary to address the unique nature of MBSSs.

The parties involved are called *STA-A* and *STA-B* between two STAs or called *MLD-A and MLD-B* between two MLDs. They are identified by their MAC addresses, STA-A‑MAC and STA-B-MAC, respectively, between two STAs or by their MLD MAC addresses MLD-A‑MAC and MLD-B-MAC, respectively, between two MLDs. STAs or MLDs begin the protocol when they discover a peer by receiving Beacon or Probe Response frame(s), or when they receive an Authentication frame indicating SAE authentication from a peer.

SAE is an RSNA authentication protocol and is selected according to 12.6.2 (RSNA selection).

SAE shall be implemented on all mesh STAs to facilitate and promote interoperability.

* Representation of a password

(#1284-Ed)Passwords are used in SAE to deterministically compute a secret element in the negotiated group, called a password element. The input to this process needs to be in the form of a binary string. For the protocol to successfully terminate, it is necessary for each side to produce identical binary strings for a given password, even if that password is in character format. There is no canonical binary representation of a character and ambiguity exists when the password is a character string. To eliminate this ambiguity, a STA or a MLD shall represent a character-based password as (M180)a UTF-8 string that is processed according to the OpaqueString profile of IETF RFC 8265, the output of which is an octet string. (M180)The octet string representation of the password, after being processed, is stored in the (#1269)dot11RSNAConfigPasswordValueTable. When a “password” is called for in the description of SAE that follows the credential from the (#1269)dot11RSNAConfigPasswordValueTable is used.

(M180)Similarly, to address ambiguity when identifying passwords, a STA or a MLD shall represent a password identifier as a UTF-8 string that is processed according to the UsernameCasePreserved profile of IETF RFC 8265, the output of which is an octet string that is stored in the dot11RSNAConfigPasswordValueTable. (#2599)(#1284)When a “password identifier” is called for in the description of SAE that follows, the identifier from the dot11RSNAConfigPasswordValueTable(Ed) is used.

(#1284)In an infrastructure BSS or between an AP MLD and a non-AP MLD for which an SAE AKM is indicated, the AP or APs affiliated with the AP MLD, respectively, shall set the SAE Password Identifiers In Use subfield(Ed) of the Extended Capabilities field of the Extended Capabilities element to 1 if any entry in the dot11RSNAConfigPasswordValueTable(Ed) has a non-NULL dot11RSNAConfigPasswordIdentifier, and shall set it to 0 otherwise. Similarly, an AP or an AP affiliated with the AP MLD, respectively, shall set the SAE Password Identifiers Used Exclusively subfield(Ed) of the Extended Capabilities field of the Extended Capabilities element to 1 if every entry in the (#1269)dot11RSNAConfigPasswordValueTable(Ed) has a non-NULL dot11RSNAConfigPasswordIdentifier and shall set it to 0 otherwise.

* Finite cyclic groups
* General

SAE uses discrete logarithm cryptography to achieve authentication and key agreement. Each party to the exchange derives ephemeral public and private keys with respect to a particular set of domain parameters that define a finite cyclic group. Groups may be based (#4621)on either finite field cryptography (FFC) or on elliptic curve cryptography (ECC). Each component of a group is referred to as an *element*. Groups are negotiated using an identifying number from a repository maintained by IANA as “Group Description” attributes for IETF RFC 2409 (IKE) [B14][B28]. The repository maps an identifying number to a complete set of domain parameters for the particular group. (M104)Not all groups defined in this repository are suitable. Only FFC groups whose prime is at least 3072 bits and ECC groups defined over a prime field whose prime is at least 256 bits are suitable for use with SAE. ECC groups defined over a characteristic 2 finite field or ECC groups with a co-factor greater than 1 shall not be used with SAE (see NIST Special Publication 800-57). For the purpose of interoperability, a STA or a MLD shall (#4132)implement support for group 19, an ECC group defined over a 256-bit prime order field.

More than one group may be configured on a STA or a MLD for use with SAE by using the dot11RSNAConfigDLCGroupTable(#4540). Configured groups are prioritized in ascending order of preference. If only one group is configured, it is, by definition, the most preferred group.

(…existing texts ….)

* Hash-to-curve generation of the password element with ECC groups(M137)

An SAE peer, e.g. a mesh STA or an AP or an AP MLD, indicates support for direct hashing to obtain an ECC password element by setting the SAE hash-to-element bit in the Extended RSN Capabilities field in all Beacon and Probe Response frames. (#4726)A STA or MLD that uses a password identifier shall use the hash-to-curve method. An SAE initiator that has identified a peer that supports this technique (through receipt of Beacon or Probe Response frames) shall derive a secret element, PT, according to the following technique and indicate this by setting the status code in the SAE Commit message to SAE\_HASH\_TO\_ELEMENT. An SAE initiator shall not indicate support for this form of element derivation unless its peer has already signalled support for this method. If an SAE Commit message is received with status code equal to SAE\_HASH\_TO\_ELEMENT the peer shall generate the PWE using the following technique and reply with its own SAE Commit message with status code set(#4672) to SAE\_HASH\_TO\_ELEMENT.

(…existing texts ….)

* Direct Generation of the password element with FFC groups(M137)

An SAE peer indicates support for direct hashing to obtain the FFC password element by setting the SAE (#4726)hash-to-element bit in the Extended RSN Capabilities field in all Beacon and Probe Response frames. (#4726)A STA or a MLD that uses a password identifier shall use the direct hashing technique. An SAE initiator that has identified a peer that supports the following technique (through receipt of Beacon or Probe Response frames) shall derive PT according to the following technique and indicate this by setting the status code in the SAE Commit message to SAE\_HASH\_TO\_ELEMENT. An SAE initiator shall not indicate support for this form of PWE derivation unless its peer has already signalled support.

(…existing texts ….)

* PWE and secret generation

Prior to beginning the protocol message exchange, the secret element ***PWE*** and two secret values are generated.

(M137)When a STA or a MLD supports directly hashing to a group element (according to 12.4.4.2.3 (Hash-to-curve generation of the password element with ECC groups(M137)) or 12.4.4.3.3 (Direct Generation of the password element with FFC groups(M137))) it computes a secret element, PT, offline at provisioning time for all groups it wishes to support with that password. Prior to initiating SAE to a STA or a MLD which also supports the direct form of hashing to a group element, or upon receipt of an SAE Commit message indicating it was generated using a direct form of hashing to a group element, it shall generate the PWE by hashing the two peer MAC addresses to produce a digest, reducing the digest modulo the order of the particular group, q, interpreting the reduced digest as an integer and using it with the secret element to generate the PWE:

*val = H(0n, MAX(STA-A-MAC, STA-B-MAC) || MIN(STA-A-MAC, STA-B-MAC)) between two STAs or*

*val = H(0n, MAX(MLD-A-MAC, MLD-B-MAC) || MIN(MLD-A-MAC, MLD-B-MAC)) between two MLDs*

*val = val* (#4666)modulo *(q – 1) + 1*

*PWE = scalar-op(val, PT)*

(M137)where 0n is a salt of all zeros whose length equals the length of the digest from the hash function used to instantiate H() (see Table 12-1 (Hash algorithm based on length of prime(M137))).

(M137)If a STA does not support a direct form of hashing to a group element,(Ed) it generates the PWE after selecting a group, either the most preferred group if the STA is initiating SAE to a peer, or the group from a received SAE Commit message if the STA is responding to a peer. The ***PWE*** shall be generated for that group (according to 12.4.4.2.2 (Generation of the password element with ECC groups by looping(M137)) or 12.4.4.3.2 (Generation of the password element with FFC groups by looping(M137)), depending on whether the group is ECC or FFC, respectively) using the identities of the two STAs and the configured password.

After generation of the ***PWE***, each STA or MLD shall generate a secret value, *rand*, and a temporary secret value, *mask*, each of which shall be chosen randomly such that 1 < *rand* < *r* and 1 < *mask* < *r* and (*rand + mask*)mod *r* is greater than 1, where *r* is the (prime) order of the group. If their sum modulo r is not greater than 1, they shall both be irretrievably deleted and new values shall be randomly generated. The values *rand* and *mask* shall be random numbers produced from a quality random number drawn from a uniform distribution generator. These values shall never be reused on distinct protocol runs.

(…existing texts ….)

* Processing of a peer’s SAE Commit message

(#2599)If the peer’s SAE Commit message contains a password identifier, the value of that identifier shall be used in construction of the password element (PWE) for this exchange. If a password identifier is present in the peer’s SAE Commit message and there is no password with the given identifier a STA or a MLD shall fail authentication.(M41)

(M137)If the peer’s SAE Commit message contains a Rejected Groups element, the list of rejected groups shall be checked to ensure that all of the groups in the list are groups that would be rejected. If any groups in the list would not be rejected then processing of the SAE Commit message terminates and the STA or the MLD shall reject the peer's authentication. While the rejected groups are appended to the Rejected Groups element as they are rejected (see 12.4.7.4 (Encoding and decoding of SAE Commit messages)) there is no inherent order to the groups in the list. The order in which they are sent and received shall be retained when deriving keys.

Upon receipt of a peer’s SAE Commit message both the scalar and element shall be verified.

If the scalar value is greater than 1(M111) and less than the order, *r*, of the negotiated group, scalar validation succeeds; otherwise, it fails. Element validation depends on the type of group. For FFC groups, the element shall be an integer greater than 1 and less than the prime number *p* minus 1, (*p –*1), and the scalar operation of the element and the order of the group, *r*, shall equal 1 modulo the prime number *p*. If either of these conditions does not hold, element validation fails; otherwise, it succeeds. For ECC groups, both the x- and y-coordinates of the element shall be non-negative integers less than the prime number *p*, and the two coordinates shall produce a valid point on the curve satisfying the group’s curve definition, not being equal to the “point at the infinity.” If either of those conditions does not hold, element validation fails; otherwise, element validation succeeds.

If either scalar validation or element validation fails, the STA or the MLD shall reject the peer’s authentication. If both the scalar and element from the peer’s SAE Commit message are successfully validated, a shared secret element, *K*, shall be derived using the scalar and element (*peer-commit-scalar* and ***PEER-COMMIT-ELEMENT***, respectively) from the peer’s SAE Commit message and the STA’s or the MLD’s secret value.

***K***= scalar-op(*rand*, (elem-op(scalar-op(*peer-commit-scalar*, ***PWE***), ***PEER-COMMIT-ELEMENT***)))

If the shared secret element, ***K***, is the identity element for the negotiated group (the value one for an FFC group or the point-at-infinity for an ECC group) the STA or the MLD shall reject the peer’s authentication. Otherwise, a secret value, *k*, shall be computed as:

(…existing texts ….)

* Anti-clogging tokens

A STA or a MLD is required to do a considerable amount of work upon receipt of an SAE Commit message. This opens up the possibility of a distributed denial-of-service attack by flooding a STA or a MLD with bogus SAE Commit messages from forged MAC addresses. To prevent this from happening, a STA or a MLD shall maintain an *Open* counter in its SAE state machine indicating the number of open and unfinished protocol instances (see 12.4.5.1 (Message exchanges)). When that counter hits or exceeds dot11RSNASAEAntiCloggingThreshold, the STA or MLD shall respond to each SAE Commit message with a rejection that includes an Anti-Clogging Token field(#2534) statelessly bound to the sender of the SAE Commit message. The sender of the SAE Commit message shall then include the(Ed) Anti-Clogging Token field(#2534) in a subsequent SAE Commit message.

(…existing texts ….)

* Parent process events and output

The parent process receives events from three sources: the SME, protocol instances, and received frames.

The SME signals the following events to the parent SAE process:

* *Initiate*—An *Initiate* event is used to instantiate a protocol instance to begin SAE with a designated peer.
* *Kill*—A *Kill* event is used to remove a protocol instance with a designated peer.

Protocol instances send the following events to the SAE parent process:

* *Fail*—The peer failed to be authenticated.
* *Auth*—The peer was successfully authenticated.
* *Del*—The protocol instance has had a fatal event.

Receipt of frames containing SAE messages signals the following events to the SAE parent process:

* *Authentication frame with Transaction Sequence number 1*—This event indicates that an SAE Commit message has been received from a peer STA or a peer MLD.
* *Authentication frame with Transaction Sequence number 2*—This event indicates that an SAE Confirm message has been received from a peer STA or a peer MLD.

The parent process generates Authentication frames with Authentication transaction sequence 1 and a Status of 76 indicating rejection of an Authentication attempt because an Anti-Clogging Token field(#2534) is required.

***TGbe editor: Modify 11.3 as follows:***

* STA/MLD authentication and association(#SP192, #Motion 25, #Motion 115 #SP88, #Motion 115 #SP86, #Motion 115 #SP76, #SP196, #SP197)
* State variables

In caluse 11.3 (STA/MLD authentication and association), the reference of a “STA” means that the “STA” is not affiliated with a MLD unless specificed otherwise.

In clause 11.3 (STA/MLD authentication and association), when referring to MLD authentication, MLD disauthentication ,MLD (re)association, MLD disassociation, or MLD 4-way handshake, the reference of “SME” means the entity that manages the MLD, i.e., “MLDME”,

A STA (local) for which dot11OCBActivated is false keeps an enumerated state variable for each STA (remote) with which direct communication via the WM is needed. In this context, direct communication refers to the transmission of any Class 2 or Class 3 frame with an Address 1 field that matches the MAC address of the remote STA.

A MLD (local) keeps an enumerated state variable for each MLD (remote) with which direct communication via the WM is needed. In this context, direct communication refers to the transmission of any Class 2 or Class 3 frame with an Address 1 field that matches the MAC address of the STA affiliated with the remote MLD and an Address 2 field that matches the MAC address of the STA affiliated with the local MLD.A STA for which dot11MeshActivated is true (i.e., a mesh STA) does not use procedures described in 11.3.5 (Association, reassociation, and disassociation). Instead, a mesh STA uses a mesh peering management protocol (MPM) or a authenticated mesh peering exchange (AMPE) to manage states and state variables for each peer STA. See 14.3 (Mesh peering management (MPM)) and 14.5 (Authenticated mesh peering exchange (AMPE)) for details.

A STA for which dot11OCBActivated is true does not use MAC sublayer authentication or association and does not keep this state variable.

For nonmesh STAs, this state variable expresses the relationship between the local STA and the remote STA. It takes on the following values:

* *State 1*: Initial start state for non-DMG STAs and for DMG STAs that perform IEEE 802.11 authentication. Unauthenticated and unassociated.
* *State 2*: Initial start state for DMG STAs that do not perform IEEE 802.11 authentication. Authenticated (except DMG STAs that do not perform IEEE 802.11 authentication, which are unauthenticated) but unassociated.
* *State 3*: Authenticated (except DMG STAs that did not perform IEEE 802.11 authentication, which are unauthenticated) and associated (Pending RSNA Authentication). The IEEE 802.1X Controlled Port is blocked.
* *State 4*: Authenticated (except DMG STAs that did not perform IEEE 802.11 authentication, which are unauthenticated) and associated (RSNA Established or Not Required). The IEEE 802.1X Controlled Port is unblocked, or not present.

For MLDs, this state variable expresses the relationship between the local MLD and the remote MLD. It takes on the following values:

* *State 1*: Initial start state for MLDs that perform IEEE 802.11 authentication. Unauthenticated and unassociated.
* *State 2*: Authenticated but unassociated.
* *State 3*: Authenticated and associated (Pending RSNA Authentication). The IEEE 802.1X Controlled Port is blocked.
* *State 4*: Authenticated and associated (RSNA Established or Not Required). The IEEE 802.1X Controlled Port is unblocked, or not present.

The state variable is kept within the MLME (i.e., is written and read by the MLME). The SME may also read this variable using the (#2369)MLME-GETAUTHASSOCSTATE.request primitive.

The SME may also read this variable using the (#2369)MLME-GETAUTHASSOCSTATE.request primitive.

Mesh STAs manage the state variable as described in 14.3.2 (State variable management).

* State transition diagram for nonmesh STAs or MLDs

Figure 11-17 (Relationship between state and services between a given pair of nonmesh STAs(#1554)(11ai)) shows the state transition diagram for nonmesh STA states or MLD states. Note that only events causing state changes are shown. The state of the sending STA or MLD given by Figure 11-17 (Relationship between state and services between a given pair of nonmesh STAs(#1554)(11ai)) is with respect to the intended receiving STA or receiving MLD, respectively.

NOTE—A transition to State 1 might occur for other reasons such as no frames having been received from a STA or a MLD for a period of time.



Figure 11-17 (Relationship between state and services between a given pair of nonmesh STAs or MLDs(#1554)(11ai))



* Frame filtering based on STA or MLD state

The current state existing between the transmitter and receiver STAs or MLDs determines the IEEE 802.11 frame types that may be exchanged between that pair of STAs or MLDs (see Clause 9 (Frame formats)). A unique state exists for each pair of transmitter and receiver STAs or MLDs. The allowed frame types are grouped into classes and the classes correspond to the STA state or the MLD state. In State 1, only Class 1 frames are allowed. In State 2, only(11ai) Class 1 or Class 2 frames are allowed. In State 3 and State 4, all frames are allowed (Classes 1, 2, and 3). In the definition of frame classes, the following terms are used:

* Within an infrastructure BSS: both the transmitting STA and the recipient STA participate in the same infrastructure BSSWithin a PBSS: both the transmitting STA and the recipient STA participate in the same PBSS
* Within an IBSS: both the transmitting STA and the recipient STA participate in the same IBSS
* dot11RSNAActivated: reference to the setting of dot11RSNAActivated at the STA or MLD that needs to determine whether a transmission or reception is permitted.

NOTE—The phrase “within a BSS” comprises “within a PBSS,” “within an IBSS,” “within an MBSS(#1165),” or “within an infrastructure BSS.”

STA A participates in the same infrastructure BSS as STA B if at least one of the following conditions is met:

* STA A is associated with STA B, and either STA A or STA B is an AP.
* STA A receives a frame with the value of its TA field equal to the MAC address of STA B and with the value of its BSSID field equal to the BSSID of the BSS with which STA A is associated.
* STA A receives an Information Response frame from the AP with which it is associated containing an explicit indication that STA B is a member of the BSS with which STA A is associated.

STA A participates in the same PBSS as STA B if at least one of the following conditions is met:

* STA A is associated with STA B, and either STA A or STA B is a PCP.
* STA A receives a frame with the value of its TA field equal to the MAC address of STA B and with the value of its BSSID field equal to the BSSID of the PBSS that STA A has joined or started.
* STA A receives a frame, i.e., an Information Response frame, from its PCP containing an explicit indication that STA B is a member of the PBSS that STA A has joined.

STA A participates in the same IBSS as STA B if STA A receives a frame with the value of its TA field equal to the MAC address of STA B and with the value of its BSSID field equal to the BSSID of the IBSS that STA A has joined or started.

The frame classes are defined as follows:

* Class 1 frames
* Control frames
* RTS
* CTS
* DMG Clear to send (DMG CTS)
* Ack
* Grant
* SSW
* SSW-Feedback
* SSW-Ack
* Grant Ack(#2699)
* CF-End
* In an IBSS and in a PBSS when dot11RSNAActivated is false, Block Ack (BlockAck)
* In an IBSS and in a PBSS when dot11RSNAActivated is false, Block Ack Request (BlockAckReq)
* Management frames
* Probe Request/Response
* Beacon
* Authentication
* Deauthentication
* ATIM
* Public Action
* Self-protected Action
* In an IBSS, all Action frames and all Action No Ack frames
* Unprotected DMG Action frames(#4816)
* In a PBSS when dot11RSNAActivated is false, all Action and Action No Ack frames except the following frames:
* ADDTS Request
* ADDTS Response
* DELTS
* Data frames
* Data frames between IBSS STAs(#59)
* Data frames within a PBSS
* (Ed)Extension frames
* DMG Beacon
* Class 2 frames
* Management frames
* Association Request/Response
* Reassociation Request/Response
* Disassociation
* Class 3 frames
* Data frames
* Data frames between STAs in an infrastructure BSS or in an MBSS

ii) Data frames between an AP MLD and a non-AP MLD associated with the AP MLD

NOTE – Data frames transmissions for a certain TID on a link between an AP MLD and a non-AP MLD associated with the AP MLD is subject to additonal constrants (see 33.3.4 (Link management)).

* Management frames
* In an infrastructure BSS, an MBSS, or a PBSS, all Action and Action No Ack frames except those that are declared to be Class 1 or Class 2 frames

ii) Between an AP MLD and a non-AP MLD associated with the AP MLD, all Action and Action No Ack frames except those that are declared to be Class 1 or Class 2 frames

NOTE – Management frames transmissions on a link between an AP MLD and a non-AP MLD associated with the AP MLD is subject to additional constraints (see 33.3.4 (Link management)).

* Control frames
* PS-Poll
* Poll
* SPR
* DMG DTS
* Block Ack (BlockAck), except those that are declared to be Class 1
* Block Ack Request (BlockAckReq), except those that are declared to be Class 1 (above)

Class 2 and Class 3 frames are not allowed in an IBSS. If an IBSS STA receives a Class 2 or Class 3 frame, it shall ignore the frame.

A STA or a STA affiliated to a MLD shall not transmit Class 2 frames unless in State 2 or State 3 or State 4.

A STA or a STA affiliated to a MLD shall not transmit Class 3 frames unless in State 3 or State 4.

A multi-band capable device that uses OCT to move from State 2 to either State 3 or State 4 shall not transmit frames before the transmitting STA becomes over-the-WM enabled (see 11.32.5 (On-channel Tunneling (OCT) operation)).

The use of the word “receive” in 11.3 (STA authentication and association) refers to a frame that meets all of the filtering criteria specified in Clause 12 (Security) and Clause 10 (MAC sublayer functional description(#107)).

* Authentication and deauthentication
* General

This subclause describes the procedures used for IEEE 802.11 authentication and deauthentication. The states used in this description are defined in 11.3.1 (State variables).

Successful authentication sets (#4304)the state for a STA or a MLD to State 2, if it was in State 1. Unsuccessful authentication leaves the (#4304)state for the STA or the MLD unchanged.

Deauthentication notification sets the (#4304)state for a STA or a MLD to State 1. Deauthentication notification when in State 3 or 4 implies disassociation as well. A STA or a MLD may deauthenticate a peer STA or a peer MLD, respectively, at any time, for any reason.

If STA A in an infrastructure BSS receives a Class 2 or Class 3 frame from STA B that is not authenticated with STA A (i.e., the state for STA B is State 1), STA A shall discard the frame. If the frame has an individual address in the Address 1 field, the MLME of STA A shall send a Deauthentication frame to STA B.

If an AP affiliated with the AP MLD receives a Class 2 or Class 3 frame from a non-AP STA affiliated with the non-AP MLD that is not authenticated with the AP MLD (i.e., the state for the non-AP MLD is State 1), the frame shall be discarded. If the frame has an individual address in the Address 1 field, the AP affiliated with the AP MLD shall send a Deauthentication frame to the non-AP STA affiliated with the non-AP MLD.

If an non-AP STA affiliated with the non-AP MLD receives a Class 2 or Class 3 frame from an AP affiliated with the AP MLD that is not authenticated with the non-AP MLD (i.e., the state for the AP MLD is State 1), the frame shall be discarded. If the frame has an individual address in the Address 1 field, the non-AP STA affiliated with the non-AP MLD shall send a Deauthentication frame to the AP affiliated with the AP MLD.

Authentication is optional in an IBSS. In a non-DMG infrastructure BSS, authentication is required. Between an AP MLD and a non-AP MLD, authentication is required. In a DMG infrastructure BSS and PBSS, the Open System authentication algorithm is not used (see 12.3.3.1 (Overview)).(#2582) APs, AP MLDs and PCPs do not initiate authentication.

* Authentication—originating STA or MLD

Upon receipt of an MLME-AUTHENTICATE.request primitive that is part of an on-channel tunneling (see 11.32.5 (On-channel Tunneling (OCT) operation)), the originating STA shall follow the rules in 11.32.5 (On-channel Tunneling (OCT) operation) in addition to the authentication procedure described below.

Upon receipt of an MLME-AUTHENTICATE.request primitive, the originating STA or MLD shall authenticate with the indicated STA or MLD, respectively, using the following procedure:

* If the STA is in an IBSS, the SME shall delete any PTKSA, GTKSA, IGTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)).
* The STA or the MLD shall execute one of the following:
* For the Open System or Shared Key authentication algorithm, the authentication mechanism described in 12.3.3.2 (Open System authentication) or 12.3.3.3 (Shared Key authentication), respectively.
* For the fast BSS/ML transition (FT) authentication algorithm in an ESS, the authentication mechanism described in 13.5 (FT protocol), or, if resource requests are included, 13.6 (FT resource request protocol).
* For SAE authentication between an AP MLD and a non-AP MLD or in an infrastructure BSS, IBSS, or MBSS, the authentication mechanism described in 12.4 (Authentication using a password).
* For FILS authentication, the authentication mechanism described in 12.11 (Authentication for FILS(11ai)). (#2582)An AP or PCP may provide estimated association response latency to a non-AP and non-PCP STA using the Association Delay Info field in the Association Delay Info element (9.4.2.174 (Future Channel Guidance element)). The value of the Association Delay Info field shall be larger than dot11HLPWaitTime(11ai).
* If the authentication was successful within the AuthenticateFailureTimeout, the state for the indicated STA or MLD shall be set to State 2 if it was State 1; the state shall remain unchanged if it was other than State 1.
* The MLME shall issue an MLME-AUTHENTICATE.confirm primitive to inform the SME of the result of the authentication.
* Authentication—destination STA or MLD

Upon receipt of an Authentication frame with authentication transaction sequence number equal to 1, the destination STA or MLD shall authenticate with the originating STA or MLD, respectively, using the following procedure:

* If Open System or Shared Key authentication algorithm is being used, the STA or the MLD shall execute the procedure described in 12.3.3.2 (Open System authentication) or 12.3.3.3 (Shared Key authentication), respectively. These result in the generation of an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request.
* If FT authentication is being used, the MLME shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request, including the FT Authentication Elements, and the SME shall execute the procedure as described in 13.5 (FT protocol) or 13.6 (FT resource request protocol).
* If SAE authentication is being used between an AP MLD and a non-AP MLD or in an infrastructure BSS, IBSS, or MBSS, the MLME shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request, including the SAE authentication elements, and the SME shall execute the procedure as described in 12.4 (Authentication using a password)(Ed).
* If FILS authentication is being used, the MLME shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request, and the SME shall execute the procedure described in 12.11 (Authentication for FILS(11ai))(11ai).
* If the STA is in an IBSS and management frame protection was not negotiated when the PTKSA(s) were created, the SME shall delete any PTKSA, GTKSA, IGTKSA and temporal keys held for communication with the originating STA by using the MLME-DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)).
* 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 or MLD 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).
* 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), with a status code of SUCCESS, and the state for the originating STA or MLD shall be set to State 2 if it was in State 1(#1403); the state shall remain unchanged if it was other than State 1.

(#1402)NOTE—If management frame protection was negotiated, the SME does not change the state for the originating STA or originating MLD and does not delete any of the previously created SAs or temporal keys as a part of this authentication procedure.

If the STA is in an IBSS, if the SME decides to initiate an RSNA, and if the SME does not know the security policy of the peer, it may issue an individually addressed Probe Request frame to the peer by invoking an MLME-SCAN.request primitive to discover the peer’s security policy.

(#2582)When a non-AP and non-PCP STA receives an Authentication frame that includes an Association Delay Info element, it sets (#2317)dot11AssociationResponseTimeOut(#2318) equal to or larger than the Association Delay Info field(11ai).

* Deauthentication—originating STA or MLD

The originating STA or MLD shall deauthenticate with the indicated STA or MLD, respectively, using the following procedure:

* The SME shall generate an MLME-DEAUTHENTICATE.request primitive containing the appropriate reason code for the STA or MLD deauthentication, as defined in Table 9-51 (Reason codes) of 9.4.1.7 (Reason Code field).
* On receipt of the MLME-DEAUTHENTICATE.request primitive, if the state for the indicated STA or MLD is State 2, State 3, or State 4, the MLME shall generate a Deauthentication frame to be transmitted to the indicated STA or MLD, respectively.

NOTE—As the Deauthentication frame is a bufferable MMPDU, the transmission of this frame might be delayed by the operation of a power saving(M101) protocol. The AID and the PTKSA are maintained (when applicable) until the frame is acknowledged or attempts to transmit the frame are abandoned.

* The state for the indicated STA or MLD shall be set to State 1.
* Once the Deauthentication frame is acknowledged or attempts to transmit the frame are abandoned, the MLME shall issue an MLME-DEAUTHENTICATE.confirm primitive to inform the SME of the deauthentication.
* The SME, upon receipt of an MLME-DEAUTHENTICATE.confirm primitive, shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA(#2116) and temporal keys held for communication with the indicated STA or MLD by using the MLME-DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)) and by generating an MLME-SETPROTECTION.request(None) primitive.
* If the STA is contained within an AP or PCP(#2582), its SME, upon receipt of an MLME-DEAUTHENTICATE.confirm primitive, shall release the AID assigned for the indicated STA, if the state for the indicated STA was State 3 or State 4.

fa) If the MLD is an AP MLD, its MLDME, upon receipt of an MLME-DEAUTHENTICATE.confirm primitive, shall release the AID assigned for the indicated non-AP MLD, if the state for the indicated MLD was State 3 or State 4.

* If the STA is contained within an AP, its SME shall inform the DS of the disassociation, if the state for the indicated STA was State 3 or State 4.

ga) If the MLD is an AP MLD, its MLDME shall inform the DS of the disassociation, if the state for the indicated non-AP MLD was State 3 or State 4.

* If the STA is a mesh STA, its SME shall inform the mesh peering instance controller (see 14.3.4 (Mesh peering instance controller)) of the deauthentication.
* Deauthentication—destination STA or MLD

A DMG STA in State 2, State 3 or State 4 that receives a Deauthentication frame shall remain in the same state if it did not perform an IEEE 802.11 authentication exchange.

Otherwise, upon receipt of a Deauthentication frame from a STA or a MLD for which the state is State 2, State 3, or State 4, the destination STA or MLD, respectively, shall deauthenticate with the originating STA or MLD, respectively, using the following procedure:

* If management frame protection was not negotiated when the PTKSA(s) were created, or if management frame protection is in use and the frame is not discarded per management frame protection processing, the MLME shall issue an MLME‑DEAUTHENTICATE.indication primitive to inform the SME of the deauthentication, and set the state for the originating STA or the originating MLD to State 1.
* Upon receiving an MLME-DEAUTHENTICATE.indication primitive, the SME shall
* Delete any PTKSA, GTKSA, IGTKSA, BIGTKSA(#2116) and temporal keys held for communication with the originating STA or the originating MLD by using the MLME-DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)) and by generating an MLME-SETPROTECTION.request(None) primitive.
* If the STA is contained within an AP or PCP(#2582), release the AID assigned for the indicated STA.

2a) If the MLD is an AP MLD, release the AID assigned for the indicated non-AP MLD.

* (#2582)If the STA is contained within an AP, inform the DS of the disassociation, if the state for the originating STA was State 3 or State 4.

3a) If the MLD is an AP MLD, inform the DS of the disassociation, if the state for the originating non-AP MLD was State 3 or State 4.

* (#2582)If the STA is a mesh STA, inform the mesh peering instance controller (see 14.3.4 (Mesh peering instance controller)) of the deauthentication.

**Part 3:**

* Association, reassociation, and disassociation
* General

Subclause 11.3.5 (Association, reassociation, and disassociation) describes the procedures used for IEEE 802.11 association, reassociation and disassociation.

The states used in this description are defined in 11.3.1 (State variables).

Successful association enables a STA or a MLD to exchange Class 3 frames. (#2223)Successful association sets the state for a (11ai)non-FILS STA or a non-FILS MLD to State 3 or State 4. Successful association sets the state for FILS STAs to State 4(11ai).

Successful reassociation enables a STA or a MLD to exchange Class 3 frames. Unsuccessful reassociation when not in State 1 leaves the (#4304)state for a STA state unchanged (with respect to the AP or PCP that was sent the Reassociation Request (which may be the current STA)) or for a non-AP MLD state unchanged (with respect to the AP MLD that was sent the Reassociation Request). Successful reassociation sets the (#4304)state for a non-FILS(11ai) STA to State 3 or State 4 (with respect to the AP or PCP that was sent the Reassociation Request frame(11ai)) or for a non-FILS non-AP MLD to State 3 or State 4 (with respect to the AP MLD that was sent the Reassociation Request frame). Successful reassociation when not in State 1 sets the (#4304)state for a STA to State 2 (with respect to the current AP or PCP, if this is not the AP or PCP that was sent the Reassociation Request frame(11ai)) or for a non-AP MLD to State 2 (with respect to the current AP MLD, if this is not the AP MLD that was sent the Reassociation Request frame). Successful reassociation sets the (#4304)state for a FILS STA to State 4 (with respect to the AP or PCP that was sent the Reassociation Request frame) and enables it to exchange Class 3 frames(11ai). Reassociation shall be performed only if the originating STA or MLD is already associated in the same ESS.

Disassociation notification when not in State 1 sets (#4304)the state for a non-FILS(11ai) STA or a non-FILS MLD to State 2. Disassociation notification when not in State 1 sets (#4304)the state for a FILS STA to State 1(11ai). The STA or MLD shall become associated again prior to sending Class 3 frames. A STA or a MLD may disassociate a peer STA or a peer MLD, respectively, at any time, for any reason.

If non-DMG STA A in an infrastructure BSS receives a Class 3 frame from STA B that is authenticated but not associated with STA A (i.e., the state for STA B is State 2), STA A shall discard the frame. If the frame has an individual address in the Address 1 field, the MLME of STA A shall send a Disassociation frame to STA B.

If an AP affiliated with an AP MLD receives a Class 3 frame from a non-AP STA affiliated with a non-AP MLD that is authenticated but not associated with the AP MLD (i.e., the state for the non-AP MLD is State 2), the frame shall be discarded. If the frame has an individual address in the Address 1 field, the AP affiliated with the AP MLD shall send a Disassociation frame to the non-AP STA affiliated with the non-AP MLD.If an non-AP STA affiliated with a non-AP MLD receives a Class 3 frame from an AP affiliated with an AP MLD that is authenticated but not associated with the non-AP MLD (i.e., the state for the AP MLD is State 2), the frame shall be discarded. If the frame has an individual address in the Address 1 field, the non-AP STA affiliated with the non-AP MLD shall send a Disassociation frame to the AP affiliated with the AP MLD.

If DMG STA A in an infrastructure BSS receives a Class 3 frame from STA B that is not associated with STA A (i.e., the state for STA B is State 2), STA A shall discard the frame. If the frame has an individual address in the Address 1 field, the MLME of STA A shall send a Disassociation frame to STA B.

If an MM-SME coordinated STA receives an Association Response frame with a result code equal to SUCCESS and with the (M101)Single AID field within MMS element equal to 1, then

* For each of its MAC entities advertised within the MMS element and for which dot11RSNAActivated is true, the state is set to State 3. Progress from State 3 to State 4 occurs independently in each such MAC entity.
* For each of its MAC entities advertised within the MMS element and for which dot11RSNAActivated is false, the state is set to State 4.

If the MM-SME coordinated STA in State 3 is assigned an AID for only the MAC entity identified by the RA field of the Association Response frame(11ai) with result code equal to SUCCESS, the MM-SME may repeat the association procedure for any other MAC entity coordinated by the MM-SME.

Association is not applicable in an IBSS. In an infrastructure BSS, association is required. Between an AP MLD and a non-AP MLD, association is required. In a PBSS, association is optional. (#2582)APs, AP MLDs, and PCPs do not initiate association.

* Non-AP STA, non-AP MLD, and non-PCP STA association initiation procedures

The SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA(#2116) and temporal keys held for communication with the AP or PCP by using MLME-DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)) before invoking MLME-ASSOCIATE.request primitive.

The MLDME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA(#2116) and temporal keys held for communication with the AP MLD by using MLME-DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)) before invoking MLME-ASSOCIATE.request primitive.

If dot11InterworkingServiceActivated is true, the STA is associating with an AP but(#2582) does not have credentials for the AP, and the STA is initiating an emergency services association procedure, the SME shall submit the MLME-ASSOCIATE.request primitive with EmergencyServices parameter set to true.

The MM-SME of a non-AP and non-PCP STA may include an MMS element in an MLME-ASSOCIATE.request primitive. The MM-SME shall include in the MMS element the MAC address associated with the MLME SAP instance to which the primitive is submitted.

Upon receipt of an MLME-ASSOCIATE.request primitive that is part of an on-channel tunneling (see 11.32.5 (On-channel Tunneling (OCT) operation)), a non-AP and non-PCP STA shall follow the rules in 11.32.5 (On-channel Tunneling (OCT) operation) in addition to the association procedures described below.

Upon receipt of an MLME-ASSOCIATE.request primitive, a non-AP, non-AP MLD, and non-PCP STA shall associate with an AP, AP MLD, or PCP, respectively, using the following procedure:

* If the state for the AP, AP MLD, or PCP(#2582) is State 1, the MLME shall inform the SME of the failure of the association by issuing an MLME-ASSOCIATE.confirm primitive, and this procedure ends.
* (#1454)All the states, agreements and allocations listed in both numbered lists in 11.3.5.4 item c) are deleted or reset to initial values.
* The MLME shall transmit an Association Request frame to the AP, AP MLD, or PCP. (#2201)The RSNE contained in the MLME-ASSOCIATE.request primitive shall be included in the Association Request frame. The RSNE shall specify exactly one pairwise cipher suite and exactly one AKM suite. If the MLME-ASSOCIATE.request primitive contained the EmergencyServices parameter equal to true, an Interworking element with the UESA field set to 1 shall be included in the Association Request frame.
* If an Association Response frame is received with a status code of SUCCESS, a DMG STA shall write to each of the following MIB attributes (M101)the corresponding subfield of the DMG BSS Parameter Configuration field of the DMG Operation element received from the AP or PCP to which it requested association:
* dot11PSRequestSuspensionInterval from the PSRequestSuspensionInterval subfield
* dot11MinBHIDuration from the MinBHIDuration subfield
* dot11BroadcastSTAInfoDuration from the BroadcastSTAInfoDuration subfield
* dot11AssocRespConfirmTime from the AssocRespConfirmTime subfield
* dot11MinPPDuration from the MinPPDuration subfield
* dot11SPIdleTimeout from the SPIdleTimeout subfield
* dot11MaxLostBeacons from the MaxLostBeacons subfield
* If an Association Response frame is received with a status code of SUCCESS, the state for the AP, AP MLD, or PCP shall be set to State 4 or, if dot11RSNAActivated is true, State 3. The state for any other AP, AP MLD, or PCP which is State 3 or State 4 prior to the association request shall be set to State 2, and the MLME shall issue an MLME-ASSOCIATE.confirm primitive to inform the SME of the successful completion of the association.
* (#181)If an Association Response frame is received with a status code of SUCCESS at an (Ed)MM-SME coordinated STA and (M101)the Single AID field within the MMS element is equal to 1, then
* For each of its MAC entities advertised within the MMS element and for which dot11RSNAActivated is true, the state is set to State 3. Progress from State 3 to State 4 occurs independently in each such MAC entity.
* For each of its MAC entities advertised within the MMS element and for which dot11RSNAActivated is false, the state is set to State 4.
* For each of its MAC entities advertised within the MMS element the state for any other AP or PCP which is State 3 or State 4 prior to the association request shall be set to State 2.
* If an Association Response frame is received with a status code other than SUCCESS or the association fails to complete within dot11AssociationResponseTimeout, the state for the AP, AP MLD, or PCP shall be set to State 2, and the MLME shall issue an MLME-ASSOCIATE.confirm primitive to inform the SME of the failure of the association. The status code returned in the Association Response frame indicates the cause of the failed association attempt. Any misconfiguration or parameter mismatch, e.g., data rates required as basic rates that the STA did not indicate as supported in the STA’s Supported Rates and BSS Membership Selectors element, shall be corrected before the SME issues an MLME-ASSOCIATE.request primitive for the same AP, AP MLD, or PCP. If the status code indicates the association failed because of a reason that is not related to configuration (e.g., the AP or PCP is unable to support additional associations) and the Association Response frame does not include a Timeout Interval element with Timeout Interval Type equal to 3 the SME shall not issue an MLME-ASSOCIATE.request primitive for the same AP, AP MLD, or PCP, until a period of at least 2 s has elapsed. If the status code indicates the association failed and the Association Response frame contains a Timeout Interval element with Timeout Interval Type equal to 3, the SME shall not issue an MLME-ASSOCIATE.request primitive for the same AP, AP MLD, or PCP until the period specified in the Timeout Interval element has elapsed.
* If an MLME-ASSOCIATE.confirm primitive is received with a ResultCode of SUCCESS, and RSNA is required, and FILS authentication was not used(11ai), then the SME shall perform a 4-way handshake to establish an RSNA with the STA or the AP MLD. As a part of a successful 4-way handshake, the SME (11ai)shall enable protection by generating an MLME-SETPROTECTION.request(Rx\_Tx) primitive. If an MLME-ASSOCIATE.confirm primitive is received with a ResultCode of SUCCESS, and FILS authentication was used, then the SME shall enable protection by generating an MLME-SETPROTECTION.request(Rx\_Tx) primitive(11ai).
* Upon receipt of the MLME-SETPROTECTION.request(Rx\_Tx) primitive, the MLME shall set the state of the STA or the AP MLD to State 4.
* AP, AP MLD, or PCP association receipt procedures

Upon receipt of an Association Request frame from a STA, a non-AP MLD, or a STA, the AP, AP MLD, or PCP, respectively, shall use the following procedure:

* The MLME shall issue an MLME-ASSOCIATE.indication primitive to inform the SME of the association request. The SME shall issue an MLME-ASSOCIATE.response primitive addressed to the STA or MLD identified by the PeerSTAAddress parameter of the MLME-ASSOCIATE.indication primitive. If the association is not successful, the SME shall indicate a specific reason for the failure to associate in the ResultCode parameter. Upon receipt of the MLME-ASSOCIATE.response primitive, the MLME shall transmit an Association Response frame.
* If the state for the STA is 1 and the STA is a non-DMG STA or the state of the non-AP MLD is 1, the SME shall refuse the association request by issuing an MLME-ASSOCIATE.response primitive with ResultCode NOT\_AUTHENTICATED.
* AP with dot11InterworkingServiceActivated true only: If the MLME-ASSOCIATE.indication primitive has the EmergencyServices parameter set to true and the RSN parameter does not include an RSNE, the SME shall not reject the association request on the basis that dot11RSNAActivated is true, thereby granting access, using unprotected frames (see 9.2.4.1.9 (Protected Frame subfield)), to the network for emergency services purposes.
* Otherwise, in an RSNA the SME shall check the values received in the RSN parameter to see whether the values received match the security policy. If they do not, the SME shall refuse the association by issuing an MLME-ASSOCIATE.response primitive with a ResultCode indicating the security policy mismatch.
* Otherwise, if the state for the STA is 4, the STA has a valid security association, the STA has negotiated management frame protection, and there has been no earlier, timed out SA Query procedure with the STA (which would have allowed a new association process to be started, without an additional SA Query procedure):
* The SME shall refuse the association request by issuing an MLME-ASSOCIATE.response primitive with ResultCode REFUSED\_TEMPORARILY and TimeoutInterval containing a Timeout Interval element with the Timeout Interval Type field set to 3 (Association Comeback time). If the SME is in an ongoing SA Query with the STA, the Timeout Interval Value field shall be set to the remaining SA Query period, otherwise it shall be set to dot11AssociationSAQueryMaximumTimeout.
* The state for the STA shall be left unchanged.
* Following this, if the SME is not in an ongoing SA Query with the STA, the SME shall issue one MLME-SA-QUERY.request primitive addressed to the STA every dot11AssociationSAQueryRetryTimeout TUs until an MLME-SA-QUERY.confirm primitive for the STA is received or dot11AssociationSAQueryMaximumTimeout TUs from the beginning of the SA Query procedure have passed. The SME shall increment the TransactionIdentifier by 1 for each MLME-SA-QUERY.request primitive, rolling it over the value to 0 after the maximum allowed value is reached.
* If no MLME-SA-QUERY.confirm primitive for the STA is received within the dot11AssociationSAQueryMaximumTimeout period, the SME shall allow a subsequent association process with the STA to be started without starting an additional SA Query procedure, except that the SME may deny a subsequent association process with the STA if an MSDU was received from the STA within this period.

NOTE—Reception of an MSDU implies reception of a valid protected frame, which obviates the need for the SA Query procedure.

* The SME shall refuse an association request from a STA or non-AP MLD that does not support all of the rates in the BSSBasicRateSet parameter (11ak)and all of the membership selectors in the BSSMembershipSelectorSet parameter of the AP or the corresponding AP in each setup link, respectively, in the MLME-START.request primitive.
* The SME shall refuse an association request from an HT STA or non-AP MLD that does not support all of the MCSs in the Basic HT-MCS Set field of the HT Operation parameter of the AP or the corresponding AP in each setup link, respectively, in the MLME-START.request primitive.
* The SME shall refuse an association request from a VHT STA or non-AP MLD that does not support all of the <VHT-MCS, NSS> tuples indicated by the Basic VHT-MCS And NSS Set field of the VHT Operation parameter of the AP or the corresponding AP in each setup link, respectively, in the MLME-START.request primitive.
* (11ak)An AP or PCP may refuse GLK association based on local policy and, if so, shall return the GLK\_NOT\_AUTHORIZED ResultCode.

(11ak)NOTE—For example, there might be a list of authorized GLK peers or clients or a limit on the number of GLK peers or clients and the peer or client is not on that list or its acceptance would exceed the limit.

* The SME shall generate an MLME-ASSOCIATE.response primitive with the PeerSTAAddress parameter set to the MAC address of the STA identified by the PeerSTAAddress parameter of the MLME-ASSOCIATE.indication primitive. If the ResultCode in the MLME-ASSOCIATE.response primitive is SUCCESS, the SME has an existing SA with the STA, and an SA Query procedure with that STA has failed to receive a valid response (i.e., has not received an MLME-SA-QUERY.confirm primitive within the dot11AssociationSAQueryMaximumTimeout period), the SME shall issue an MLME-DISASSOCIATE.request primitive addressed to the STA with ReasonCode INVALID\_AUTHENTICATION.

NOTE—This MLME-DISASSOCIATE.request primitive generates a protected Disassociation frame. If the association request was genuine, the STA has deleted the PTKSA by this point and so the protected Disassociation frame is ignored. The purpose is to inform a STA which has for some reason failed to respond to an SA Query procedure triggered by a forged association request.

* (#1454)If the ResultCode in the MLME-ASSOCIATE.response primitive is SUCCESS, all the states, agreements and allocations pertaining to the associating STA or the associating non-AP MLD and listed in both numbered lists in 11.3.5.4 item c) are deleted or reset to initial values.
* If the ResultCode in the MLME-ASSOCIATE.response primitive is SUCCESS, the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA(#2116) and temporal keys held for communication with the STA or non-AP MLD by using the MLME-DELETEKEYS.request primitive (see 11.5.18 (RSNA security association termination)).
* If the MLME-ASSOCIATE.indication primitive includes an MMS parameter, the AP or PCP shall generate the MLME-ASSOCIATE.response primitive directed to the MLME of the STA identified by the PeerSTAAddress parameter of the MLME-ASSOCIATE.request primitive and take the following additional action, as appropriate:
* If the Single AID field in the MMS parameter of the MLME-ASSOCIATE.indication primitive is equal to 1, the AP or PCP may allocate a single AID for all of the STAs included in the MMS element. If the AP or PCP allocates the same AID to each STA whose MAC address was included in the MMS element, it shall include the MMS element received from the MM-SME coordinated STA in the (#4804)MLME-ASSOCIATE.response primitive.
* If the Single AID field is 0, the AP or PCP shall allocate a distinct AID for each STA specified in the MMS element.
* If an Association Response frame with a status code of SUCCESS is acknowledged by the STA or the non-AP MLD, the state for the STA or the non-AP MLD shall be set to State 4 or, if dot11RSNAActivated is true, State 3.
* If the ResultCode in the MLME-ASSOCIATE.response primitive is not SUCCESS and management frame protection is in use the state for the STA or the non-AP MLD shall be left unchanged. If the ResultCode is not SUCCESS and management frame protection is not in use the state for the STA or the non-AP MLD shall be set to State 3 if it was State 4.
* If the ResultCode in the MLME-ASSOCIATE.response primitive is SUCCESS and RSNA establishment is required, and FILS authentication was not used(11ai), the SME shall attempt a 4-way handshake with the STA or the non-AP MLD. Upon a successful completion of the(11ai) 4-way handshake, the SME shall enable protection by issuing an MLME-SETPROTECTION.request(Rx\_Tx) primitive. If FILS authentication was used, the SME shall enable protection by generating an MLME-SETPROTECTION.request(Rx\_Tx) primitive. In either case, upon receipt of the MLME-SETPROTECTION.request(Rx\_Tx) primitive(MDR2), the MLME shall set the state for the STA or the non-AP MLD to State 4(11ai).
* AP or AP MLD only: The SME shall inform the DS of any changes in the state of the STA or the non-AP MLD.
* Non-AP, non-AP MLD and non-PCP STA reassociation initiation procedures

Except when the association is part of a fast BSS/ML transition, the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA(#2116) and temporal keys held for communication with the AP, AP MLD, or PCP by using the MLME-DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)) before invoking an MLME-REASSOCIATE.request primitive.

If dot11InterworkingServiceActivated is true and the STA was associated to the ESS for unsecured access to emergency services, the SME shall submit the MLME-REASSOCIATE.request primitive with EmergencyServices parameter set to true.

The MM-SME of a non-AP and non-PCP STA may include an MMS element in an MLME-REASSOCIATE.request primitive. The MM-SME shall include in the MMS element the MAC address associated with the MLME SAP instance to which the primitive is submitted.

Upon receipt of an MLME-REASSOCIATE.request primitive that is part of an on-channel tunneling (see 11.32.5 (On-channel Tunneling (OCT) operation)), a non-AP and non-PCP STA shall follow the rules in 11.32.5 (On-channel Tunneling (OCT) operation) in addition to the reassociation procedures described below.

Upon receipt of an MLME-REASSOCIATE.request primitive, a non-AP, non-AP MLD, and non-PCP STA shall reassociate with an AP, AP MLD, or PCP, respectively, using the following procedure:

* If the STA, non-AP MLD, or STA is not associated in the same ESS or the state for the new AP, AP MLD, or PCP(#2582), respectively, is State 1, the MLME shall inform the SME of the failure of the reassociation by issuing an MLME-REASSOCIATE.confirm primitive, and this procedure ends.
* The MLME shall transmit a Reassociation Request frame to the new AP, AP MLD, or PCP. (#2201)The RSNE contained in the MLME-ASSOCIATE.request primitive shall be included in the Reassociation Request frame. The RSNE shall specify exactly one pairwise cipher suite and exactly one AKM suite. If the MLME-REASSOCIATE.request primitive contained the EmergencyServices parameter equal to true, an Interworking element with the UESA field set to 1 shall be included in the Reassociation Request frame.
* If a Reassociation Response frame is received with a status code of SUCCESS, the state variable for the new AP, AP MLD, or PCP shall be set to State 4 or to State 3 if dot11RSNAActivated is true and the FT protocol is not used with respect to the new AP, AP MLD, or PCP and, unless the old AP, AP MLD, or PCP and new AP, AP MLD, or PCP, respectively, are the same, to State 2 with respect to the old AP, AP MLD, or PCP, and the MLME shall issue an MLME-REASSOCIATE.confirm primitive to inform the SME of the successful completion of the reassociation.

If the MLME-REASSOCIATION.request primitive has the new AP’s, AP MLD’s or PCP’s(#2582) MAC address in the CurrentAPAddress parameter (reassociation to the same AP, AP MLD or PCP(#2582)), the following states, agreements and allocations shall be deleted or reset to initial values:

* All EDCAF state
* Any block ack agreements that are not GCR agreements(#2228)
* Sequence number
* Packet number
* Duplicate detection caches
* Anything queued for transmission
* Fragmentation and reassembly buffers
* Power management mode
* WNM sleep mode
* TPKSAs established with any peers(#59)
* TSPECs(#17)
* DMG TSPECs(#17)
* GLK-GCR agreement.(11ak)

(#2250)If the reassociation is to the same AP (as described above), the following states, agreements and allocations are not affected by the reassociation procedure:

* PSMP sessions
* Enablement/Deenablement
* GDD enablement
* TDLS agreements(#59)
* MMSLs
* GCR agreements that are not GLK-GCR agreements(#2228)
* DMS agreements
* TFS agreements
* FMS agreements
* Triggered autonomous reporting agreements
* FTM sessions
* DMG SP and CBAP allocations
* PTP TSPECs.(#17)

(#2582)In the case of reassociation to a different APor PCP (the CurrentAPAddress parameter is not the new AP’s or PCP’s MAC address), all the states, agreements and allocations listed above are deleted or reset to initial values(#179).

* If a Reassociation Response frame is received with a status code of SUCCESS, a DMG STA shall write to each of the following MIB attributes (M101)the corresponding subfield of the DMG BSS Parameter Configuration field of the DMG Operation element received from the AP or PCP to which it requested reassociation:
* dot11PSRequestSuspensionInterval from the PSRequestSuspensionInterval subfield
* dot11MinBHIDuration from the MinBHIDuration subfield
* dot11BroadcastSTAInfoDuration from the BroadcastSTAInfoDuration subfield
* dot11AssocRespConfirmTime from the AssocRespConfirmTime subfield
* dot11MinPPDuration from the MinPPDuration subfield
* dot11SPIdleTimeout from the SPIdleTimeout subfield
* dot11MaxLostBeacons from the MaxLostBeacons subfield
* (#181)If an Association Response frame is received with a status code of SUCCESS at an (Ed)MM-SME coordinated STA and (M101)the Single AID field within the MMS element is equal to 1, then
* For each of its MAC entities advertised within the MMS element and for which dot11RSNAActivated is true, the state is set to State 3. Progress from State 3 to State 4 occurs independently in each such MAC entity.
* For each of its MAC entities advertised within the MMS element and for which dot11RSNAActivated is false, the state is set to State 4.
* For each of its MAC entities advertised within the MMS element the state for any other AP or PCP which is State 3 or State 4 prior to the association request shall be set to State 2.
* If a Reassociation Response frame is received with a status code other than SUCCESS or the reassociation fails to complete within dot11AssociationResponseTimeout:
* Except when the association is part of a fast BSS/ML transition, the state for the AP, AP MLD, or PCP shall be set to State 2 with respect to the new AP, AP MLD, or PCP.
* The MLME shall issue an MLME-REASSOCIATE.confirm primitive to inform the SME of the failure of the reassociation. The ResultCode returned in the MLME-REASSOCIATE.confirm primitive indicates the cause of the failed reassociation attempt. Any misconfiguration or parameter mismatch, e.g., data rates required as basic rates that the STA did not indicate as supported in the STA’s Supported Rates and BSS Membership Selectors element, shall be corrected before the SME issues an MLME-REASSOCIATE.request primitive for the same AP, AP MLD, or PCP. If the status code indicates the reassociation failed because of a reason that is not related to configuration (e.g., the AP or PCP is unable to support additional associations) and the Reassociation Response frame does not include a Timeout Interval element with Timeout Interval Type equal to 3 the SME shall not issue an MLME-REASSOCIATE.request primitive for the same AP, or AP MLD, or PCP until a period of at least 2 s has elapsed. If the status code indicates the reassociation failed and the Reassociation Response frame contains a Timeout Interval element with Timeout Interval Type equal to 3, the SME shall not issue an MLME-REASSOCIATE.request primitive for the same AP, AP MLD, or PCP until the period specified in the Timeout Interval element has elapsed.
* If an MLME-REASSOCIATE.confirm primitive is received with a ResultCode of SUCCESS, and RSNA is required, and FILS authentication was not used(11ai), and the STA or the non-AP MLD is in State 3, then the SME shall perform a 4-way handshake to establish an RSNA with the STA or the AP MLD. As a part of a successful 4-way handshake, the SME shall enable protection by generating(11ai) an MLME-SETPROTECTION.request(Rx\_Tx) primitive. If an MLME-REASSOCIATE.confirm primitive is received with a ResultCode of SUCCESS, and FILS authentication was used, and the STA is in State 3, then the SME shall enable protection by generating an MLME-SETPROTECTION.request(Rx\_Tx) primitive(11ai).
* Upon receipt of the MLME-SETPROTECTION.request(Rx\_Tx) primitive(MDR2), the MLME shall set the state of the STA or the AP MLD to State 4.
* AP, AP MLD, or PCP reassociation receipt procedures

Upon receipt of a Reassociation Request frame from a STA, a non-AP MLD, or a STA, the AP, AP MLD, or PCP, respectively, shall use the following procedure:

* The MLME shall issue an MLME-REASSOCIATE.indication primitive to inform the SME of the reassociation request. The SME shall issue an MLME-REASSOCIATE.response primitive addressed to the STA or the non-AP MLD identified by the PeerSTAAddress parameter of the MLME-REASSOCIATE.indication primitive. If the reassociation is not successful, the SME shall indicate a specific reason for the failure to reassociate in the ResultCode parameter. Upon receipt of the MLME-REASSOCIATE.response primitive, the MLME shall transmit a Reassociation Response frame.
* If the state for the STA is 1 and the STA is a non-DMG STA or the state for the non-AP MLD is 1, the SME shall refuse the reassociation request by issuing an MLME REASSOCIATE.response primitive with ResultCode NOT\_AUTHENTICATED.
* AP with dot11InterworkingServiceActivated true only: If the MLME-REASSOCIATE.indication primitive has the EmergencyServices parameter set to true and the RSN parameter does not include an RSNE, the SME shall not reject the reassociation request on the basis that dot11RSNAActivated is true and dot11PrivacyInvoked is true thereby granting access, using unprotected frames (see 9.2.4.1.9 (Protected Frame subfield)), to the network for emergency services purposes.
* Otherwise, in an RSNA the SME shall check the values received in the RSN parameter to see whether the values received match the security policy. If they do not, SME shall refuse the reassociation by issuing an MLME-REASSOCIATE.response primitive with a ResultCode indicating the security policy mismatch.
* Otherwise, if the state for the STA is 4, the STA has a valid security association, the STA has negotiated management frame protection, the reassociation is not a part of a fast BSS transition, and there has been no earlier, timed out SA Query procedure with the STA (which would have allowed a new reassociation process to be started, without an additional SA Query procedure):
* The SME shall refuse the reassociation request by issuing an MLME-REASSOCIATE.response primitive with ResultCode REFUSED\_TEMPORARILY and TimeoutInterval containing a Timeout Interval element with the Timeout Interval Type field set to 3 (Association Comeback time). If the SME is in an ongoing SA Query with the STA, the Timeout Interval Value field shall be set to the remaining SA Query period, otherwise it shall be set to dot11AssociationSAQueryMaximumTimeout.
* The state for the STA shall be left unchanged.
* Following this, if the SME is not in an ongoing SA Query with the STA, the SME shall issue one MLME-SA-QUERY.request primitive addressed to the STA every dot11AssociationSAQueryRetryTimeout TUs until an MLME-SA-QUERY.confirm primitive for the STA is received or dot11AssociationSAQueryMaximumTimeout TUs from the beginning of the SA Query procedure have passed. The SME shall increment the TransactionIdentifier by 1 for each MLME-SA-QUERY.request primitive, rolling it over to 0 after the maximum allowed value is reached.
* If no MLME-SA-QUERY.confirm primitive for a STA is received within the dot11AssociationSAQueryMaximumTimeout period, the SME shall allow a subsequent reassociation process to be started without starting an additional SA Query procedure, except that the SME may deny a subsequent reassociation process with the STA if an MSDU was received from the STA within this period.

NOTE—Reception of an MSDU implies reception of a valid protected frame, which obviates the need for the SA Query procedure.

* The SME shall refuse a reassociation request from a STA or a non-AP MLD that does not support all the rates in the BSSBasicRateSet parameter (11ak)and all of the membership selectors in the BSSMembershipSelectorSet parameter of the AP or the corresponding AP in each setup link, respectively, in the MLME-START.request primitive.
* The SME shall refuse a reassociation request from an HT STA or a non-AP MLD that does not support all of the MCSs in the Basic HT-MCS Set field of the HT Operation parameter of the AP or the corresponding AP in each setup link, respectively, in the MLME-START.request primitive.
* The SME shall refuse a reassociation request from a VHT STA or a non-AP MLD that does not support all of the <VHT-MCS, NSS> tuples indicated by the Basic VHT-MCS And NSS Set field of the VHT Operation parameter of the AP or the corresponding AP in each setup link, respectively, in the MLME-START.request primitive.
* If the ResultCode in the MLME-REASSOCIATE.response primitive is SUCCESS, the SME has an existing SA with the STA, and an SA Query procedure with that STA has failed to receive a valid response (i.e., has not received an MLME-SA-QUERY.confirm primitive within the dot11AssociationSAQueryMaximumTimeout period), the SME shall issue an MLME-DISASSOCIATE.request primitive addressed to the STA with ReasonCode INVALID\_AUTHENTICATION.

NOTE—This MLME-DISASSOCIATE.request primitive generates a protected Disassociation frame. If the reassociation request was genuine, the STA has deleted the PTKSA by this point and so the protected Disassociation frame is ignored. The purpose is to inform a STA which has for some reason failed to respond to an SA Query procedure triggered by a forged reassociation request.

* If the ResultCode in the MLME-REASSOCIATE.response primitive is SUCCESS and the reassociation is not part of a fast BSS/ML transition, the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA(#2116) and temporal keys held for communication with the STA or the non-AP MLD by using the MLME-DELETEKEYS.request primitive (see 11.5.18 (RSNA security association termination)).
* If the MLME-REASSOCIATE.indication primitive includes an MMS parameter, the AP or PCP shall take the following additional action, as appropriate:
* If the Single AID field in the MMS parameter of the MLME-REASSOCIATE.indication primitive is equal to 1, the AP or PCP may allocate a single AID for all of the STAs included in the MMS element. If the AP or PCP allocates the same AID to all STAs whose MAC address was included in the MMS element, it shall include the MMS element received from the MM-SME coordinated STA in the MLME-REASSOCIATE.response primitive.
* If the Single AID field is 0, the AP or PCP shall allocate a distinct AID for each STA specified in the MMS element.
* If a Reassociation Response frame with a status code of SUCCESS is acknowledged by the STA or the non-AP MLD, the state for the STA or the non-AP MLD shall be set to State 4, or to State 3 if dot11RSNAActivated is true and the reassociation is not part of a fast BSS/ML transition.
* If the ResultCode in the MLME-REASSOCIATE.response primitive is not SUCCESS and management frame protection is in use the state for the STA or the non-AP MLD shall be left unchanged. If the ResultCode is not SUCCESS, management frame protection is not in use, and the reassociation is part of a fast BSS/ML transition, the state for the STA or the non-AP MLD shall be left unchanged. If the ResultCode is not SUCCESS, management frame protection is not in use, and the reassociation is not part of a fast BSS/ML transition, the state for the STA or the non-AP MLD shall be set to State 3 if it was State 4.
* If the ResultCode in the MLME-REASSOCIATE.response primitive is SUCCESS, RSNA establishment is required, and the reassociation is not part of a fast BSS/ML transition, and FILS is not in use(11ai), the SME shall attempt a 4-way handshake with the STA or the non-AP MLD. Upon a successful completion of a 4-way handshake, the SME shall enable protection by issuing an MLME-SETPROTECTION.request(Rx\_Tx) primitive. If FILS authentication was used, the SME shall enable protection by generating an MLME-SETPROTECTION.request(Rx\_Tx) primitive. In either case, upon receipt of the MLME-SETPROTECTION.request(Rx\_Tx) primitive(MDR2), the MLME shall set the state for the STA or the non-AP MLD to State 4(11ai).
* AP or AP MLD only: The SME shall inform the DS of any changes in the state of the STA or the non-AP MLD.
* (#1454)(#2582)If the ResultCode in the MLME-REASSOCIATE.response primitive is SUCCESS and the CurrentAPAddress parameter in the MLME-REASSOCIATION.indication primitive is this (#2599)AP’s, AP MLD’s or PCP’s MAC address (reassociation to the same AP, AP MLD, or PCP, respectively), the AP, AP MLD, or PCP, respectively, shall match the non-AP STA’s, non-AP MLD’s, or non-AP STA’s, respectively, treatment of the listed agreements and allocations as described in 11.3.5.4 (Non-AP and non-PCP STA reassociation initiation procedures) item c). The AP, AP MLD, or PCP deletes or resets to initial values those items that the non-AP STA, non-AP MLD, or non-AP STA, respectively, is required in 11.3.5.4 (Non-AP and non-PCP STA reassociation initiation procedures) item c) to delete or reset to initial values, and the AP, AP MLD, or PCP does not modify the states, agreements and allocations that are listed as not affected by the reassociation procedure.
* (#1454)If the ResultCode in the MLME-REASSOCIATE.response primitive is SUCCESS and the CurrentAPAddress parameter in the MLME-REASSOCIATION.indication primitive is not this AP’s, AP MLD’s or PCP’s MAC address (reassociation to a different AP or PCP(#2582)), all the states, agreements and allocations pertaining to the associating STA, non-AP MLD, or STA, respectively, and listed in both numbered lists in 11.3.5.4 (Non-AP and non-PCP STA reassociation initiation procedures) item c) are deleted or reset to initial values.
* Non-AP, non-AP MLD, and non-PCP STA disassociation initiation procedures

The SME shall issue an MLME-DISASSOCIATE.request primitive that includes an appropriate Reason Code as defined in Table 9-51 (Reason codes) of 9.4.1.7 (Reason Code field).

Upon receipt of an MLME-DISASSOCIATE.request primitive, a non-AP, non-AP MLD, and non-PCP STA’s MLME shall disassociate from an AP, AP MLD, or PCP, respectively, using the following procedure:

* If the state for the AP, AP MLD, or PCP is State 3 or State 4, the MLME shall transmit a Disassociation frame to the AP, AP MLD, or PCP.
* The state for the AP, AP MLD, or PCP shall be set to State 2 if it was not State 1. In the case of an MM-SME coordinated STA, the MLME shall perform this for each STA whose address was included in the MMS parameter of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive that established the association.
* The MLME shall issue an MLME-DISASSOCIATE.confirm primitive to inform the SME of the successful completion of the disassociation.
* Upon receiving an MLME-DISASSOCIATE.confirm primitive, the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA(#2116) and temporal keys held for communication with the AP, AP MLD, or PCP by using the MLME‑DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)) and by invoking an MLME-SETPROTECTION.request(None) primitive. In the case of an MM-SME coordinated STA, the MLME shall perform this for each STA whose address was included in the MMS parameter of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive that established the association.
* Non-AP, non-AP MLD, and non-PCP STA disassociation receipt procedure

Upon receipt of a Disassociation frame from an AP, AP MLD, or PCP for which the state is State 3 or State 4, if management frame protection was not negotiated when the PTKSA(s) were created, or if management frame protection is in use and the frame is not discarded per management frame protection processing, a non-AP, non-AP MLD, and non-PCP STA, respectively, shall disassociate from the AP, AP MLD, or PCP using the following procedure:

* The state for the AP, AP MLD, or PCP shall be set to State 2.
* The MLME shall issue an MLME-DISASSOCIATE.indication primitive to inform the SME of the disassociation.
* Upon receiving the MLME-DISASSOCIATE.indication primitive, the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA(#2116) and temporal keys held for communication with the AP, AP MLD, or PCP by using the MLME‑DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)) and by invoking an MLME-SETPROTECTION.request(None) primitive. The MM-SME shall perform this process for each STA whose address was included in the MMS parameter of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive that established the association.
* If the reason code indicates a configuration or parameter mismatch as the cause of the disassociation, the SME shall not attempt to associate or reassociate with the AP, AP MLD, or PCP until the configuration or parameter mismatch has been corrected.
* If the reason code indicates the STA, non-AP MLD, STA, was disassociated for a reason other than configuration or parameter mismatch, the SME shall not attempt to associate or reassociate with the AP, AP MLD, or PCP until a period of 2 s has elapsed.
* AP, AP MLD, or PCP disassociation initiation procedure

The SME shall issue an MLME-DISASSOCIATE.request primitive that includes an appropriate Reason Code as defined Table 9-51 (Reason codes) of 9.4.1.7 (Reason Code field).

Upon receipt of an MLME-DISASSOCIATE.request primitive, an AP, AP MLD, or PCP shall disassociate a STA, a non-AP MLD, or STA, respectively, using the following procedure:

* If the state for the STA, non-AP MLD, or STA is State 3 or State 4, the AP, AP MLD, or PCP, respectively, shall generate a Disassociation frame to be transmitted to the indicated STA, the indicated non-AP MLD, or the indicated STA, respectively.

NOTE—As the Disassociation frame is a bufferable MMPDU, the transmission of this frame might be delayed by the operation of a power saving(M101) protocol. The AID and the PTKSA are maintained (when applicable) until the frame is acknowledged or attempts to transmit the frame are abandoned.

* The state for the STA, non-AP MLD, or STA shall be set to State 2, if it was not State 1. The MM-SME shall perform this process for each STA whose address was included in the MMS parameter of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive that established the association.
* Once the Disassociation frame is acknowledged or attempts to transmit the frame are abandoned, the MLME shall issue an MLME-DISASSOCIATE.confirm primitive to inform the SME of the disassociation.
* Upon receiving an MLME-DISASSOCIATE.confirm primitive, the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA(#2116) and temporal keys held for communication with the STA or the non-AP MLD by using the MLME‑DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)) and by invoking an MLME-SETPROTECTION.request(None) primitive. The MM-SME shall perform this process for each STA whose address was included in the MMS parameter of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive that established the association.
* Upon receiving an MLME-DISASSOCIATE.confirm primitive, the SME shall release the AID assigned for the indicated STA or the indicated non-AP MLD if the state for the indicated STA or the indicated non-AP MLD, respectively was State 3 or State 4.
* AP or AP MLD only: The SME shall inform the DS of the disassociation.
* AP, AP MLD, or PCP disassociation receipt procedure

Upon receipt of a Disassociation frame from a STA, a non-AP MLD, or a STA for which the state is State 3 or State 4, if management frame protection was not negotiated when the PTKSA(s) were created, or if management frame protection is in use and the frame is not discarded per management frame protection processing, the AP, AP MLD, or PCP, respectively, shall disassociate the STA, the non-AP MLD, or the STA using the following procedure:

* The state for the STA, the non-AP MLD, or the STAshall be set to State 2. The MM-SME shall perform this process for each STA whose address was included in the MMS parameter of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive that established the association.
* The MLME shall issue an MLME-DISASSOCIATE.indication primitive to inform the SME of the disassociation.
* Upon receiving an MLME-DISASSOCIATE.indication primitive the SME shall delete any PTKSA, GTKSA, IGTKSA, BIGTKSA(#2116) and temporal keys held for communication with the STA, the non-AP MLD, or the STA, respectively by using the MLME‑DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)) and by invoking an MLME-SETPROTECTION.request(None) primitive. The MM-SME shall perform this process for each STA whose address was included in the MMS parameter of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive that established the association.
* AP or AP MLD only: The SME shall inform the DS of the disassociation.
* The SME shall release the AID assigned for the indicated STA, the indicated non-AP MLD, or the indicated STA, respectively.