IEEE P802.11  
Wireless LANs

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Proposed Texts for CNSA 2.0 | | | | | | Date: 2025-09-13 | | | | | | Author(s): | | | | | | Name | Affiliation | Address | Phone | email | | Po-Kai Huang | Intel |  |  | po-kai.huang@intel.com | | Ido Ouzieli |  |  | ido.ouzieli@intel.com | | Johannes Berg |  |  | johannes.berg@intel.com | |  |  |  |  | |  |  |  |  |  | |

Abstract

This submission proposes texts to satisfy requrieemnts of CNSA 2.0.

**Revision History:**

R0: Initial version.

## Discussion:

CNSSP 15 states that by January 1, 2027, all new acquisitions for NSS will be required to be CNSA 2.0 compliant unless otherwise noted. [1] In terms of product readiness to ship the equipment to the government in 2027 time frame, we do not need every item defined in 11bt PAR, and as a minimum, we only need AKMs for CNSA 2.0, use SHA-384 and migrate DH exchange to ML-KEM-1024 exchange. This document proposes spec texts that will meet the CNSA 2.0 timeline of January 1 2027.

* Two AKMs
  + Counterparts of AKM 12 and 13 (FT roaming case).
* Reuse DH parameter element to carry ML KEM parameter (encapsulation key/ciphertext) if derivation of shared secrect for perfect forward secrecy is required.
  + Make the element fragmentable when carrying ML KEM parameter
  + General flow is shown below as described in FIPS 203.
  + Suitable for the purpose of PFS

A diagram of a key generation

AI-generated content may be incorrect.

* The proposed texts in this document does not address Hybrid
  + Based on [1], “NSA has confidence in CNSA 2.0 algorithms and will not require NSS developers to use hybrid certified products for security purposes. However, product availability and interoperability requirements may lead to adopting hybrid solutions. NSA recognizes that some protocols may require using hybrid-like constructions to accommodate the larger sizes of ML-KEM-1024 or ML-DSA-87 and will work with industry and SDOs to identify the best options for implementation.”
  + Hybrid can be done by simply adding two elements to derive two shared secrect
* 11bn texts to be added when 11bn texts are stable.

[1] https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/0/CSI\_CNSA\_2.0\_FAQ\_.PDF

## Proposed Text:

**TGbt Editor: *Instruction: Modify 9.4.1.9 as shown below***

* Status Code field

(…existing texts…)

|  |  |  |
| --- | --- | --- |
| * Status codes | | |
| Status code | Name | Meaning |
| … |  |  |
| 77 | UNSUPPORTED\_FINITE\_CYCLIC\_GROUP | Authentication is rejected because the offered finite cyclic group is not supported. |
| …. |  |  |
| 136 | INVALID\_PUBLIC\_KEY | Public key format is invalid. |
| …. |  |  |
| <ANA> | UNSUPPORTED\_ML\_KEM\_PARAMETER | Authentication is rejected because the offered ML-KEM parameter is not supported. |
| <ANA> | INVALID\_ML\_KEM\_PARAMETER | ML-KEM parameter format is not valid. |

**TGbt Editor: *Instruction: Modify 9.4.2.1 as shown below***

* Elements
* General

(…existing texts…)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * Element IDs | | | | |
| Element | Element ID | Element ID Extension | Extensible | Fragmentable |
| …… |  |  |  |  |
| Diffie-Hellman/ML-KEM Parameter (see 9.4.2.312 (Diffie-Hellman/ML-KEM Parameter element)) | 255 | 32 | No | See Note 3. |
| ….. |  |  |  |  |
| NOTE 1—See 10.28.6 (Element parsing) on the parsing of elements.  (#11be)NOTE 2—Yes for a Reassociation Response frame transmitted during ML resetup by an AP affiliated with an AP MLD, in response to a Reassociation Request frame received from a non-AP STA affiliated with a non-AP MLD. No otherwise.  NOTE 3 – Yes if the Group/ML-KEM field indicates ML-KEM. No otherwise. | | | | |

**TGbt Editor: *Instruction: Modify 9.4.2.23.2 as shown below***

9.4.2.23.2 Cipher suites

(…existing texts…)

NOTE—If the AKM Suite List field is present, the Group Data Cipher Suite field and the Pairwise Cipher Suite List field are present. If AKM 00-0F-AC:11, 00-0F-AC:12, 00-0F-AC:13, 00-0F-AC:<ANA#1>, or 00-0F-AC:<ANA#2> is present in the AKM Suite List field, and the MFPC bit of the RSN Capabilities field is 1, the default group management cipher suite is not suitable.

**TGbt Editor: *Instruction: Modify 9.4.2.23.3 as shown below***

* AKM suites

(…existing texts…)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * AKM suite selectors | | | | | | |
| OUI | Suite type | Meaning | | | Authentication algorithm numbers  (see 9.4.1.1 (Authentication Algorithm Number field)) | Cipher suite selector and key exchange methodrestriction |
| Authentication  type | Key management  type | Key derivation type |
| ….. |  |  |  |  |  |  |
| 00-0F-AC | 12 | Authentication negotiated over  IEEE Std 802.1X using a CNSA Suite compliant EAP method | RSNA key management as defined in 12.7 (Keys and key distribution) | Defined in 12.7.1.6.2 (Key derivation function (KDF))  using SHA-384 | 0 (open) or 8  (IEEE 802.1X) | Used only with cipher suite selector values 00-0F-AC:9 (GCMP-256), 00-0F-AC:10 (CCMP-256), 00-0F-AC:13 (BIP-CMAC-256), and 00-0F-AC:12 (BIP-GMAC-256) |
| 00-0F-AC | 13 | FT authentication negotiated over  IEEE Std 802.1X | FT key management as defined in 12.7.1.6 (FT key hierarchy) | Defined in 12.7.1.6.2 (Key derivation function (KDF))  using SHA-384 | 2 (FT) for FT  protocol  reassociation as  defined in 13.5 (FT  protocol)  0 (open) or 8  (IEEE 802.1X) for  FT Initial Mobility  Domain Association  over  IEEE Std 802.1X or  PMKSA caching | Used only with cipher suite selector values 00-0F-AC:9 (GCMP-256), 00-0F-AC:10 (CCMP-256), 00-0F-AC:13 (BIP-CMAC-256), and 00-0F-AC:12 (BIP-GMAC-256) |
| …. |  |  |  |  |  |  |
| 00-0F-AC | <ANA#1> | Authentication negotiated over  IEEE Std 802.1X using a CNSA Suite 2.0 compliant EAP method | RSNA key management as defined in 12.7 (Keys and key distribution) | Defined in 12.7.1.6.2 (Key derivation function (KDF))  using SHA-384 | 0 (open) or 8  (IEEE 802.1X) | Used only with cipher suite selector values 00-0F-AC:9 (GCMP-256), 00-0F-AC:10 (CCMP-256), 00-0F-AC:13 (BIP-CMAC-256), and 00-0F-AC:12 (BIP-GMAC-256) and used only with ML-KEM-1024 key exchange method |
| 00-0F-AC | <ANA#2> | FT authentication negotiated over  IEEE Std 802.1X | FT key management as defined in 12.7.1.6 (FT key hierarchy) | Defined in 12.7.1.6.2 (Key derivation function (KDF))  using SHA-384 | 2 (FT) for FT  protocol  reassociation as  defined in 13.5 (FT  protocol)  0 (open) or 8  (IEEE 802.1X) for  FT Initial Mobility  Domain Association  over  IEEE Std 802.1X or  PMKSA caching | Used only with cipher suite selector values 00-0F-AC:9 (GCMP-256), 00-0F-AC:10 (CCMP-256), 00-0F-AC:13 (BIP-CMAC-256), and 00-0F-AC:12 (BIP-GMAC-256) and used only with ML-KEM-1024 key exchange method |

(…existing texts…)

**TGbt Editor: *Instruction: Modify 9.4.2.312 as shown below***

* Diffie-Hellman/ML-KEM Parameter element

The Diffie-Hellman/ML-KEM Parameter element contains a Diffe-Hellman public value and an indicator of the finite cyclic group from which it was obtained or a ML-KEM parameter and an indicator of ML-KEM from which it was obtained. See Figure 9-1119 (Diffie-Hellman Parameter element format).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Element ID | Length | Element ID Extension | Group/ML-KEM | Public Key/ML-KEM Parameter |
| Octets: | 1 | 1 | 1 | 2 | variable |
| * Diffie-Hellman/ML-KEM Parameter element format | | | | | |

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

The Group/ML-KEM field is a 16-bit unsigned integer that maps an identifying number from the “Transform Type 4 – Key Exchange Method Transform IDs” registry maintained by IANA for IETF RFC 7296 to a complete domain parameter set.

The Public Key/ML-KEM Parameter field is a Diffie-Hellman public key, an element in the group described by the domain parameter set indicated by the value in the Group/ML-KEM field or a ML-KEM parameter, which is either an encapsulation key or a ciphertext described by the ML-KEM indicated in the Group/ML-KEM field.

**TGbt Editor: *Instruction: change “Diffie-Hellman Parameter element” to “Diffie-Hellman/ML-KEM Parameter element”***

**TGbt Editor: *Instruction: Modify 12.7.1.3 as shown below***

12.7.1.2 PRF

(..existing texts…)

When the negotiated AKM is 00-0F-AC:12, 00-0F-AC:20, or 00-0F-AC:<ANA#1>, the KDF specified in 12.7.1.6.2 (Key derivation function (KDF)) shall be used instead of the PRF construction defined here. In this case, A is used as the KDF label and B as the KDF context, and the PRF function is defined as follows:

PRF-704(K, A, B) = KDF-SHA-384-704(K, A, B)

When the negotiated AKM is 00-0F-AC:13 or 00-0F-AC:<ANA#2>, the KDF specified in 12.7.1.6.2 (Key derivation function (KDF)) shall be used instead of the PRF construction defined here. In this case, A is used as the KDF label and B as the KDF context, and the PRF functions are defined as follows:

PRF-384(K, A, B) = KDF-SHA-384-384(K, A, B)

PRF-512(K, A, B) = KDF-SHA-384-512(K, A, B)

PRF-704(K, A, B) = KDF-SHA-384-704(K, A, B)

(..existing texts…)

**TGbt Editor: *Instruction: Modify 12.7.1.3 as shown below***

* Pairwise key hierarchy

Except when preauthentication or FILS authentication is used, the pairwise key hierarchy utilizes PRF-384, PRF-512, or PRF-704 to derive session specific keys from a PMK, as depicted in Figure 12-33 (Pairwise key hierarchy). When using AKM 00-0F-AC:12, 00-0F-AC:15, 00-0F-AC:20, 00-0F-AC:23, AKM 00-0F-AC:<ANA#1>, the length of the PMK, PMK\_bits, shall be 384 bits. When using AKM 00-0F-AC:24 or 00-0F-AC:25, the length of the PMK, PMK\_bits, shall have the length of the digest generated by H() identified in 12.4.2 (Assumptions on SAE). When using AKM suite selectors for which the Authentication type column indicates FT authentication (see Table 9-208 (AKM suite selectors)), the FT key hierarchy is used to derive session specific keys from an MPMK as defined in 12.7.1.6 (FT key hierarchy). With all other AKM suite selectors, the length of the PMK, PMK\_bits, shall be 256 bits. The pairwise key hierarchy takes a PMK and generates a PTK.

(..existing texts…)

PTK = PRF-Length(PMK, “Pairwise key expansion”, Min(AA,SPA) || Max(AA,SPA) || Min(ANonce,SNonce) || Max(ANonce,SNonce) || DHss/MLKEMss) if key derivation with Authentication frame exchange for IEEE 802.1X is used as defined in 12.16.8.2 (IEEE 802.1X).

Otherwise, PTK = PRF-Length(PMK, “Pairwise key expansion”, Min(AA,SPA) || Max(AA,SPA) || Min(ANonce,SNonce) || Max(ANonce,SNonce))

(..existing texts…)

When the negotiated AKM is 00-0F-AC:12 or 00-0F-AC:<ANA#1>, and the PMK identifier is defined as

PMKID = Truncate-128(HMAC-SHA-384(PTK-KCK, “PMK Name” || AA || SPA))

(..existing texts…)

**TGbt Editor: *Instruction: Modify 12.7.1.6.5 as shown below***

* FT key hierarchy

**12.7.1.6.3 PMK-R0**

(…existing texts…)

If the negotiated AKM is 00-0F-AC:13 or 00-0F-AC:<ANA#2>, then Q = 384 and

— MPMK = ExtractBits(MSK, 0, 384)

— PMKID = Truncate-128(HMAC-SHA-384(MPMK, “PMK Name” || AA || SPA))

(…existing texts…)

* PTK

(…existing texts…)

PTK = KDF-*Hash*-*Length*(PMK-R1, “FT-PTK”, SNonce || ANonce || BSSID || STA-ADDR || DHss/MLKEMss) if key derivation with Authentication frame exchange for FT is used as defined in 12.16.8.1 (FT protocol).

Otherwise, PTK = KDF-*Hash-Length*(PMK-R1, “FT-PTK”, SNonce || ANonce || BSSID || STA-ADDR)

(…existing texts…)

**TGbt Editor: *Instruction: Modify 12.7.3 as shown below***

* EAPOL-Key PDU construction and processing

(…existing texts…)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| * Integrity and key wrap algorithms | | | | | | | |
| AKM | Integrity algorithm | KCK\_bits | Size of MIC  (octets) | Key wrap algorithm | KEK\_bits | KCK2\_bits | KEK2\_bits |
| ….. |  |  |  |  |  |  |  |
| 00-0F-AC:12 | HMAC-SHA-384 | 192 | 24 | NIST AES Key Wrap | 256 | 0 | 0 |
| 00-0F-AC:13 | HMAC-SHA-384 | 192 | 24 | NIST AES Key Wrap | 256 | 0 | 0 |
| ….. |  |  |  |  |  |  |  |
| 00-0F-AC:<ANA#1> | HMAC-SHA-384 | 192 | 24 | NIST AES Key Wrap | 256 | 0 | 0 |
| 00-0F-AC: :<ANA#2> | HMAC-SHA-384 | 192 | 24 | NIST AES Key Wrap | 256 | 0 | 0 |
| NOTE 1—Keys derived using PASN AKM are not used to protect EAPOL frames.  NOTE 2—AES-SIV-256 and AES-SIV-512 are AEAD ciphers. | | | | | | | |

(…existing texts…)

**TGbt Editor: *Instruction: Modify 12.16.8.2 as shown below***

* FT protocol

If an FTO or FTR (see 13 (Fast BSS transition)) sets the (Re)Association Frame Encryption Support field in the RSNXE to 1, then the FTO or FTR supports the additional rules defined in this subclause.

An FTO that sets the (Re)Association Frame Encryption Support field in the RSNXE to 1 and receives an RSNXE from the FTR with the (Re)Association Frame Encryption Support field set to 1 shall:

* Include a Diffie-Hellman/ML-KEM Parameter element in the first message of the FT protocol (see 13.8 (FT authentication sequence)).
* If the AKM is not AKM 00-0F-AC:<ANA#2>, select a finite cyclic group in the Diffie-Hellman/ML-KEM Parameter element from the dot11RSNAConfigDLCGroupTable that is at least of the security strength provided by the AKM and cipher suites.
* If the AKM is AKM 00-0F-AC:<ANA#2>, indicate ML-KEM-1024 in the Group/ML-KEM field of the Diffie-Hellman/ML-KEM Parameter element
* If the AKM is not AKM 00-0F-AC:<ANA#2>, generate an ephemeral (random) private key with the chosen finite cyclic group, use the selected group’s scalar operation (see 12.4.4.1 (General)) with the private key to generate its ephemeral public key, and indicate the ephemeral public key in the Diffie-Hellman/ML-KEM Parameter element.
* If the AKM is AKM 00-0F-AC:<ANA#2>, generate the ephemeral decapsulation key and the ephemeral encapsulation key using the ML-KEM key generation algorithm, and indicate the ephemeral encapsulation key in the Diffie-Hellman/ML-KEM Parameter element.

Otherwise, an FTO that sets dot11EPPReAssociationFrameEncryptionSupportActivated to false or does not receive the RSNXE from the FTR with the (Re)Association Frame Encryption Support field set to 1 shall not include a Diffie-Hellman Parameter/ML-KEM element in the first message of the FT protocol.

For the purpose of interoperability, an FTO and an FTR shall support group 19, an ECC group defined over a 256-bit prime order field.

An FTR that sets the (Re)Association Frame Encryption Support field in the RSNXE to 1 and receives the first message of the FT protocol with the (Re)Association Frame Encryption Support field in the RSNXE set to 1 shall:

* If the AKM is not AKM 00-0F-AC:<ANA#2>, validate that finite cyclic group indicated in the Diffie-Hellman/ML-KEM Parameter element in the first message is supported (present in dot11RSNAConfigDLCGroupTable). Otherwise, the FTR shall reject the first message with status code set to UNSUPPORTED\_FINITE\_CYCLIC\_GROUP.
* If the AKM is AKM 00-0F-AC:<ANA#2>, validate that the Group/ML-KEM field of the Diffie-Hellman/ML-KEM Parameter element in the first message indicates ML-KEM-1024. Otherwise, the responder shall reject the first message with status code set to UNSUPPORTED\_ML\_KEM\_PARAMETER.
* If the AKM is not AKM 00-0F-AC:<ANA#2>, verify the public key indicated in the Diffie-Hellman/ML-KEM Parameter element in the first message as specified in 5.6.2.3 of NIST SP 800-56A R2. If verification fails, the FTR shall reject the first message with status code set to INVALID\_PUBLIC\_KEY.
* If the AKM is AKM 00-0F-AC:<ANA#2>, verify that the encapsulation key indicated in the Diffie-Hellman/ML-KEM Parameter element in the first message according to encapsulation key check as defined in 7.2 (ML-KEM Encapsulation) in NIST FIPS 203. If verification fails, the FTR shall reject the first message with status code set to INVALID\_ML\_KEM\_PARAMETER.
* If the AKM is not AKM 00-0F-AC:<ANA#2>, generate an ephemeral (random) private key with the chosen finite cyclic group and use the selected group’s scalar operation with the private key to generate its ephemeral public key if the first message is not rejected. Perform the group's scalar-op (see 12.4.4.1 (General)) with the FTO's ephemeral public key and its own ephemeral private key to produce an ephemeral Diffie-Hellman shared secret, DHss.
* If the AKM is AKM 00-0F-AC:<ANA#2>, using the FTO’s ephemeral encapsulation key and the ML-KEM encapsulation algorithm to generate the ephemeral ciphertext and the ephemeral ML-KEM shared secret, MLKEMss.
* Derive PTK with DHss/MLKEMss as defined in 12.7.1.6.5 (PTK).
* Irretrievably delete the shared secret, DHss/MLKEMss, upon completion of PTK generation.
* Include a Diffie-Hellman/ML-KEM Parameter element in the second message of the FT protocol (see 13.8 (FT authentication sequence)).
* If the AKM is not AKM 00-0F-AC:<ANA#2>, indicate the chosen finite cyclic group in the Diffie-Hellman/ML-KEM Parameter element of the second message, which is the same as the finite cyclic group in the Diffie-Hellman/ML-KEM Parameter element of the first message.
* If the AKM is AKM 00-0F-AC:<ANA#2>, indicate ML-KEM-1024 in the Group/ML-KEM field of the Diffie-Hellman/ML-KEM Parameter element
* If the AKM is not AKM 00-0F-AC:<ANA#2>, indicate its ephemeral public key in the Diffie-Hellman/ML-KEM Parameter element of the second message.
* If the AKM is AKM 00-0F-AC:<ANA#2>, indicate the ephemeral ciphertext in the Diffie-Hellman/ML-KEM Parameter element of the second message.
* Calculate the MIC in the FTE by using the key, the algorithm, and the MIC size as defined in 13.8.5 (FT authentication sequence: contents of fourth message) on the concatenation of the following data, in the order given here as the input:
* FTO’s MAC address.
* FTR’s MAC address.
* RSNE sent in the Beacons transmitted by the AP with MAC address equal to Address 1 field of the first message.
* RSNXE sent in the Beacons transmitted by the AP with MAC address equal to Address 1 field of the first message.
* the Frame Body field of the second message with MIC field of the FTE set to 0.
* Set the MIC field in the FTE as the calculated MIC.

Otherwise, an FTR that sets dot11EPPReAssociationFrameEncryptionSupportActivated to false or does not receive the RSNXE in the first message of the FT protocol with the (Re)Association Frame Encryption Support field set to 1 shall not include a Diffie-Hellman/ML-KEM Parameter element in the second message of the FT protocol.

After receiving the second message of the FT protocol with the status code set to SUCCESS, an FTO shall:

* Validate that there is a Diffie-Hellman/ML-KEM Parameter element included in the second message of the FT protocol if the FTO included a Diffie-Hellman/ML-KEM Parameter element in the first message of the FT protocol. If the validation fails, the FTO shall discard the frame and terminate further protocol processing.
* Validate that there is no Diffie-Hellman/ML-KEM Parameter element included in the second message of the FT protocol if the FTO did not include a Diffie-Hellman/ML-KEM Parameter element in the first message of the FT protocol. If the validation fails, the FTO shall discard the frame and terminate further protocol processing.
* Validate that the Group/ML-KEM field indication in the Diffie-Hellman/ML-KEM Parameter element in the second message is the same as the Group/ML-KEM field indication in the Diffie-Hellman/ML-KEM Parameter element in the first message if the FTO included a Diffie-Hellman/ML-KEM Parameter element in the first message of the FT protocol. If the validation fails, the FTO shall discard the frame and terminate further protocol processing.
* If the AKM is not AKM 00-0F-AC:<ANA#2>, verify the public key indicated in the Diffie-Hellman/ML-KEM Parameter element in the second message as specified in 5.6.2.3 of NIST SP 800-56A R2. If the verification fails, the FTO shall discard the frame and terminate further protocol processing.
* If the AKM is AKM 00-0F-AC:<ANA#2>, verify that the ephemeral ciphertext indicated in the Diffie-Hellman/ML-KEM Parameter element in the second message according to decapsulation input check as defined in 7.3 (ML-KEM Decapsulation) in NIST FIPS 203. If verification fails, the FTO shall discard the frame and terminate further protocol processing.
* If the AKM is not AKM 00-0F-AC:<ANA#2>, perform the group’s scalar-op (see 12.4.4.1 (General)) with the FTR’s ephemeral public key and its own ephemeral private key to produce an ephemeral Diffie-Hellman shared secret, DHss, if the second message is not discarded.
* If the AKM is AKM 00-0F-AC:<ANA#2>, using the FTR’s ephemeral ciphertext, its own ephemeral decapsulation key and the ML-KEM decapsulation algorithm to generate the ephemeral ML-KEM shared secret, MLKEMss, if the second message is not discarded.Derive PTK with DHss/MLKEMss as defined in 12.7.1.6.5 (PTK).
* Irretrievably delete the shared secret, DHss/MLKEMss, upon completion of PTK generation.
* Verify the MIC in the FTE using the S1KH of the FTO. If the verification fails, the FTO shall discard the frame and terminate further protocol processing.

**TGbt Editor: *Instruction: Modify 12.16.8.2 as shown below***

* + - 1. IEEE 802.1X

If an originator or a responder defined in 12.16.5 (IEEE 802.1X authentication utilizing Authentication frames) sets the (Re)Association Frame Encryption Support field in the RSNXE to 1, then the originator or the responder supports the additional rules defined in this subclause when performing IEEE 802.1X Authentication frame exchange.

An originator that sets the (Re)Association Frame Encryption Support field in the RSNXE to 1 receives an RSNXE from the responder with the (Re)Association Frame Encryption Support field set to 1, and intends to continue association after authentication shall do the following in the first Authentication frame:

* Include a Nonce element to indicate SNonce.
* Include an RSNE to indicate the AKM and the pairwise cipher suite. The Version field shall be set to 1. The Pairwise Cipher Suite Count field shall be set to 1. The AKM Suite Count field shall be set to 1. The PMKID count field and the PMKID List field is set corresponding to PMKSA identifiers if exists. All other fields shall be as specified in 9.4.2.5 (TIM element) and 12.6.3 (RSNA policy selection in an infrastructure BSS).
* Not include an AKM Suite Selector element.
* Include an RSNXE.
* Include a Diffie-Hellman/ML-KEM Parameter element.
* If the AKM is not AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2> , select a finite cyclic group in the Diffie-Hellman/ML-KEM Parameter element from the dot11RSNAConfigDLCGroupTable that is at least of the security strength provided by the AKM and cipher suites.
* If the AKM is AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, indicate ML-KEM-1024 in the Group/ML-KEM field of the Diffie-Hellman/ML-KEM Parameter element
* If the AKM is not AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, with the chosen finite cyclic group, generate an ephemeral (random) private key, use the selected group’s scalar operation (see 12.4.4.1 (General)) with the private key to generate its ephemeral public key, and indicate the ephemeral public key in the Diffie-Hellman/ML-KEM Parameter element.
* If the AKM is AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, generate the ephemeral decapsulation key and the ephemeral encapsulation key using the ML-KEM key generation algorithm, and indicate the ephemeral encapsulation key in the Diffie-Hellman/ML-KEM Parameter element.

Otherwise, an originator that sets dot11EPPReAssociationFrameEncryptionSupportActivated to false or does not receive the RSNXE from the responder with the (Re)Association Frame Encryption Support field set to 1 shall not include a Diffie-Hellman/ML-KEM Parameter element nor an RSNE nor an RSNXE nor a Nonce element in the first Authentication frame for IEEE 802.1X authentication.

For the purpose of interoperability, an authenticator and a supplicant shall support group 19, an ECC group defined over a 256-bit prime order field.

A responder that sets the (Re)Association Frame Encryption Support field in the RSNXE to 1 receives the first Authentication frame with a Nonce element, RSNE, RSNXE, and a Diffie-Hellman/ML-KEM Parameter element shall:

* Verify that the AKM indicated in the RSNE is supported. Otherwise, the responder shall reject the first message with status code set to STATUS\_INVALID\_AKMP.
* Verify that the pairwise cipher indicated in the RSNE is supported. Otherwise, the responder shall reject the first message with status code set to STATUS\_INVALID\_PAIRWISE\_CIPHER.
* If the AKM is not AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, validate that the finite cyclic group indicated in the Diffie-Hellman/ML-KEM Parameter element in the first Authentication frame is supported (present in dot11RSNAConfigDLCGroupTable). Otherwise, the responder shall reject the first message with status code set to UNSUPPORTED\_FINITE\_CYCLIC\_GROUP.
* If the AKM is AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, validate that the Group/ML-KEM field of the Diffie-Hellman/ML-KEM Parameter element in the first Authentication frame indicates ML-KEM-1024. Otherwise, the responder shall reject the first message with status code set to UNSUPPORTED\_ML\_KEM\_PARAMETER.
* If the AKM is not AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, verify the public key indicated in the Diffie-Hellman/ML-KEM Parameter element in the first message as specified in 5.6.2.3 of NIST SP 800-56A R2. If verification fails, the responder shall reject the first Authentication frame with status code set to INVALID\_PUBLIC\_KEY.
* If the AKM is AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, validate that the encapsulation key indicated in the Diffie-Hellman/ML-KEM Parameter element in the first message according to encapsulation key check as defined in 7.2 (ML-KEM Encapsulation) in NIST FIPS 203. If verification fails, the responder shall reject the first Authentication frame with status code set to INVALID\_ML\_KEM\_PARAMETER.
* Verify that a PMKSA named via a PMKID in the RSNE exists for the specified AKM in the RSNE if one or more PMKIDs are included.
* If a PMKSA is identified, the responder shall use PMKSA caching, shall not process the EAPOL PDU in the first Authentication frame, and shall not include the EAPOL PDU in the second authentication frame.
* If no PMKSA is identified, continue the IEEE 802.1X authentication.

If the first Authentication frame is not rejected, the responder shall:

* If the AKM is not AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, store the indicated SNonce and generate an ephemeral (random) private key with the chosen finite cyclic group and use the selected group’s scalar operation with the private key to generate its ephemeral public key.
* If the AKM is not AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, perform the group's scalar-op (see 12.4.4.1 (General)) with the originator’s ephemeral public key and its own ephemeral private key to produce an ephemeral Diffie-Hellman shared secret, DHss.
* If the AKM is AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, store the indicated SNonce and using the ephemeral encapsulation key and the ML-KEM encapsulation algorithm to generate the ephemeral ciphertext and the ephemeral ML-KEM shared secret, MLKEMss
* Use PMKSA caching if a PMKSA is identified via a PMKID in the RSNE in the first Authentication frame and before sending the second Authentication frame:
* Derive PTK with the identified PMKSA and DHss/MLKEMss as defined in 12.7.1.3 (Pairwise key hierarchy).
* Irretrievably delete the shared secret, DHss/MLKEMss, upon completion of PTK generation.

The responder shall do the following in the second Authentication frame:

* Include an RSNE to indicate the AKM and pairwise cipher indicated in the first Authentication frame.
* If a PMKSA is identified via a PMKID in the RSNE in the first Authentication frame, the responder shall include the PMKID corresponding to the PMKSA in the RSNE.
* Otherwise, the responder shall not include any PMKID in the RSNE.
* Not include an AKM Suite Selector element.
* Include a Diffie-Hellman/ML-KEM Parameter element.
* If the AKM is not AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, indicate the chosen finite cyclic group in the Diffie-Hellman/ML-KEM Parameter element, which is the same as the finite cyclic group in the Diffie-Hellman/ML-KEM Parameter element of the first Authentication frame.
* If the AKM is AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, indicate ML-KEM-1024 in the Group/ML-KEM field of the Diffie-Hellman/ML-KEM Parameter element.
* If the AKM is not AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, indicate its ephemeral public key in the Diffie-Hellman/ML-KEM Parameter element.
* If the AKM is AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, indicate the ephemeral ciphertext in the Diffie-Hellman/ML-KEM Parameter element.
* Include a Nonce element to indicate ANonce.

Otherwise, a responder that sets dot11EPPReAssociationFrameEncryptionSupportActivated to false or does not receive the RSNXE in the first Authentication frame with the (Re)Association Frame Encryption Support field set to 1 shall not include a Diffie-Hellman/ML-KEM Parameter element nor a Nonce element nor an RSNE in the second Authentication frame for IEEE 802.1X authentication.

After receiving the second Authentication frame with the status code set to SUCCESS, an originator shall:

* Validate that there is a Diffie-Hellman/ML-KEM Parameter element and an RSNE included in the second Authentication frame and there is no AKM Suite Selector element in the second Authentication frame if the originator included a Diffie-Hellman/ML-KEM Parameter element in the first Authentication frame. If the validation fails, the originator shall discard the frame and terminate further protocol processing.
* Validate that there is no Diffie-Hellman/ML-KEM Parameter element and no RSNE included in the second Authentication frame if the originator did not include a Diffie-Hellman/ML-KEM Parameter element in the first Authentication frame. If the validation fails, the originator shall discard the frame and terminate further protocol processing.
* Validate that the Group/ML-KEM field indication in the Diffie-Hellman/ML-KEM Parameter element in the second Authentication frame is the same as the Group/ML-KEM field indication in the Diffie-Hellman/ML-KEM Parameter element in the first Authentication frame if the originator included a Diffie-Hellman/ML-KEM Parameter element in the first Authentication frame. Validate that the pairwise cipher suite and the AKM indicated in the second Authentication frame are the same as the pairwise cipher suite and the AKM indicated in the first Authentication frame if the originator includes a Diffie-Hellman/ML-KEM Parameter element in the first Authentication frame. If the validation fails, the originator shall discard the frame and terminate further protocol processing.
* If the AKM is not AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, verify the public key indicated in the Diffie-Hellman/ML-KEM Parameter element in the second Authentication frame as specified in 5.6.2.3 of NIST SP 800-56A R2. If verification fails, the originator shall discard the frame and terminate further protocol processing.
* If the AKM is AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, verify that the ephemeral ciphertext indicated in the Diffie-Hellman/ML-KEM Parameter element in the second Authentication frame according to decapsulation input check as defined in 7.3 (ML-KEM Decapsulation) in NIST FIPS 203. If verification fails, the originator shall discard the frame and terminate further protocol processing.
* Validate that the Encapsulation Length field is set to 0 and validate that the PMKID included in the second Authentication frame matches one of the PMKID(s) indicated in the first Authentication frame if the originator includes one or more PMKIDs in the first Authentication frame, and the second Authentication frame includes a PMKID. If verification succeeds, the originator shall use PMKSA caching with the PMKSA identified by the PMKID indicated in the second Authentication frame and shall not continue the IEEE 802.1X Authentication frame exchange. If verification fails, the originator shall discard the frame and terminate further protocol processing.
* Validate that there is no PMKID included in the second Authentication frame if the originator does not include any PMKID in the first Authentication frame. If verification fails, the originator shall discard the frame and terminate further protocol processing.
* Store the indicated ANonce
* If the AKM is not AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, perform the group's scalar-op (see 12.4.4.1 (General)) with the originator’s ephemeral public key and its own ephemeral private key to produce an ephemeral Diffie-Hellman shared secret, DHss, if the second Authentication frame is not discarded.
* If the AKM is AKM 00-0F-AC:<ANA#1> or AKM 00-0F-AC:<ANA#2>, using the originator’s ephemeral ciphertext, its own ephemeral decapsulation key and the ML-KEM decapsulation algorithm to generate the ephemeral ML-KEM shared secret, MLKEMss, if the second Authentication frame is not discarded.
* Derive the PTK with the identified PMKSA and DHss/MLKEMss as defined in 12.7.1.3 (Pairwise key hierarchy) if a PMKSA is identified. Irretrievably delete the shared secret, DHss/MLKEMss, upon completion of PTK generation.

If a PMKSA is not identified through PMKSA caching, before sending the Authentication frame carrying EAP Success, a responder shall:

* Derive the PTK with DHss/MLKEMss as defined in 12.7.1.3 (Pairwise key hierarchy).
* Irretrievably delete the shared secret, DHss/MLKEMss, upon completion of PTK generation.

If a PMKSA is not identified throughPMKSA caching, after receiving the Authentication frame carrying EAP Success, an originator shall:

* Derive the PTK with DHss/MLKEMss as defined in 12.7.1.3 (Pairwise key hierarchy).
* Irretrievably delete the shared secret, DHss/MLKEMss, upon completion of PTK generation.

**TGbt Editor: *Instruction: Modify 13.2.2 as shown below***

**13.2.2 Authenticator key holders**

(…existing texts…)

The R0KH derives the PMK-R0 for use in the mobility domain utilizing the MSK (when the negotiated AKM is 00-0F-AC:3, 00-0F-AC:13, 00-0F-AC:<ANA#2>), the PSK (when the negotiated AKM is 00-0F-AC:4) or the PMK (when the negotiated AKM is 00-0F-AC:9 or 00-0F-AC:25), or the FILS-FT (when the negotiated AKM is 00-0F-AC:16 or 00-0F-AC:17). The R0KH shall be responsible for deriving a PMK-R1 for each R1KH within the mobility domain.

(…existing texts…)

**TGbt Editor: *Instruction: Modify 13.2.3 as shown below***

**13.2.3 Supplicant key holders**

(…existing texts…)

The S0KH derives the PMK-R0 for use in the mobility domain utilizing the MSK (when the negotiated AKM is 00-0F-AC:3, 00-0F-AC:13, 00-0F-AC:<ANA#2>), the PSK (when the negotiated AKM is 00-0F-AC:4) or the PMK (when the negotiated AKM is 00-0F-AC:9 or 00-0F-AC:25), or the FILS-FT (when the negotiated AKM is 00-0F-AC:16 or 00-0F-AC:17).

(…existing texts…)

**TGbt Editor: *Instruction: Modify 13.8.4 as shown below***

**13.8.4 FT authentication sequence: contents of third message**

(…existing texts…)

If present, the FTE shall be set as follows:

….

* When the negotiated AKM is 00-0F-AC:13 or 00-0F-AC:<ANA#2>, the MIC shall be calculated using the PTK-KCK and the HMAC-SHA-384 algorithm. The output of the HMAC-SHA-384 shall be truncated to 192 bits.

…..

(…existing texts…)

**TGbt Editor: *Instruction: Modify 13.8.4 as shown below***

**13.8.5 FT authentication sequence: contents of fourth message**

(…existing texts…)

If present, the FTE shall be set as follows:

* When the negotiated AKM is 00-0F-AC:13 or 00-0F-AC:<ANA#2>, the MIC shall be calculated using the PTK-KCK and the HMAC-SHA-384 algorithm. The output of the HMAC-SHA-384 shall be truncated to 192 bits.

(…existing texts…)