IEEE P802.11
Wireless LANs

|  |
| --- |
| AKM for SHA-384 |
| Date: 2021-06-09 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Po-Kai Huang | Intel |  |  |  |
| Ido Ouzieli |  |  |  |
| Ehud Reshef |  |  |  |
| Daniel F Bravo |  |  |  |
| Stacey Robert |  |  |  |
| Emily Qi |  |  |  |

Abstract

This document proposes two new AKMs to allow mixed key length for individual addressed data and group addressed data and enable SHA-384 under “for FT authentication negotiated over IEEE Std 802.1X” and “Authentication negotiated over IEEE Std 802.1X” scenarios without the need to comply with CNSA suite requirement like AKM 12 and AKM 13.

Discussion:

In the current AKM suite selectors, AKM 8 (SAE), AKM 9 (SAE under FT), AKM 19 (PSK under FT), AKM 20 (PSK) enable SHA-384 and allow GCMP-256 for pairwise cipher and CCMP-128 for group data cipher. For Authentication negotiated over IEEE Std 802.1X and FT authentication negotiated over IEEE Std 802.1X, the current spec only has AKM 12 and AKM 13 to enable SHA-384, but it requires usage of 256 bit key length across the board to comply with CNSA suite requirement.

We think two new AKM options for Authentication negotiated over IEEE Std 802.1X and FT authentication negotiated over IEEE Std 802.1X using SHA-384 and without the requirement of CNSA suite to have 256 bit key length across the board should be added to complete the picture.

While providing the texts for the two new AKMs, we also notice that AKM 19 do not have corresponding change in Table 12-10 and PRF definiotns, so bug fixes for AKM 19 are proposed as well. AKM 20 do not have corresponding change in Table 12-10, but has PRF definition fix in 11-21-716r2. Hence, bug fixes for AKM 20 are proposed as well.

*TGme editor: Change Clause 9.4.2.24.3* *as follows (track change on):*

* AKM suites

The AKM Suite Count field indicates the number of AKM suite selectors that are contained in the AKM Suite List field. The value 0 is reserved.

The AKM Suite List field contains a series of AKM suite selectors. In an IBSS only a single AKM suite selector is specified because IBSS STAs use the same AKM suite and because there is no mechanism to negotiate the AKMP in an IBSS (see 12.6.5 (RSNA policy selection in an IBSS)).

Each AKM suite selector specifies an AKMP. Table 9-151 (AKM suite selectors) gives the AKM suite selectors defined by this -standard. An AKM suite selector has the format shown in Figure 9-288 (Suite selector format).

|  |
| --- |
| * AKM suite selectors
 |
| OUI | Suite type | Meaning | Authentication algorithm numbers (see 9.4.1.1 (Authentication Algorithm Number field)) |
| Authentication type | Key management type | Key derivation type  |
| 00-0F-AC | 0 | Reserved | Reserved | Reserved | Reserved |
| 00-0F-AC | 1 | Authentication negotiated over IEEE Std 802.1X | RSNA key management as defined in 12.7 (Keys and key distribution) | Defined in 12.7.1.2 (PRF) | 0 (open) |
| 00-0F-AC | 2 | PSK | RSNA key management as defined in 12.7 (Keys and key distribution) | Defined in 12.7.1.2 (PRF) | 0 (open) |
| 00-0F-AC  | 3 | 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-256 | 2 (FT) for FT protocol reassociation as defined in 13.5 (FT protocol)0 (open) for FT Initial Mobility Domain Association over IEEE Std 802.1X or PMKSA caching |
| 00-0F-AC  | 4 | FT authentication using PSK | 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-256 | 2 (FT) for FT protocol reassociation as defined in 13.5 (FT protocol)0 (open) for FT Initial Mobility Domain Association using PSK |
| 00-0F-AC  | 5 | Authentication negotiated over IEEE Std 802.1X | 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-256 | 0 (open) |
| 00-0F-AC  | 6 | PSK | 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-256 | 0 (open) |
| 00-0F-AC | 7 | TDLS | TPK handshake | Defined in 12.7.1.6.2 (Key derivation function (KDF)) using SHA-256 | N/A |
| 00-0F-AC | 8 | SAE authentication  | RSNA key management as defined in 12.7 (Keys and key distribution), or authenticated mesh peering exchange as defined in 14.5 (Authenticated mesh peering exchange (AMPE)) | Defined in 12.7.1.6.2 (Key derivation function (KDF)) using the hash algorithm specified in 12.4.2 (Assumptions on SAE)  | 3 (SAE) for SAE Authentication0 (open) for PMKSA caching |
| 00-0F-AC | 9 | FT authentication over SAE | FT key management defined in 12.7.1.6 (FT key hierarchy) | Defined in 12.7.1.6.2 (Key derivation function (KDF)) using the hash algorithm specified in 12.4.2 (Assumptions on SAE)  | 3 (SAE) for FT Initial Mobility Domain Association2 (FT) for FT protocol reassociation as defined in 13.5 (FT protocol) 0 (open) for FT Initial Mobility Domain Association over PMKSA caching |
| 00-0F-AC | 10 | APPeerKey Authentication with SHA-256 | 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-256 | N/A |
| 00-0F-AC | 11 | Authentication negotiated over IEEE Std 802.1X using a Suite B compliant EAP method supporting SHA-256 | 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-256 | 0 (open) |
| 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) |
| 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) for FT Initial Mobility Domain Association over IEEE Std 802.1X or PMKSA caching |
| 00-0F-AC | 14 | Key management over FILS using SHA-256 and AES-SIV-256, or authentication negotiated over IEEE Std 802.1X | FILS key management defined in 12.11.2.5 (Key establishment with FILS authentication) | Defined in 12.11.2.5 (Key establishment with FILS authentication) using SHA-256 | 4, 5 or 6 (FILS) for FILS Authentication0 (open) for IEEE Std 802.1X |
| 00-0F-AC | 15 | Key management over FILS using SHA-384 and AES-SIV-512, or authentication negotiated over IEEE Std 802.1X | FILS key management defined in 12.11.2.5 (Key establishment with FILS authentication) | Defined in 12.11.2.5 (Key establishment with FILS authentication) using SHA-384 | 4, 5 or 6 (FILS) for FILS Authentication0 (open) for IEEE Std 802.1X |
| 00-0F-AC | 16 | FT authentication over FILS with SHA-256 and AES-SIV-256 or 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-256 | 4, 5 or 6 (FILS) for FT Initial Mobility Domain Association over FILS2 (FT) for FT protocol reassociation as defined in 13.5 (FT protocol)0 (open) for FT Initial Mobility Domain Association over IEEE Std 802.1X or PMKSA caching |
| 00-0F-AC | 17 | FT authentication over FILS with SHA-384 and AES-SIV-512, or 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  | 4, 5 or 6 (FILS) for FT Initial Mobility Domain Association over FILS2 (FT) for FT protocol reassociation as defined in 13.5 (FT protocol)0 (open) for FT Initial Mobility Domain Association over IEEE Std 802.1X or PMKSA caching |
| 00-0F-AC | 18 | Reserved | Reserved | Reserved | Reserved |
| 00-0F-AC | 19 | FT authentication using PSK | 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) for FT Initial Mobility Domain Association using PSK |
| 00-0F-AC | 20 | PSK | 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) |
| 00-0F-AC  | 21 | 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) for FT Initial Mobility Domain Association over IEEE Std 802.1X or PMKSA caching |
| 00-0F-AC  | 22 | Authentication negotiated over IEEE Std 802.1X | 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) |
| 00-0F-AC |  23–255  | Reserved | Reserved | Reserved | Reserved |
| Other OUI or CID | Any | Vendor-specific | Vendor-specific | Vendor-specific | Vendor-specific |

NOTE 1—The selector value 00-0F-AC:1 specifies only that IEEE Std 802.1X-2010 is used as the authentication transport. IEEE Std 802.1X-2010 selects the authentication mechanism.

The AKM suite selector value 00-0F-AC:8 (i.e., SAE authentication with SHA-256 is used when either a password or PSK is used with RSNA key management.

NOTE 2—Selector values 00-0F-AC:1 and 00-0F-AC:8 can simultaneously be enabled by an Authenticator.

The AKM suite selector value 00-0F-AC:2 (PSK) is used when an alternate form of PSK is used with RSNA key management.

NOTE 3—Selector values 00-0F-AC:1 and 00-0F-AC:2 can simultaneously be enabled by an Authenticator.

The AKM suite selector value 00-0F-AC:11 is used only with cipher suite selector values 00-0F-AC:8 (GCMP-128) and 00-0F-AC:11 (BIP-GMAC-128). The AKM suite selector value 00-0F-AC:12 is 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). The AKM suite selector value 00-0F-AC:13 is 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).

NOTE 4—The AKM suite selector value 00-0F-AC:11 is deprecated.

NOTE 5—The usage of selector values with authentication algorithms is defined in the Authentication algorithm numbers column of Table 9-151 (AKM suite selectors); see 9.4.1.1 (Authentication Algorithm Number field).

The AKM suite selector values 00-0F-AC:19, 00-0F-AC:20, 00-0F-AC:21 and 00-0F-AC:22 are used only with pairwise cipher suite selector values 00-0F-AC:9 (GCMP-256), and 00-0F-AC:10 (CCMP-256).

A PMKSA established using a given AKM selector value may be cached and used in a subsequent (re)association as defined in 12.6.10.3 (Cached PMKSAs and RSNA key management).

* PRF

***TGbe editor: Insert a new pagagraph after the seventh pagraph (When the negotiated AKM is 00-0F-AC:15 or 00-0F-AC:17, the KDF specified in 12.7.1.6.2 (Key derivation function (KDF)) shall be used instead of the PRF construction defined here. …) as follows (track change on):***

When the negotiated AKM is 00-0F-AC:15 or 00-0F-AC:17, 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-640(K, A, B) = KDF-SHA-384-640(K, A, B)

PRF-768(K, A, B) = KDF-SHA-384-768(K, A, B)

PRF-1024(K, A, B) = KDF-SHA-384-1024(K, A, B)

PRF-1152(K, A, B) = KDF-SHA-384-1152(K, A, B)

PRF-1408(K, A, B) = KDF-SHA-384-1408(K, A, B)

PRF-1536(K, A, B) = KDF-SHA-384-1536(K, A, B)

When the negotiated AKM is 00-0F-AC:19 or 00-0F-AC:21 or 00-0F-AC:22, 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-length(K, A, B) = KDF-SHA-384-length(K, A, B)

* Pairwise key hierarchy

*TGbe editor: Modify the first pagraph as follows (track change on):*

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-
30 (Pairwise key hierarchy). When using AKM suite selector 00-0F-AC:12 or 00-0F-AC:15 or 00-0F-AC:22, the length of the PMK, PMK\_bits, shall be 384 bits. When using AKM suite selectors for which the Authentication type column indicates FT authentication (see Table 9-151 (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. The PTK is partitioned into KCK, KEK, and a temporal key, which is used by the MAC
to protect individually addressed communication between the Authenticator’s and Supplicant’s respective
STAs. PTKs are used between a single Supplicant and a single Authenticator.

*TGbe editor: Modify the twelveth pagraph (When the negotiated AKM is 00-0F-AC:20, the PMK identifier is defined as…) as follows (track change on):*

When the negotiated AKM is 00-0F-AC:20 or 00-0F-AC:22, the PMK identifier is defined as

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

* PMK-R0

***TGbe editor: Insert a new pagagraph after the ninth pagraph (If the negotiated AKM is 00-0F-AC:19, then Q = 384 and…) as follows (track change on):***

If the negotiated AKM is 00-0F-AC:19, then Q = 384 and

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

If the negotiated AKM is 00-0F-AC:21, then Q = 384 and

* MPMK = L(MSK, 0, 384), i.e., the first 384 bits of the MSK (which is derived from the IEEE
802.1X authentication)
* PMKID = Truncate-128(HMAC-SHA-384(MPMK, “PMK Name” || AA || SPA))
* EAPOL-Key frame construction and processing

***TGbe editor: change table 12-10 as follows***

|  |
| --- |
| * Integrity and key wrap algorithms
 |
| AKM | Integrity algorithm | KCK\_bits | Size of MIC | Key wrap algorithm | KEK\_bits | KCK2\_bits | KEK2\_bits |
| Deprecated | HMAC-MD5 | 128 | 16 | ARC4 | 128 | 0 | 0 |
| 00-0F-AC:1 | HMAC-SHA-1-128 | 128 | 16 | NIST AES Key Wrap | 128 | 0 | 0 |
| 00-0F-AC:2  | HMAC-SHA-1-128 | 128 | 16 | NIST AES Key Wrap | 128 | 0 | 0 |
| 00-0F-AC:3 | AES-128-CMAC | 128 | 16 | NIST AES Key Wrap | 128 | 0 | 0 |
| 00-0F-AC:4 | AES-128-CMAC | 128 | 16 | NIST AES Key Wrap | 128 | 0 | 0 |
| 00-0F-AC:5 | AES-128-CMAC | 128 | 16 | NIST AES Key Wrap | 128 | 0 | 0 |
| 00-0F-AC:6 | AES-128-CMAC | 128 | 16 | NIST AES Key Wrap | 128 | 0 | 0 |
| 00-0F-AC:8 | AES-128-CMAC | 128 | 16 | NIST AES Key Wrap | 128 | 0 | 0 |
| 00-0F-AC:9 | AES-128-CMAC | 128 | 16 | NIST AES Key Wrap | 128 | 0 | 0 |
| 00-0F-AC:11 | HMAC-SHA-256 | 128 | 16 | NIST AES Key Wrap | 128 | 0 | 0 |
| 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:14 | AES-SIV-256 | 0 | 0 | AES-SIV-256 | 256 | 0 | 0 |
| 00-0F-AC:15 | AES-SIV-512 | 0 | 0 | AES-SIV-512 | 512 | 0 | 0 |
| 00-0F-AC:16 | AES-SIV-256/ AES-128-CMAC | 0 | 0/16 | AES-SIV-256/ NIST AES Key Wrap | 256 | 128 | 128 |
| 00-0F-AC:17 | AES-SIV-512/ HMAC-SHA-384 | 0 | 0/24 | AES-SIV-512/ NIST AES Key Wrap | 512 | 192 | 256 |
| 00-0F-AC:19 | HMAC-SHA-384 | 192 | 24 | NIST AES Key Wrap | 256 | 0 | 0 |
| 00-0F-AC:20 | HMAC-SHA-384 | 192 | 24 | NIST AES Key Wrap | 256 | 0 | 0 |
| 00-0F-AC:21 | HMAC-SHA-384 | 192 | 24 | NIST AES Key Wrap | 256 | 0 | 0 |
| 00-0F-AC:22 | HMAC-SHA-384 | 192 | 24 | NIST AES Key Wrap | 256 | 0 | 0 |

**References:**