IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resolution of CIDs for section 11.11.2.3 | | | | |
| Date: 2014-05-15 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Dan Harkins | Aruba Networks | 1322 Crossman ave, Sunnyvale, CA | +1 408 227 4500 | dharkins at aruba networks dot com |
|  |  |  |  |  |

Abstract

This submission proposes resolution to comments related to the key derivation setion, 11.11.2.3. Namely: CIDs 4083, 4292, 4329, 4331, 4893, 4950, 5075, and 5076.

***Instruct editor to modify section 11.11.2.3 as indicated:***

**11.11.2.3 Key derivation with FILS authentication**

Key derivation with FILS Authentication uses the KDF from 11.6.1.7.2 (Key derivation function (KDF)) to keys for a PMKSA a Pairwise Master Key (PMK) and a PTKSA-a key confirmation key (KCK), a key encryption key (KEK), and a temporal key (TK). In both cases, when the AKM used is 00-0F-AC:<ANA-1> the hash algorithm used for the KDF shall be SHA256 and when the AKM used is 00-0F-AC:<ANA-2> the hash algorithm used for the KDF shall be SHA384.

When using PMKSA caching, a new PMKSA is not created. Instead, the PMKSA used for PMKSA caching remains and continues to be identified by the appropriate PMKID. Regardless of whether PMKSA caching is used or not, a PTKSA shall be generated with each FILS authentication exchange.

**11.11.2.3.1 PMK key derivation with FILS authentication**

For PMKSA generation, the inputs to the KDF are a concatentation of the two nonces, NSTA and NAP, in that order, a constant label, the EAP-RP secret result if shared key authentication is being used, and, the Diffie-Hellman shared secret, ss, if PFS is being used or public key authentication is being used. The KDF produces a PMK and a PMKID which is used to uniquely identify the PMKSA. The length of the PMK shall be 256 bits, and the length of the PMKID shall be 128 bits:

PMKID | PMK = KDF-384(NSTA | NAP, "FILS PMKSA Derivation", [rMSK |][ss])

Where:

* Brackets indicate optional data
* <zero> is a string of zeros of length 256 or a length of 384, depending on the AKM used
* rMSK is the output of the EAP-RP exchange if shared key authentication was used
* ss is the result of the Diffie-Hellman exchange if public key authentication was used or if PFS was used with shared key authentication

Upon completion of PMK generation the shared secret, ss, and rMSK, if applicable, shall be irretrievably destroyed.

**11.11.2.3.2 PTK key derivation with FILS authentication**

For PTKSA key generation, the inputs to the KDF are the two 16 octet nonces NSTA and NAP produced by the STA and AP, respectively, a constant label, and the PMK of the PMKSA. When the AKM used is 00-0F-AC:<ANA-1>, the length of KEK shall be 128 bits, and the length of the KCK 256 bits. When the AKM used is 00-0F-AC:<ANA-2> the length of the KEK shall be 256 bits, and the length of KCK shall be 384 bits, The total amount of bits extracted from the KDF shall therefore be 384+TK or 640+TK bits depending on the AKM used, where TK\_bits is determined from Table 11-4 (Cipher suite key lengths).

KCK | KEK | TK = KDF-X(PMK, “FILS PTKSA Derivation”, SPA | AA)

Where:

— X is 384+TK\_bits or 640+TK bits from Table 11-4 (Cipher suite key lengths) depending on the AKM used

* PMK is the PMK from the PMKSA, either created from an initial FILS connection or from a cached PMKSA, when PMKSA caching is used.
* SPA is the STA’s MAC address and AA is the AP’s BSSID.

If the negotiated AKM is 00-0F-AC-<ANA-1> or 00-0F-AC-<ANA-2>, FILS requires an additional element: a 13 octet AEAD counter to be part of the newly created PTKSA. The STA shall set the AEAD counter to 13 octets of zero and the AP shall set the first octet to the value 128 and the remaining octets to zero (i.e. the first bit of the AEAD counter is 1 and the rest of the bits in the counter are 0). To allow for proper processing, each side shall include the AEAD counter of the other as a peer's AEAD counter (see 11.11.2.5 (AEAD cipher mode)). AEAD counters are processed per 11.11.2.5 (AEAD cipiher mode for FILS).

**References:**