IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resolution of CIDs for section 11.11.2.3 | | | | |
| Date: 2014-06-02 | | | | |
| Author: | | | | |
| Name | Affiliation | Address | Phone | email |
| Mark RISON | Samsung Cambridge Solution Centre | SJH, CB4 0DS, U.K. | +44 1223 434600 | at samsung (a global commercial entity) I'm the letter emme then dot rison |
| Original author (r0/r1): |  |  |  |  |
| 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 resolutions to comments related to the key derivation subclause, 11.11.2.3. Namely: CIDs 4082, 4083, 4292, 4329, 4330, 4331, 4391, 4392, 4393, 4776, 4893, 4950, 5042, 5075, and 5076, plus some small editorials. It is derived from 14/0692r1 from Dan Harkins (Aruba Networks).

***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 derive keys for:

* a PMKSA (a pairwise master key (PMK) and the corresponding PMKID)
* 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, but regardless of whether PMKSA caching is used or not, a new PTKSA shall be generated with each FILS authentication exchange.

**11.11.2.3.1 PMKSA key derivation with FILS authentication**

For PMKSA key generation, the inputs to the KDF are:

* the STA and AP nonces, SNonce and ANonce
* a constant label
* the EAP-RP secret result, rMSK, if shared key authentication is being used
* 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 corresponding PMKID. The length of the PMK shall be 256 bits, and the length of the PMKID shall be 128 bits:

PMKID || PMK = KDF-384(SNonce || ANonce, "FILS PMKSA Derivation", context)

where the context is:

* rMSK if shared key authentication is being used without PFS
* rMSK || ss if shared key authentication is being used with PFS
* ss if public key authentication is being used

Upon completion of PMK generation, ss and rMSK, if generated, shall be irretrievably destroyed.

**11.11.2.3.2 PTKSA key derivation with FILS authentication**

For PTKSA key generation, the inputs to the KDF are:

* the PMK of the PMKSA, either created from an initial FILS connection or from a cached PMKSA
* a constant label
* the STA’s MAC address, SPA
* the AP’s BSSID, AA
* the STA and AP nonces, SNonce and ANonce

The KDF produces a KCK, a KEK and a TK. When the AKM used is 00-0F-AC:<ANA-1>, the length of the KEK shall be 128 bits, and the length of the KCK shall be 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, X, shall therefore be 384+TK\_bits 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 || SNonce || ANonce)

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:**

IEEE P802.11ai/D2.0

IEEE P802.11-REVmc/D2.0