IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resolution of Some Section 11 CIDs | | | | |
| Date: 2014-12-29 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Dan Harkins | Aruba Networks | 1322 Crossman avenue, Sunnyvale, California, United States of America | +1 408 227 4500 | dharkins at aruba networks dot com |
|  |  |  |  |  |

Abstract

This submission proposes resolution to CIDs 6026, 6034, 6035, 6036, 6071, 6075, 6476, 6801, and 6803

***Instruct the editor to modify section 11.11.1 as indicated:***

**11.11.1 Assumptions on FILS authentication**

The security of FILS authentication depends on the following assumptions:

* If a TTP is used, communication between each STA and the TTP is protected with a secure deterministic authenticated encryption function.
* When shared key authentication is used, each STA shares either a valid rRK as defined in IETF RFC 6696 with a TTP that is capable of being used with EAP-RP, or a PMK cached from a previous FILS authenticated connection.

***Instruct the editor to modify section 11.11.2.2.2 as indicated:***

**11.11.2.2.2 Key establishment with FILS public key authentication**

FILS public key authentication performs key establishment with a Diffie-Hellman exchange. Prior to beginning the exchange, the non-AP STA selects a finite cyclic group from the dot11RSNConfigDLCGroup table in which to perform the Diffie-Hellman exchange. It then generates a random nonce, generates an ephemeral private key, and uses the selected group's scalar-op (see 11.3.4.1 (General)) with its private key to generate its ephemeral public key. It then constructs an Authentication frame (see 8.3.3.11 (Authentication frame format)) as follows:

1. The Authentication algorithm number shall be set to <ANA-1> and the Authentication transaction sequence number shall be set to one (1).
2. The random nonce shall be encoded in the FILS nonce field (see 8.4.1.59 (FILS Nonce field))
3. The FILS authentication type shall be set to indicate FILS public key authentication (2)
4. The chosen finite cyclic group shall be encoded in the Finite Cyclic Group field (see 8.4.1.42 (Finite Cyclic Group field))
5. The STA's public key shall be encoded into the Element field (see 8.4.1.40 (Element field)) according to the element to octet-string conversion in 11.3.7.2.4 (Element to octet string conversion).

The STA shall then transmit the Authentication frame to the AP.

Upon receipt, the AP processes the STA's Authentication frame as follows:

1. If the finite cyclic group indicated by the Finite Cyclic Group field is not acceptable, the AP shall respond with an Authentication frame with the status code of 77 (“Authentication is rejected because the offered finite cyclic group is not supported”) and terminate the FILS authentication protocol.
2. If the finite cyclic group is acceptable, the AP shall verify the validity of the STA's public key:
3. The public key shall be converted from an octet string to an element according to the conversion in 11.3.7.2.5 (Octet string to element conversion).
4. The public key, as a group element, shall be verified in a group-specific fashion as described in 5.6.2.3 of NIST SP 800-56a-2013. If verification fails, the AP shall terminate the FILS authentication protocol.

3. The STA’s nonce and validated public key shall be extracted from the Authentication frame.

Next, the AP then shall generate a random nonce, generate a random, ephemeral private key, and then use the agreed-upon group's scalar-op (see 11.3.4.1 (General)) with its private key to generate its ephemeral public key. The AP then constructs an Authentication frame (see 8.3.3.11 (Authentication frame format)) as follows:

1. The Authentication algorithm number set to <ANA-1>, and the Authentication transaction sequence number set to two (2)
2. The FILS authentication type to indicate FILS public key authentication (2)
3. The random nonce shall be encoded in the FILS nonce field (see 8.4.1.59 (FILS Nonce field))
4. The finite cyclic group shall be encoded in the Finite Cyclic Group field (see 8.4.1.42 (Finite Cyclic Group field))
5. The AP's public key shall be encoded in the Element field (see 8.4.1.40 (Element field)) according to the element to octet-string conversion in 11.3.7.2.4 (Element to octet string conversion).

The AP shall then transmit the Authentication frame to the STA. The AP shall then compute the Diffie-Hellman shared secret, ss, based on the STA's ephemeral public key and its own private key with the chosen group's scalar-op, and the AP shall then perform Key Derivation (see 11.11.2.3 (Key derivation with FILS authentication)).

Upon receipt, the STA processes the AP's Authentication frame as follows:

1. It verifies that the finite cyclic group in the AP's response is equal to the group selected by the STA. If these differ, the STA shall terminate the authentication exchange.
2. The STA shall verify the validity of the AP's public key.
3. he public key shall be converted from an octet string to an element according to the conversion in 11.3.7.2.5 (Octet string to element conversion)
4. The public key, as a group element, shall be verified in a group-specific fashion according to 5.6.2.3 of NIST SP 800-56a-2013. If public key validation fails the STA shall terminate the authentication exchange.

3. The STA shall extract the AP’s nonce and verified public key from the Authentication frame.

Next, the STA shall compute the Diffie-Hellman shared secret, ss, based on the AP's ephemeral public key and its own private key with the chosen group's scalar-op to derive ss. The STA then performs Key Derivation (see 11.11.2.3 (Key derivation with FILS authentication)) and begins Key Confirmation (see 11.11.2.4 (Key confirmation with FILS authentication)).

***Instrut the editor to modify section 11.11.2.3.1 as indicated:***

**11.11.2.3.1 PMKSA key derivation with FILS authentication**

The Extract function used to derive the PMK takes the two nonces as salt and the secret(s) from FILS Key establishment as input keying material. A PMKID used to identify the PMKSA is generated using the hash algorithm from the negotiated AKM on input data specific to the FILS Key Establishment step. The length of the PMK shall be either 256 bits or 382 bits depending on the negotiated AKM, and the length of the PMKID shall be 128 bits. If FILS shared key authentication was used to generate input keying material, the PMK and PMKID are derived as:

PMK = HMAC-Hash(SNonce || ANonce, rMSK [|| ss])

PMKID = Truncate-128(Hash(EAP-Initiate/Reauth))

And when FILS public key authentication is used to generate input keying material, the PMK and PMKID are derived as:

PMK = HMAC-Hash(SNonce || ANonce, ss)

PMKID = Truncate-128(Hash(gSTA || gAP)

Where:

* SNonce is the STA nonce and ANonce is the AP nonce
* rMSK is the shared secret from the EAP-RP exchange
* ss is the shared secret derived from the Diffie-Hellman exchange, when performed
* EAP-Initiate/Reauthn is the EAP-RP packet sent by the STA during key establishment with FILS shared key authentication
* gSTA is the STA’s Diffie-Hellman value and gAP is the AP’s Diffie-Hellman value
* Hash is the AKM-specific hash function

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

irretrievably destroyed.

***Instruct the editor to modify section 11.11.2.3.2 as indicated:***

**11.11.2.3.2 PTK key derivation with FILS authentication**

For PTKSA key generation, the inputs to the KDF are the PMK of the PMKSA, a constant label, and a concatenation of the STA MAC address, the AP’s BSSID, the STA’s nonce, and the AP’s nonce. 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 PTK Derivation”, SPA || AA || ANonce || SNonce)

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 the AA is the AP’s BSSID.
* SNonce is the STA’s nonce and ANonce is the AP’s nonce.

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 for FILS)).

**References:**