IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resolution to Comments : CID 2493, 2204,2194,3086,3259,2805,2896,2897,2199,3088,3086,2195,3153,  3192,2495,2198,2497,2877,2876,3003,3155,3194,3089,2995,2201,  2997,  2990,2993,2996,2202,3154,3193,3002,2222,2205, 2197,3245,3087,3243,2991,2986,2987,3001,2994 | | | | |
| Date: 2013-12-03 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Rob Sun | Huawei Technology | Suite 400, 303 Terry Fox drive,  Kanata, On | +1 613 2781948 | Rob.sun@huawei.com |
|  |  |  |  |  |

Abstract

This document presents suggested proposal towards CID

2493, 2204,2194,3086,3259,2805,2896,2897,2876,2199,3088,3086,2195,3153,

3192,2495,2198,2497,2877,3003,3155,3194,3089,2995,2201,2997,2205,2197,3245,3087,3243,2991,2986,2987,3001,2994

2990,2993,2996,2202,3154,3193,3002,

***Modify the following definition into 10.3.1 as highlighted in red texts:***

* STA authentication and association

***Discussion:***

Clause 11.11.2.3 to 11.11.2.7 outlines the procedures and key confirmation after the FILS authentication. Lots of the CIDs are related to the confusion of how to define the FILS key hierarchy and the length of the keys should be depending on the cipher suites selected. Also some of comments are regarding the definition of the PMKIDs which needs to be generated for the purpose of PMK management. Some CIDs are towards the problems of AEAD scheme which currently utilizes the static nonce.

Notes to Editor: Adopt the following subclauses: 11.11.2.3 to 11.11.2.7

**11.11.2.3 Key derivation with FILS authentication**

**[Note: CID 2493, 2204, 2205,2194,3086,3259,2805, 2896, 2897,3243,2986]**

Key derivation with FILS Authentication generates 3keys KCK2, KEK2and a traffic key (TK) respectively. The FILS key hierarchy follows the key structure specified in subclause 11.6.1.3. When the AKM for FILS is selected (00-0F-AC: 11 and 00-0F-AC:12 in table 8-113), the length of the keys are 128. When the AKM for FILS is 00-0F-AC:13, the length of the keys is 256bits

When using the EAP-RP with TTP, the FILS PMK is derived from rMSK. The PMK shall be computed as the first 256bits(bits 0-255) of the r MSK: PMK 🡨 L(rMSK,0,256), at this truncated key derivation, the r MSK needs to be at least 256 bits.[Note: 2197,2987]

When the Shared Secret ss is used for deriving the FILS PMK, if the PFS is being used, the KDF function specified in 11.6.1.7.2.1 shall be employed to generate the FILS PMK. PMK 🡨 HKDF-256(ss, “PMK Generation from Shared Secret”, MAC\_a||MAC\_[0]32).

The FILS PMKSA is managed and distributed as the PMKSA (section 11.5.1.1.2)

Figure xx: FILS Key Hierarchy

The pairwise key hierarchy takes an PMK and generates an FILS PTK. Then the PTK is partitioned into KCK2, KEK2 for the FILS association frame encryption and authentication, TK for session encryption. The KCK2 and KEK2 are derived for the sole purpose of protecting the FILS association frame per FILS association, and both are temporary and shall only be used during the FILS authentication protocol. Upon completion of the key derivation computation, the shared secret ss and rMSK, as applicable shall be irretrievably destroyed.

The following applies and are depicted in Figure xx-xx FILS Key Hierarchy

The B denotes the bit length of the key which is either 128 bit or 256 bit depending on the negotiated FILS AKM cipher modes in table 8-101 (section 8.4.2.27.3) [Note CID: 2199,3088,3086]. When the AKM for FILS is selected (00-0F-AC: 11 and 00-0F-AC:12 in table 8-113), B is 128 bits. When the AKM for FILS is 00-0F-AC:13, the B is 256bits

1. sNonce is the random or pseudorandom value contributed by the STA.
2. aNonce is the random or pseudorandom value contributed by the AP.
3. MAC\_s is the MAC address from the STA
4. MAC\_a is the MAC address from the AP
5. PMK is derived from rMSK or Shared Secret ss

PTK 🡨KDF-X ( PMK, “FILS PTK Generation”, Min(MAC\_s, MAC\_a)||Max(MAC\_s, MAC\_a)

||Min(sNonce,aNonce)|| Max(sNonce, aNonce))

Where the size of X is depending on the FILS AKM cipher suites. The Min and Max operations for IEEE 802 addresses are with the address coverted to a positive integer treating the first transmitted octet as the most significant octet of the integer. The Min and Max operations for nonces are with the nonces treated as positive integers converted as specified in 8.2.2 (Conventions).

1. The KCK2 shall only be used with key confirmation (see 11.11.2.4) and shall be computed as the first B bits of the PTK

KCK2 <- L ( PTK,0, B)

The KCK2 is ONLY used for the FILS association request/response for data origin authenticity protection (Section 11.11.2.4)

1. The KEK2 shall be computed as the bits from B –> 2 \* B of the PTK

KEK2 <- L (PTK, B,B)

The KEK2 is shall only be used with the encrypt-and-authenticate (see 11.11.2.6) and decrypt-and-verify (see 11.11.2.7) functions



KCK2 and KEK2 are temporary and shall only be used during the FILS authentication protocol.[Note: CID 3245]

1. The Temporal Key (TK) shall be computed as bits 2\*B -> (B+TK\_bits) of the PTK.

TK <- L(PTK, 2\*B, TK\_bits)

The TK serves the same role as the RSNA Temporal Key for both RSNA Supplicant and Authenticator in the EAPOL-Key state machine.

1. A FILS PMK identifier is defined and computed depending on the FILS AKM is negotiated and by default is defined as [Note CID 2195,3153,3192,3001]

PMKID = HMAC-SHA-256(FMK, “FILS PMKID Generation”|| aNonce||MAC\_a||sNonce||MAC\_s)

The HMAC-SHA-X is either the first 128 bits or 256 bits of the HMAC-SHA-256 of its argument list depending on the negotiated FILS AKM.

***Editor note: Add the following subclause 11.6.1.7.2.1***

***11 .6.1.7.2.1 Key Derivation Function(KDF) for FILS Key derivation with Shared Secret [CID 2197,2987]***

The KDF function is a variant of KDF function( section 11.6.1.7.2) for the FILS key derivation with the ss (shared secret) as the root of the FILS PMK (section 11.11.2.3) and utilizes the “ Extraction-then-Expansion” Key derivation procedures in accordance with provisions of NIST SP 800-56a, SP 800-56c and RFC 5869.

**Output 🡨HKDF-Length (ss, label, Context, Salt) where**

Input:

*ss*, a variable length shared secret

*label*, a string identifying the purpose of the keys derived using this KDF

*Context*, a bit string that provides context to identify the derived key

*Length*, the length of the derived key in bits

*Salt: Salt is optional value to increase the strength of the key*

Output:

a *Length*-bit derived key

// Key Extraction

*if Salt is optional*

*Salt 🡨 Chr(0)\*Length*

**K 🡨** HMAC-SHA256(Salt, ss)

// Key Expansion

*result 🡨“”*

*iterations**Length*+255)/256

*t🡨 “ “*

**do** *i* = 1 **to** *iterations // Key Expansion to the Length*

*t 🡨 HMAC-SHA256(K, t||label||Context||Length)*

*result* *result* ||*t*

**od**

**return** first *Length* bits of *result,* and securely delete all unused bits

**11.11.2.4 Key confirmation with FILS authentication**

Key confirmation for FILS Authentication is is carried over the IEEE 802.11 Association Request and the 802.11 Association Response. [CID 2495]

The FILS Association Request and Association Response shall be protected using the KEK2 according to 11.11.2.6 and 11.11.2.7.

Upon the completion of key establishment (11.11.2.2) and key derivation (11.11.2.3) the STA shall construct FILS and swithin FILS Key Confirmation element. The content of the Key Auth field of the Key Confirmation element depends on the type of FILS authentication.

The AP transfers any necessary KDEs to the STA in the Association Response frame. The AP may include follow 11.6.2 (EAPOL Key Frames).

For FILS Authentication using a trusted third party, the Key Auth field of the Key Confirmation element of

the Association Request shall be:

Key-Auth = HMAC-SHA-X(KCK2, NSTA | NAP | STA-MAC | AP-BSSID).

The HMAC-SHA-Xis the first 128 bits or 256 bits of the HMAC-SHA1 of its argument list depending on the FILS AKM. (CID 2198)

For FILS Authentication without a trusted third party, the Key Auth field of the Key Confirmation element

in the Association Request shall contain a digital signature using the STA's private key, the specific construction

Key-Auth = Sig-STA(gSTA | gAP | NSTA | NAP | STA-MAC | AP-BSSID).

Where Sig-STA indicates a digital signature using the STA's private key, gSTA is the octet-string representation

of the STA's public Diffie-Hellman value, gAP is the octet-string representation of the AP's public

Diffie-Hellman value, NSTA is the nonce selected by the STA, and NAP is the nonce selected by the AP.

The 802.11 Association Request frame shall be secured as follows:

— The input key shall be the KEK2

— The input plaintext shall be the contents of the Association Request frame that follow the FILS Session

element

— The input AAD shall be:

a) The STA MAC

b) The AP BSSID

c) The STA's nonce

d) The AP's nonce

~~e) The contents of the Association Request frame from the capability (inclusive) to the FILS Session~~

~~element (inclusive)~~

— The input key, the plaintext, and the AAD shall be passed to the encrypt-and-authenticate operation

specified in 11.11.2.5.

— The output ciphertext from 11.11.2.5 shall become the remainder of the Association Request frame

that follows the FILS Session element.

The resulting 802.11 Association Request frame shall be transmitted to the AP.

The received 802.11 Association Request frame shall be processed as follows:

— The input key shall be the KEK2

— The input ciphertext shall be the contents of the Association Request frame that follow the FILS Session

element

— The input AAD shall be:

a) The STA MAC

b) The AP BSSID

c) The STA's nonce

d) The AP's nonce

~~e) The contents of the Association Request frame from the capability (inclusive) to the FILS Session~~

~~element (inclusive)~~  [Note: CID 2497]

— The input keys, the ciphertext, and the AAD shall be passed to the decrypt-and-verify operation

specified in 11.11.2.~~6.~~7

If the output from 11.11.2.~~6~~ 7 returns a failure, authentication shall be deemed a failure. If the output returns plaintext, the Key-Auth from the decrypted Association Response frame shall be checked. If it is incorrect, authentication shall be deemed a failure. If authentication is deemed a failure, the KCK2, KEK2 and TK shall be irretrievably destroyed. If authentication is not deemed a failure, the AP shall check

the Key-Auth field in the Key Confirmation element

.

For FILS Authentication using a trusted third party, the AP shall construct a verifier as follows:

Key-Auth' = HMAC-SHA-X(KCK2, NSTA | NAP | STA-MAC | AP-BSSID) [Note: 3087]

The HMAC-SHA-Xis the first 128 bits or 256 bits of the HMAC-SHA1 of its argument list depending on the FILS AKM. (CID 2198)

If Key-Auth' differs from the Key-Auth field in the Key Confirmation element, authentication shall be

deemed a failure.

For FILS Authentication without a trusted third party, the AP shall use the STA's (certified) public key from

the FILS Public Key element in the Association frame to verify the contents of the Key-Auth field of the

Key Confirmation element. The specific technique for verification depends on the crypto-system used by thepublic key. If verification fails, authentication shall be deemed a failure.

If authentication is a failure, the KCK2, KEK2, TK shall be irretrievably destroyed. Otherwise,

the AP shall then construct an 802.11 associate response frame confirming the selected ciphersuite,

For FILS authentication using a trusted third party, the Key Auth field of the Key Confirmation element in

the Association Response shall be:

Key-Auth = HMAC-SHA256(KCK2, NAP | NSTA | AP-BSSID | STA-MAC).

For FILS Authentication without a trusted third party, the Key Auth field of the Key Confirmation element

in the Association Response shall contain a digital signature using the AP's private key, the specific construction

of the digital signature depends on the crypto-system of the public/private keypair:

Key-Auth = Sig-AP(gAP | gSTA | NAP | NSTA | AP-BSSID | STA-MAC ).

Where Sig-AP indicates a digital signature using the AP's private key, and where gSTA, gAP, NSTA, and

NAP are the same as in the construction of the Association Request.

The 802.11 Association Response frame shall be protected as follows:

— The input keys shall be the KEK2

— The input plaintext shall be the contents of the Association Request frame that follow the FILS Session

element

— The input AAD shall be:

a) The AP BSSID

b) The STA MAC

c) The AP's nonce

d) The STA's nonce

— The input keys, the plaintext, and the AAD shall be passed to the encrypt-and-authentication operation

specified in 11.11.2.5.

— The output ciphertext shall become the remainder of the Association Response frame that follows the

FILS Session element.

The resulting 802.11 Association Response frame shall be transmitted to the STA.

The STA shall process the received 802.11 Association Response frame as follows:

— The input key shall be the KEK2

— The input ciphertext shall be the contents of the Association Response frame that follow the FILS

Session element

— The input AAD shall be:

a) The AP BSSID

b) The STA MAC

c) The AP's nonce

d) The STA's nonce

— The input keys, the tag, the ciphertext, and the AAD shall be passed to the decrypt-and-verify operation

specified in 11.11.2.6.

If the output from 11.11.2.6 returns failure, authentication shall be deemed a failure. If the output returns

plaintext, the Key-Auth from the decrypted Authentication frame shall be checked. If it is incorrect, authentication

shall be deemed a failure. If authentication is deemed a failure, the KCK2, KEK2, and

TK shall be irretrievably destroyed. If authentication is not deemed a failure, the STA [CID 2876]shall check the Key-Auth field in the Key Confirmation element.

For FILS Authentication using a trusted third party, the STA shall construct a verifier as follows:

Key-Auth' = HMAC-SHA256(KCK2, NAP | NSTA | AP-BSSID | STA-MAC).

If Key-Auth' differs from the Key-Auth field in the Key Confirmation element, authentication shall be

deemed a failure.

For FILS Authentication without a trusted third party, the STA shall use the AP's (certified) public key from

the FILS Public Key element in the Association frame to verify the contents of the Key-Auth field of the

Key Confirmation element. The specific technique for verification depends on the crypto-system used by the

public key. If verification fails, authentication shall be deemed a failure.

If authentication is a failure, the KCK2, KEK2, and TK shall be irretrievably destroyed.

Otherwise authentication succeeds. In that case, STA and AP shall irretrievably destroy the temporary keys

**11.11.2.5 AEAD scheme [Note: 2877, 3003, 3155, 3194, 3089,2995]**

The authenticated encryption with associated data scheme to be used shall be the negotiated cipher indicated by the cipher suite in the FILS Association Request and Response frames. The selected AEAD scheme are either based on AES-CCMP (OUI: 00-0F-AC:4) or AES-GCMP (OUI:00-0F-AC:8) as defined in table 8-111 in section 8.4.2.24.2. .

~~A, with the following instantiation:~~

~~— The block cipher shall be AES-CCM-128 (see FIPS Pub 197);~~

~~— The parameter t, q, n and shall be set to t=16, q=2, and n=13.~~

**11.11.2.6 Encrypt and authenticate operation for FILS association frames**

The AEAD scheme of 11.11.2.5 shall be used with the 802.11 Associate Request frame (for deciphering byAP) or with the 802.11 Association Response frame (for deciphering by STA), with the following instantiation:

— The key *K* shall be set to KEK2;

— The AAD should be constructed as per 11.11.2.5 [CID 2997, 2990,2993,2996,2991,2994]

— The string *P* shall be set to the plaintext;

— The nonce *N* shall be set to

a) For processing by STA: use the 13-octet all-zero string and increment by 1 in each FILS association frame transmission

b) For processing by AP: use the 13-octet all-one string and decrement by 1 in each Fils association frame transmission [CID 2202, 3154,3193,3002]

**11.11.2.7 Decrypt and verify operation for FILS association frames**

The AEAD scheme of 11.11.2.5 shall be used with the 802.11 Associate Request frame (for deciphering by

STA) or with the 802.11 Associate Response frame (for deciphering by AP), with the following instantiation:

— The key K shall be set to KEK2;

— The AAD should be constructed as per 11.11.2.5 [CID 2997, 2990,2993,2996,2991,2994];

— The string C shall be set to the ciphertext;

— The nonce N shall be set to

a) For processing by AP: use the 13-octet all-zero string and increment by 1 in each FILS association frame transmission

b) For processing by STA: use the 13-octet all- string and decrement by 1 in each Fils association frame transmission [CID ,2202, 3154,3193,3002]

The Nonce N is only temporary per FILS association and will not be re-used.

The function shall output the payload string P as the plaintext if the decryption -verification process is

successful and shall output a failure otherwise.

**~~11.11.2.8 Decrypt and verify operation for FILS association frames~~**

~~In FILS Authentication, GTK is transferred in Association Response frame from AP to STA. After successful~~

~~authentication (11.11.2.2), key derivation (11.11.2.3) and key confirmation (11.11.2.4) by the AP, the~~

~~AP shall construct Key RSC TLV (8.4.2.186.5) and GTK Transfer TLV (8.4.2.186.6). The AP shall put~~

~~these TLVs into FILS Secure Container (8.4.2.186) and shall encrypt it as described in 8.4.2.186 and~~

~~11.11.2.6.~~

~~STA shall decrypt FILS Secure Container as described in 8.4.2.186 and 11.11.2.7. The STA install GTK and~~

~~set key RSC.~~

~~GTK rekeying shall be performed as described in 11.6.7 (Group Key Handshake).~~

(Note: CID 2222)

Note to Editor: Please modify the table 8-113- AKM suite selectors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OUI | Suite type | Meaning | | |
| Authentication Type | Key Management type | Key derivation type |
| 00-0F-AC | 11 | FILS Authentication with AES-CCM (128bits) | FILS key management defined in 11.11.2.3 | Defined in 11.11.2.3 |
| 00-0F-AC | 12 | FILS Authentication with AES-GCM (128bits) | FILS key management defined in 11.11.2.3 | Defined in 11.11.2.3 |
| 00-0F-AC | 13 | FILS Authentication with AES-GCM(256bits) | FILS key management defined in 11.11.2.3 | Defined in 11.11.2.3 |