IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resolution to Comments : CID3001,3153,3192,2195,3086,2197,2199,2493,2494,2495,3259,2987,2104,2805,3243,3244,2986,2205 | | | | |
| Date: 2013-09-30 | | | | |
| 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 3001,3153,3192,2195,3086,2197,2199,2493,2494,2495,3259,2987,2104,2805,3243,3244,2986,2205

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

* STA authentication and association

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause Number(C)** | **Comment** | **Proposed Change** | **Resolution** |
| 3001 | 11.11.2.3 | PMKID needs to be defined, The length of the keys should not be set. | This appears to be incomplete. The length of the keys should be expandable/flexible/derived. | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 3153 | 11.11.2.3 | PMKID needs to be defined, The length of the keys should not be set. | This appears to be incomplete. The length of the keys should be expandable/flexible/derived. | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 3192 | 11.11.2.3 | PMKID needs to be defined, The length of the keys should not be set. | This appears to be incomplete. The length of the keys should be expandable/flexible/derived. | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 2195 | 11.11.2.3 | FILS needs to define how to generate a PMKID | A PMK is derived but there is no PMK identifier. Define one. And don't hash the PMK either. It should not be based on the PMK but, instead, based on data that uniquely generates the PMK (like the diffie-hellman exponentials) | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 3086 |  | 11.11.2.3 - p103  To account for 256-bit keys introduced in 11aci (Draft 5?), I think length of  KEK2 should at least be the length of encryption key derived from TK, and not 128 bits. |  | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 2197 | 11.11.2.3 | The hash algorithm used in the KDF should depend on the AKM used, and not be hardcoded to SHA256. If someone does Diffie-Hellman with the NIST P384 curve then they should not be required to use SHA256 with the KDF. | Specify the hash algorithm in the AKM and then in 11.11.2.3 say that the particular hash algorithm used in the KDF depends on the AKM. | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 2199 | 11.11.2.3 | The length of keys should depend on the AKM and not be hardcoded to 128 (or 256, depending on the key). This allows for security levels to be set and to use consistent cryptographic primitives. | Make the length of the keys derived, and therefore used (like the KEK and KCK) be based on the security level afforded by the AKM. | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 2493 | 4 | Problems with consistency in the use of acronyms. KEK, KCK are well specified acronyms. KEK2 and KCK2 are variable names that are included for describing a method. Define KEK2 and KCK2 precisely. | as in comment | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 2494 | 6 | Acronym overload. TK is already defined as temporal key. This paragraph redefines it to be Traffic Key. I believe the intention is still Temporal Key, if not, pick a different acronym | Amend acronym | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 2495 | 40 | Change sentence: "Key confirmation for FILS authentication is carried in an Association Request followed by ...." | as in comment | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 3259 | 11.11.2.3 | How and where to store KCK2/KEK2 which should be transient and only for FILS key confirmation? It needs to define when to destroy the KCK2/KEK2. The assumption is PMKSA which is lack of definition in elements for FILS would store the KCK2/KEK2 which may not be ideal | Please clarify | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 2987 | 11.11.2.3 | Where is the description of how the keys are derived from the KDF output. | Eiher explicity show the Key derivations from the KDF output or point ot the existing sub-clauses where these are defined. | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 2104 |  | "Where X is 1024+TK\_bits from table 11-4, rMSK is the output of the EAP-RP exchange if a trusted third party was used, and ss is the shared secret ss and rMSK, as applicable resulting from the Diffie-Hellman exchange if PFS was used" please format this according to IEEE-SA style | As in comment | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 2805 | 11.11.2.3 | The wording referring to the ss and the rMSK seems confusing to me, please clarify. | Change the sentence on line 18 from: "Where X is 1024+TK\_bits from table 11-4, rMSK is the output of the EAP-RP exchange if a trusted third party was used, and ss is the shared secret ss and rMSK, as applicable resulting from the Diffie-Hellman exchange if PFS was used." to "Where X is 1024+TK\_bits from table 11-4, rMSK is the output of the EAP-RP exchange if a trusted third party was used, and ss is the shared secret, as applicable resulting from the Diffie-Hellman exchange if PFS was used." | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 3243 | 11.11.2.3 | The reference for encrypt-and-authentication should be 11.11.2.6, and decrypt-and-verify should be 11.11.2.7. | Change as per the comment | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 3244 | 11.11.2.3 | This clause states that the KCK2 and KEK2 are temporary, which implies that they are used for only the Association Request and Response as described in clause 11.11.2.4. But there is not guidance given as to whether these keys are used for the Reassociation Request and Response frames. | Clarify how keys are generated when used with the Reassociation Request and Response frames. | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 2986 | 11.11.2.3 | Circular reference "...shall be used in exactly the same way as same-named keys of IEEE 802.11-2012 (but now derived as specified above)." | Point to the specific clauses in IEEE 802.11-2012 that this sentence references. | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |
| 2205 | 11.11.2.3 | Imposed requirements cannot be unactionable and vague | what does "used in exactly the same way as same-named keys" mean? What does this refer to? What does "(but now derived as specified above)" mean? If this is "but now" then what happens later? Entire sentence is vague and needs to be properly rewritten. | Revised  See changes in doc: IEEE 802.11-13/xxxxr0 |

***Discussion:***

Clause 11.11.2.3 outlines the procedures and key derivation with 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. I felt changing minor changes can not satisfy the comments thus I put forth a proposal with re-writing the section 11.11.2.3 and associated definitions to be complete.

In order to complete the new definitions of FMK, FTK and the security associations for FMK and FTK, the correspoinding sections are also added and updated.

***Proposed Resolution:***

**Revised**

### TGai Editor:

### Please apply the following changes to the subclauses of 11.11.2.3,

### 2) Please add the following subclauses of 11.5.1.1.13 and 11.5.1.1.14 ,

### 3) Please update the terms with the new definitions

Notes to editor: The striked texts are removed, the highlight texts are the modifications.

**11.11.2.3 Key derivation with FILS authentication**

~~Key derivation with FILS Authentication uses the KDF from 11.6.1.7.2 to produce six keys, two key~~

~~encryption keys (KEK and KEK2), two confirmation keys (KCK and KCK2), a Pairwise Master Key~~

~~(PMK), and a traffic key (TK). The inputs to the KDF are the two 16 octet nonces NSTA and NAP produced~~

~~by the STA and AP, a constant label, the EAP-RP secret result if a TTP is being used, and, the Diffie-Hellman~~

~~shared secret, ss, if PFS is being used. The length of the KEK and KEK2 shall be 128 bits, and the~~

~~length of the KCK, KCK2,and PMK shall be 256 bits, and therefore the output from the KDF shall be~~

~~1024+TK\_bits, where TK\_bits is determined from table 11-4.~~

~~KCK2 | KEK2 | KCK | KEK | PMK | TK = KDF-X(NSTA | NAP, “FILS KECK PTK Derivation”,~~

~~[rMSK]|[ ss]))~~

~~Where X is 1024+TK\_bits from table 11-4, rMSK is the output of the EAP-RP exchange if a trusted third~~

~~party was used, and ss is the shared secret ss and rMSK, as applicable resulting from the Diffie-Hellman~~

~~exchange if PFS was used.~~

~~Upon completion of the key derivation computation, the shared secret ss and rMSK, as applicable shall be~~

~~irretrievably destroyed.~~

~~The KCK2 shall only be used with key confirmation (see 11.11.2.4). The KEK2 shall only be used with the~~

~~encrypt-and-authenticate (see 11.11.2.5) and decrypt-and-verify (see 11.11.2.6) functions. Both keys KCK2~~

~~and KEK2 are temporary and[ shall only be used during the FILS authentication protocol.~~

~~The keys KCK, KEK, and TK may be used after successful completion of the FILS authentication protocol~~

~~and shall be used in exactly the same way as same-named keys of IEEE 802.11-2012 (but now derived as~~

~~specified above).~~

The FILS key hierarchy utilizes PRF-384 or PRF-512 to derive session-specific keys from an

FMK (FILS Master Key), as depicted in Figure xx. The FMK could be either mapped from rMSK if ERP with TTP is being used for FILS AKM authentication or the shared secret ss if the Diffie-Hellman shared secret for non-online TTP for FILS AKM authentication. FMK shall be 512 bits. The pairwise key hierarchy takes an FMK and generates an FTK. Then the FTK is partitioned into KCK2, KEK2, and FPTK which is further partitioned into KCK, KEK and TK for FILS session encryption. The KCK2 and KEK2 are derived for the sole purpose of protecting the FILS key confirmation (section 11.11.2.4) and both are temporary and shall only be used during the FILS authentication protocol.

Figure xx-xx FILS Key Hierarchy

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

1. The B denotes the bit length of the key which is either 128 bit or 256 bit depending on the selected cipher modes (AKM Cipher suites defined in section 8.4.2.24.3).
2. sNonce is the random or pseudorandom value contributed by the STA.
3. aNonce is the random or pseudorandom value contributed by the AP.
4. MAC\_s is the MAC address from the STA
5. MAC\_a is the MAC address from the AP
6. FMK is equal to or mapped from rMSK or Shared Secret

FTK 🡨PRF-X( FMK, “FILS FTK 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 cipher-suite as defined in . 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 be computed as the first B bits ( B is either 128 bits or 256 bit cipher suite dependent) of the FTK

KCK2 <- L ( FTK,0, B)

The KCK2 is ONLY used for the FILS key confirmation for data origin authenticity protection (Section 11.11.2.4)

1. The KEK2 shall be computed as bits ) B –> 2 \* B ( B is either 128 bits or 256 bit cipher suite dependent) of the FTK

KEK2 <- L (FTK, B,B)

The KEK2 is ONLY used for the FILS key confirmation for Key wrapping (section 11.11.2.4)

1. The rest of bits of FTK can be grouped into the PTK (PTK) in order to be mapped into the PTKSA as regular PTK management.
2. The KCK out of the FPTK is computed as bits 2\*B -> 3\*B ( B is either 128 bits or 256 bit cipher suite dependent) of the FTK

KCK <- L(FTK, 2\*B, B)

KCK is used to by IEEE Std 802.1X-2010 to provide data origin authenticity in the 4-Way Handshake and Group Key Handshake message.

1. The KEK is computed as bits 3\*B ->4\*B( B is either 128 bits or 256 bit cipher suite dependent) of the FTK

KEK <- L (FTK, 3\*B,B)

KEK is used by the EAPOL-Key frames to provide data confidentiality in the 4-way Handshake and Group Key handshake messages.

1. The Temporal Key (TK) shall be computed as bits 4\*B to (B+TK\_bits) of the FTK.

TK <- L(FTK, 4B, 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 FMK identifier is defined and computed depending on the FILS AKM is negotiated and by default is defined as

FMKID = HMAC-SHA-128 (FMK, “FMK Generation”|| aNonce||MAC\_a||sNonce||MAC\_s)

The HMAC-SHA-128 is the first 128 bits of the HMAC-SHA1 of its argument list.

If the negotiated FILS AKM is 00-0F-AC:5 or 00-OF-AC:6, HMAC-SHA-256 is used to calculated the FMKID, and the FMK identifier is defined as

FMKID = Truncate-128 (HMAC-SHA-256 (FMK, “FMK Generation”|| aNonce||MAC\_a||sNonce||MAC\_s)) )

**11.5.1.1.13 FMKSA**

An FMKSA is the result of a successful FILS Authentication and association by the FILS STA and AP. It is derived from parameters provided by STAs and AP. This security association is bidirectional between the STA and AP. In other words, both parties use the information in the security association for both sending and receiving. The FMKSA is cached for up to their lifetime. The SMKSA consists of the following elements:

— FMKID, as defined in 11.11.2.3. The SMKID identifies the security association.

— BSSID

— STA’s MAC address (MAC\_s)

— AP’sMAC address (MAC\_a)

— FMK

— Lifetime, as equal to the PMK life time settings defined in 11.6.1.3

— Pairwise cipher suite selector list, as proposed by initiator STA

— Pairwise cipher suite selector, as selected by peer STA

-

11.5.1.1.14 FTKSA

The FTKSA is a result of successful completion of the FILS authentication. The FTKSA is generated and derived from FMKSA after the FILS authentication. This security association is bidirectional between the FILS STA and AP. The FTKSA is used to manage the session keys after successful FILS authentication. FTKSAs are cached for the life of the FMKSA or until the FILS connection ends, whichever comes first. The FTKSA consists of the following elements:

— FTK (FPTK is the bit string of concatenation of KCK2, KEK2 and PTK)

--- KCK2

--- KEK2

— Pairwise cipher suite selector

— STA’s MAC address (MAC\_s)

— AP’s MAC address (MAC\_a)

— Key ID

Terms

FILS Master Key (FMK): The key derived from a rMSK generated by an Extensible Authentication Protocol (EAP) method or obtained from the shared secret.

FILS Transient Key (FTK): A concatenation of session keys derived from FMK