IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resolutions for some comments on 11ai/D4.0 (LB209) | | | | |
| Date: 2015-05-11 | | | | |
| Author(s): | | | | |
| 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 |

Abstract

This submission proposes resolutions for CIDs 7369, 7371, 7393, 7455 on 11ai/D4.0.

r1: Changes made prior to and during Mon AM1 session in Vancouver; also some of the editorial changes agreed during this session.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 7369 Mark RISON  8.4.2.186  70.55 | "The PMKID list contains a PMKID Count followed by that number of PMKIDs. The size of each PMKID is 16 octets so the length of the PMKID list element is based on the number of PMKIDs included." is useless. The first sentence is about the structure, which is shown in the figure already. The first half of the second sentence is duplication of Claus 11, and the second half of the second sentence is obvious from the figure. But there's nothing which actually tells me what the element is for! | Change to "The PMKID List element is used to <something>. The format of the PMKID List element is shown in Figure <whatever>." |

Discussion:

There is a standard way to introduce elements, and there is no reason to deviate from it.

Proposed changes:

Change 8.4.2.186 on page 70 onwards as follows:

**8.4.2.186 PMKID ~~l~~List element**

The PMKID ~~l~~List element is used to identify cached PMKSAs that the STA believes to be valid for the destination AP, or to identify the cached PMKSA selected by the AP. The format of the PMKID List element is shown in Figure 8-575ad ***[Editor: make sure this is an actual cross-reference, not just text]***. ~~contains a PMKID Count followed by that number of PMKIDs. The size of each PMKID is 16 octets so the length of the PMKID list element is based on the number of PMKIDs included.~~

***Editor: rename “PMK caching” to “PMKSA caching” throughout the spec (cf. 11mc/D4.0), and similarly for “PMK cached” and “cached PMK”.***



**Figure 8-575ad—PMKID ~~l~~List element format**

***Editor: rename the “Sequence of PMKIDs” field in Figure 8-575ad to “PMKID List”.***

The Element ID~~, Element ID Extension,~~ and Length fields are defined in 8.4.2.1 (General).

The PMKID List field contains one or more PMKIDs. The PMKID Count field indicates the number of PMKIDs in the ~~Sequence of PMKIDs~~ PMKID List field.

~~The Sequence of PMKIDs is one or more PMKIDs (as defined in 11.11.2.3 (Key establishment with FILS public key authentication) [this should have been 11.11.2.2 FILS shared key per 123.30, right?]) concatenated together.~~

Proposed resolution:

REVISED

Make the changes described in $thisdoc under “Proposed changes:” for CID 7369, which canonicalise the wording as suggested by the commenter.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 7371 Mark RISON  8.3.3.11  39 | Some of the baseline fields are used for FILS Authentication frames, so need to be described as such, in the same way the RSNE was extended at 37.34 | Extend the descriptions of the Finite Cyclic Group, Mobility Domain, Element and Fast BSS Transition to cover their possible use for FILS Authentication frames |

Discussion:

The fields in the Authentication frame vary depending on the authentication algorithm and transaction sequence number, as indicated in Table 8-44. All fields present in FILS Authentication frames need to be so specified to be present in Table 8-43, by including them in the reference to Table 8-44, as was done for the RSNE (only).

Note there is wording like “The Element field is used with SAE authentication” in 8.4.1.40 Element field (ditto Finite Cyclic Group and possibly others) which needs to be extended to cover FILS authentication; also need to check the FTE and MDE stuff too just in case they’ve been written to be FT-specific.

Proposed changes:

Add the following changes for Table 8-43:

|  |  |  |
| --- | --- | --- |
| 6 | Mobility Domain | The MDE is present in the FT Authentication frames and FILS Authentication frames as defined in Table 8-36 (Presence of fields and elements in Authentication frames). |
| 7 | Fast BSS Transition | An FTE is present in the FT Authentication frames and FILS Authentication frames as defined in Table 8-36 (Presence of fields and elements in Authentication frames). |
| 10 | Finite Cyclic Group | An unsigned integer indicating a finite cyclic group as described in 11.3.4 (Finite cyclic groups). This is present in SAE Authentication frames and FILS Authentication frames as defined in Table 8-36 (Presence of fields and elements in Authentication frames). |
| 14 | Element | A field element from a finite field encoded as described in 11.3.7.4 (Encoding and decoding of SAE Commit messages) and 11.11.2.3. This is present in SAE Authentication frames and FILS Authentication frames as defined in Table 8-36 (Presence of fields and elements in Authentication frames). |

Proposed resolution:

REVISED

Make the changes described in $thisdoc under “Proposed changes:” for CID 7371, which effect the change suggested by the commenter.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 7393 Mark RISON | Various definitions in the baseline probably need to be updated for FILS, since it does not use a 4WH. Examples: GTKSA, PTKSA, RSNA key management |  |

Discussion:

When RSNAs were introduced, an RSNA and a pre-RSNA could be distinguished by the use of a 4WH in the former, and various definitions were written accordingly. However, with FILS (and also FT (the non-initial, a.k.a. “FT Protocol authentication” (12.5.2/3), as distinct from the initial, a.k.a. “FT association” (12.4.2)) in an RSN)?) this is no longer the case, i.e. an association can be an RSNA even though a 4WH was not used.

Question: is it “FT Protocol authentication” or “FT protocol”? Make it TGmc’s problem!

So let’s search for “4-way”…

Proposed changes:

Indicate the following changes w.r.t. 11mc/D4.0:

**3.2 Definitions specific to IEEE Std 802.11**

**group temporal key security association (GTKSA):** The context resulting from a successful group temporal key (GTK) distribution exchange via ~~either~~ a Group Key Handshake ~~or~~, a 4-Way Handshake, or FILS authentication.

**pairwise transient key security association (PTKSA):** The context resulting from a successful 4-Way Handshake exchange between a peer and Authenticator or from a successful FILS authentication.

**pre-robust security network association (pre-RSNA):** The type of association used by a pair of stations (STAs) if the procedure for establishing authentication or association between them did not include the 4-Way Handshake and was not FILS authentication.

**robust security network association (RSNA):** The type of association used by a pair of stations (STAs) if the procedure to establish authentication or association between them includes the 4-Way Handshake or is FILS authentication. Note that existence of an RSNA between two STAs does not of itself provide robust security. Robust security is provided when all STAs in the network use RSNAs.

**robust security network association (RSNA) key management:** Key management that ~~includes the 4-Way Handshake, the Group Key Handshake, and the PeerKey Handshake. If fast basic service set (BSS) transition (FT) is enabled, the FT 4-Way Handshake and FT authentication sequence are also included.~~ is used in an RSNA.

**4.5.4.5 Key management**

The enhanced data confidentiality, data authentication, and replay protection mechanisms require fresh cryptographic keys and corresponding security associations. The procedures defined in this standard provide fresh keys by means of various protocols and handshakes ~~called the 4-Way Handshake, FT 4-Way Handshake, FT Protocol, FT Resource Request Protocol, and Group Key Handshake~~.

**4.10.2 IEEE Std 802.11 usage of IEEE Std 802.1X-2010**

IEEE Std 802.11 depends upon IEEE Std 802.1X-2010 and various IEEE Std 802.11 protocols and handshakes~~the 4-Way Handshake, FT 4-Way Handshake, FT Protocol, FT Resource Request Protocol, and Group Key Handshake~~, described in Clause 11 (Security) and Clause 12 (Fast BSS transition), to establish and change cryptographic keys. Keys are established after authentication has completed. Keys might change for a variety of reasons, including expiration of an IEEE Std 802.1X authentication timer, key compromise, danger of compromise, or policy.

**4.10.7 PMKSA caching**

The STA can supply a list of PMK or PSK key identifiers in the (Re)Association Request frame or first FILS Authentication frame. Each key identifier names a PMKSA; the PMKSA can contain a single PMK. The Authenticator can specify~~ies~~ a~~the~~ selected PMK or PSK key identifier in Message 1 of the 4-Way Handshake or the second FILS Authentication frame. The selection of the key identifiers to be included by the STA and Authenticator~~within the (Re)Association Request frame and Message 1 of the 4-Way Handshake~~ is out of the scope of this standard.

**10.3.5.2 Non-AP and non-PCP STA association initiation procedures**

f) If an MLME-ASSOCIATE.confirm primitive is received with a ResultCode of SUCCESS, and RSNA is required, and FILS authentication was not used, then the SME shall perform a 4-way handshake to establish an RSNA. As a part of a successful 4-way handshake, the SME shall enable~~s~~ protection by generating an MLME-SETPROTECTION.request(Rx\_Tx) primitive. If an MLME-ASSOCIATE.confirm primitive is received with a ResultCode of SUCCESS, and FILS authentication was used, then the SME shall enable protection by generating an MLME-SETPROTECTION.request(Rx\_Tx) primitive.

**WE GOT TO HERE IN THIS COMMENT AT THE END OF MON AM2 IN VANCOUVER.**

**10.3.5.3 AP or PCP association receipt procedures**

n) If RSNA establishment is required, and FILS authentication was not used, the SME shall attempt a 4-way handshake. Upon a successful completion of a 4-way handshake, the SME shall enable protection by generating an MLME-SETPROTECTION.request(Rx\_Tx) primitive [need to add to 10.3.4.2/3 auth for FILS?]. Upon receipt of the MLME-SETPROTECTION.request(Rx\_Tx), the MLME shall set the state for the STA to State 4.

**10.3.5.4 Non-AP and non-PCP STA reassociation initiation procedures**

f) If an MLME-REASSOCIATE.confirm primitive is received with a ResultCode of SUCCESS, and RSNA is required, and FILS authentication was not used, and the STA is in State 3, then the SME shall perform a 4-way handshake to establish an RSNA. As a part of a successful 4-way handshake, the SME shall enable protection by generation an MLME-SETPROTECTION.request(Rx\_Tx) primitive [need to add to 10.3.4.2/3 auth for FILS?].

**10.3.5.5 AP or PCP reassociation receipt procedures**

o) If RSNA establishment is required and FT and FILS are ~~is~~ not in use, the SME shall attempt a 4-way handshake. Upon a successful completion of a 4-way handshake, the SME shall enable protection by generating an MLME-SETPROTECTION.request(Rx\_Tx) primitive [need to add to 10.3.4.2/3 auth for FILS?]. Upon receipt of the MLME-SETPROTECTION.request(Rx\_Tx), the MLME shall set the state for the STA to State 4.

**11.4.4.4 BIP replay protection**

When management frame protection is negotiated, the receiver shall maintain a 48-bit replay counter for each IGTK. The receiver shall set the receive replay counter to the value of the IPN in the IGTK key data encapsulation (KDE) (see 11.6.2 (EAPOL-Key frames)) provided by the Authenticator in ~~either~~ the 4-Way Handshake, FT 4-Way Handshake, FT Handshake, ~~or~~ Group Key Handshake, or FILS authentication. The transmitter may reinitialize the sequence counter when the IGTK is refreshed. See 11.4.4.5 (BIP transmission) and 11.4.4.6 (BIP reception) for per packet BIP processing.

**11.5.1.1.1 General**

— PTKSA: A result of a successful 4-Way Handshake, FT 4-Way Handshake, ~~or~~ FT authentication sequence, or FILS authentication.

— Mesh TKSA: A result of a successful authenticated mesh peering exchange (AMPE).

— GTKSA: A result of a successful Group Key Handshake, 4-Way Handshake, FT 4-Way Handshake, ~~or~~ FT authentication sequence, or FILS authentication.

— IGTKSA: A result of a successful Group Key Handshake, successful 4-Way Handshake, successful FT 4-Way Handshake, ~~or~~ the Reassociation Response message of the fast BSS transition protocol when successful, or successful FILS authentication.

**11.5.1.1.6 PTKSA**

The PTKSA is a result [why no “successful” (cf. GTKSA)?] of the 4-Way Handshake, FT 4-Way Handshake, FT Protocol, ~~or~~ FT Resource Request Protocol, or FILS authentication. This security association is also bidirectional. PTKSAs are cached for the life of the PMKSA or PMK-R1 security association. Because the PTKSA is tied to the PMKSA or to a PMK-R1 security association, it only has the additional information from the 4-Way Handshake. For the PTKSA derived as a result of the 4-Way Handshake, there shall be only one PTKSA per band (see 11.5.19 (Protection of robust Management frames)) with the same Supplicant and Authenticator MAC addresses. For the PTKSA derived as a result of an initial mobility domain association or fast BSS transition or FILS authentication [is this correct here?], there shall be only one PTKSA with the same STA’s MAC address and BSSID.

During the 4-Way Handshake defined in 11.6.6.5 (4-Way Handshake Message 4) and the FT 4-Way Handshake defined in 12.4.2 (FT initial mobility domain association in an RSN), there is state created between Message 1 and Message 3 of the Handshake. This does not create a PTKSA until Message 3 is validated by the Supplicant and Message 4 is validated by the Authenticator.

During the FT authentication sequence defined in 12.8 (FT authentication sequence), the PTKSA is validated when Message 3 is validated by the R1KH and Message 4 is validated by the S1KH.

During the FILS authentication sequence defined in ***[Editor: insert cross-reference]***, the PTKSA is validated by key confirmation using (Re)Association Request and (Re)Association Response frames.

**11.5.1.1.8 GTKSA**

The GTKSA results from a successful 4-Way Handshake, FT 4-Way Handshake, FT Protocol, FT Resource Request Protocol, ~~or~~ the Group Key Handshake, or FILS authentication, and is unidirectional. In an infrastructure BSS, there is one GTKSA, used exclusively for encrypting group addressed MPDUs that are transmitted by the AP and for decrypting group addressed transmissions that are received by the STAs. In an IBSS each STA defines its own GTKSA, which is used to encrypt its group addressed transmissions, and stores a separate GTKSA for each peer STA so that encrypted group addressed traffic received from other STAs may be decrypted. A GTKSA is created by the Supplicant’s SME when Message 3 of the 4-Way Handshake is received or when Message 1 of the Group Key Handshake is received or when FILS authentication is performed. The GTKSA is created by the Authenticator’s SME when the SME changes the GTK and has sent the GTK to all STAs with which it has a PTKSA. A GTKSA consists of the following elements:

**11.5.1.1.9 IGTKSA**

When management frame protection is enabled, a non-AP STA’s SME creates an IGTKSA when it receives a valid Message 3 of the 4-Way Handshake or FT 4-Way Handshake, the Reassociation Response message of the fast BSS transition protocol with a status code indicating success, a Mesh Peering Open Message of the Authenticated Mesh Peering Exchange (AMPE) protocol [this one doesn’t need to be “valid” or “successful”?], ~~or~~ a valid Message 1 of the Group Key Handshake, or successfully performs FILS authentication. The Authenticator’s SME creates an IGTKSA when it establishes or changes the IGTK with all STAs to which it has a valid PTKSA or MTKSA.

**11.5.3 RSNA policy selection in an ESS**

If an RSNA-capable AP receives a (Re)Association Request frame that includes an RSNE and if it chooses to accept the association as a secure association, then it shall use the authentication and pairwise cipher suites in the (Re)Association Request frame, unless the AP includes an optional second RSNE in Message 3 of the 4-Way Handshake. If the second RSNE is supplied in Message 3, then the pairwise cipher suite used by the security association, if established, shall be the pairwise cipher from the second RSNE. [does this apply to/work for FILS too?]

Management frame protection is negotiated when an AP and non-AP STA set the Management Frame Protection Capable field to 1 in their respective RSNEs in the (re)association procedure, and both parties confirm the Management Frame Protection Capable bit set to 1 in the 4-Way Handshake, FT 4-Way Handshake, or the FT fast BSS transition protocol. [is there such a bit in FILS?]

**11.5.10.3 Cached PMKSAs and RSNA key management**

In a non-FT environment, a STA might ~~retain~~cache PMKSAs it establishes as a result of previous authentication. The PMKSA cannot be changed while cached. The PMK in the PMKSA is used with the 4-Way Handshake or FILS authentication to establish fresh PTKs.

Upon receipt of a (Re)Association Request frame with one or more PMKIDs, an AP checks whether its Authenticator has ~~retained~~cached a PMKSA for the PMKIDs, whether the AKM in the cached PMKSA matches the AKM in the (Re)Association Request, and whether the PMK is still valid [how can this be determined? How is this even possible – doesn’t invalidation of a PMK also invalidate the whole PMKSA?]; and if so, it shall assert possession of that PMK by beginning the 4-Way Handshake after association has completed. If the Authenticator does not have a PMK for the PMKIDs in the (Re)Association Request, its behavior depends on how the PMKSA was established. If SAE authentication was used to establish the PMKSA, then the AP STA shall reject (re)association by sending a (Re)Association Response frame with status code STATUS\_INVALID\_PMKID. Note that his allows the non-AP STA to fall back to full SAE authentication to establish another PMKSA. If IEEE Std 802.1X authentication was used to establish the PMKSA, the AP begins a full IEEE Std 802.1X authentication after association has completed.

Upon receipt of a FILS Authentication frame with one or more PMKIDs, an AP checks whether its Authenticator has cached a PMKSA for the PMKIDs, whether the AKM in the cached PMKSA matches the AKM in the FILS Authentication frame, and whether the PMK is still valid; and if so, it shall assert possession of that PMK by indicating it in the FILS Authentication frame it responds with. If the Authenticator does not have a PMK for the PMKIDs in the FILS Authentication frame, its behavior [is what?]

If both sides assert possession of a cached PMKSA, but the 4-Way Handshake or FILS authentication fails, both sides may delete the cached PMKSA for the selected PMKID.

**11.5.1.3.2 Security association in an ESS**

— A STA (AP) can ~~retain~~cache PMKSAs for APs (STAs) in the ESS to which it has previously performed a full IEEE Std 802.1X authentication or SAE authentication. If a STA wishes to roam to an AP for which it has cached one or more PMKSAs, it can include one or more PMKIDs in the RSNE of its

(Re)Association Request frame. An AP that has ~~retained~~cached the PMKSA for one or more of the PMKIDs can proceed with the 4-Way Handshake. The AP shall include the PMKID of the selected PMKSA in Message 1 of the 4-Way Handshake

**11.5.21 RSNA rekeying**

When a PTKSA is deleted, a non-AP and non-PCP STA may reassociate with the same AP or PCP and/or establish a new RSNA with the AP or PCP. If the non-AP and non-PCP STA has cached one or more PMKSAs, it may skip the PMKSA establishment and proceed with the creation of a new PTKSA by using 4-Way Handshake or FILS authentication.

When a GTKSA is deleted, an originating STA may create a new GTKSA by using 4-Way Handshake or Group Key Handshake [OK for FILS too?].

An Authenticator/STA\_I may initiate a 4-Way Handshake for the purpose of renewing the key associated with a PTKSA [OK for FILS too?] or STKSA. A supplicant/STA\_P may send an EAPOL request message to the authenticator/STA\_I to request rekeying. In addition, if both the Authenticator and the Supplicant support multiple keys for individually addressed traffic, a smooth switchover to the new key is possible using the following procedure.

**11.5.22 Multi-band RSNA**

**11.5.22.1 General**

A multi-band capable and RSNA-capable STA shall include the Multi-band element in the (Re)Association Request frame and in Message 2 and Message 3 of the 4-Way Handshake [do we need to allow for this in FILS Authentication frames too?].

**11.5.22.2 Nontransparent multi-band RSNA**

If the Joint Multi-band RSNA subfield within the RSN Capabilities field of either the RSNA initiator or RSNA responder is 0, the STA pair uses a 4-Way Handshake to establish a PTKSA for the current band/channel and may start a separate 4-Way Handshake in the current operating band/channel to negotiate a pairwise cipher suite for the supported band/channel and establish a PTKSA for the supported band/channel. As specified in 11.6.6 (4-Way Handshake), Message 2 and Message 3 of the 4-Way Handshake convey the Multi-band element associated with the supported band/channel. The Multi-band element in Message 2 includes the selected pairwise cipher suite for the supported band/channel. Message 3 includes the Multiband element that the STA would send in a Beacon, DMG Beacon, Announce, Probe Response, or Information Response frame. Message 3 may include a second Multi-band element that indicates the STA’s pairwise cipher suite assignment for the supported band/channel.

If the Joint Multi-band RSNA subfield within the RSN Capabilities field is 1 for both the RSNA initiator and the RSNA responder and at least one of the STAs uses different MAC addresses for different bands/channels, the STAs shall use a single 4-Way Handshake to negotiate pairwise cipher suites and establish PTKSAs for both the current operating band/channel and the other supported band(s)/channel(s). As specified in 11.6.6 (4-Way Handshake), Message 2 and Message 3 of the 4-Way Handshake convey the RSNE and the Multi-band element(s). The RSNE in Message 2 includes the selected pairwise cipher suite for the current operating band/channel, and the Multi-band element(s) in Message 2 includes the selected pairwise cipher suite(s) for the other supported band(s)/channel(s). Message 3 includes the RSNE and the Multi-band element(s) that the STA would send in a Beacon, DMG Beacon, Announce, Probe Response, or Information Response frame. Message 3 may include a second RSNE and Multi-band element(s) that indicate the STA’s pairwise cipher suite assignments for the current operating band/channel and the other supported band(s)/channel(s). KCK and KEK associated with the current operating band/channel shall be used in the 4-Way Handshake.

[it just goes on and on – needs FILS modifications?]

**11.6.1.3 Pairwise key hierarchy**

NOTE 4—The Authenticator and Supplicant normally derive a PTK only once per association. A Supplicant or an Authenticator use the 4-Way Handshake or FILS authentication to derive a new PTK. Both the Authenticator and Supplicant create a new nonce value for each 4-Way Handshake or FILS authentication instance. [or does this subclause not apply to FILS? If so, its title is misleading.]

**11.6.10 RSNA Supplicant key management state machine**

**11.6.11 RSNA Authenticator key management state machine**

[Need changes for FILS?]

Proposed resolution:

REVISED

Make the changes described in $thisdoc under “Proposed changes:” for CID 7393, which address the issue raised by the commenter.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 7455 Mark RISON  11.11.2.4.2  128.10 | "TK[ || FILS-FT ]" -- what does this notation mean? One of several instances referenced; others at 128.65, 130.58, 127.29 | State that square brackets indicate optional terms, or follow the approach in 14/0692r2 |

Discussion:

Here are the referenced instances:

1. PMK = HMAC-Hash(SNonce || ANonce, rMSK[||ss])
2. KCK || KEK || TK[ || FILS-FT ] = KDF-X(PMK “FILSPTK Derivation”, SPA || AA ||SNonce || ANonce)
3. Key-Auth = HMAC-Hash(KCK, SNonce || ANonce || STA-MAC || AP-BSSID[ || gSTA || gAP])..  
   where:  
   […]  
   — The brackets indicate the optional inclusion of the Diffie-Hellman public values when doing PFS with FILS shared key authentication
4. Key-Auth = HMAC-Hash(KCK, ANonce || SNonce || AP-BSSID || STA-MAC[ || gAP || gSTA]).  
   where:  
   […]  
   — The brackets indicate the optional inclusion of the Diffie-Hellman public values when doing PFS with FILS shared key authentication.

3) and 4) do state that brackets indicate optional terms, so there is no issue here (apart from the spurious full stops and missing spaces). Only the first two need addressing.

Actually, the wording is misleading. Inclusion of the term when doing PFS with shared key auth is not optional, it’s mandatory. It’s doing PFS with shared key auth which is optional.

Proposed changes:

At 127.47 add:

— The brackets indicate the inclusion of the shared secret when doing a Diffie-Hellman exchange; there is no shared secret to include otherwise

At 128.24 add:

— The brackets indicate the generation of FILS-FT when doing IKM [what is this? Got it from 133.25 but this term is never defined either in 11mc/D4.0 or in 11ai/D4.0]; FILS-FT is not generated otherwise

Delete one of the “IKM”s at 133.25.

Change 129.10 and 131.4 as follows:

— The brackets indicate the ~~optional~~ inclusion of the Diffie-Hellman public values when doing PFS with FILS shared key authentication; there are no Diffie-Hellman public values to include otherwise

Make the following changes in relation to spaces and full stops (look for angle brackets):

1. PMK = HMAC-Hash(SNonce || ANonce, rMSK<addspace>[<addspace>||<addspace>ss<addspace>])
2. KCK || KEK || TK<addspace>[ || FILS-FT ] = KDF-X(PMK “FILS<addspace>PTK Derivation”, SPA || AA ||<addspace>SNonce || ANonce)
3. Key-Auth = HMAC-Hash(KCK, SNonce || ANonce || STA-MAC || AP-BSSID<addspace>[ || gSTA || gAP<addspace>])<delete these two full stops>..  
   where:  
   […]  
   — The brackets indicate the inclusion of the Diffie-Hellman public values when doing PFS with FILS shared key authentication; there are no Diffie-Hellman public values to include otherwise
4. Key-Auth = HMAC-Hash(KCK, ANonce || SNonce || AP-BSSID || STA-MAC<addspace>[ || gAP || gSTA<addspace>])<delete this full stop>.  
   where:  
   […]  
   — The brackets indicate the inclusion of the Diffie-Hellman public values when doing PFS with FILS shared key authentication; there are no Diffie-Hellman public values to include otherwise<delete this full stop>.

Proposed resolution:

REVISED

Make the changes described in $thisdoc under “Proposed changes:” for CID 7455, which address the issue raised by the commenter in two of the referenced instances (the other two do describe the meaning of the square brackets, though the wording is misleading and has been clarified).

**References:**

P802.11ai/D4.0