**IEEE P802.11  
Wireless LANs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proposed Resolutions for some Security CIDs from LB240 | | | | |
| Date: 2019-08-13 | | | | |
| Author(s): | | | | |
| Name | Company | Address | Phone | Email |
| Nehru Bhandaru | Broadcom Inc. | 250 Innovation Drive, San Jose CA 95134 USA | 408-391-2159 | [nehru.bhandaru@broadcom.com](mailto:ganesh.venkatesan@intel.com) |

**Abstract**

This submission contains proposed resolutions for the following CIDs corresponding to some of the security related comments received for 802.11az Letter Ballot LB240 : 1853, 1918, 1447, 1107, 2016. In addition, additional changes to remove some inconsistencies in the length(s) used in key derivation and inaccuracies in test vectors are proposed.

Discussion of the comments, rationale and proposed changes are included. Where changes are primarily editorial and/or simple, the resolution column in the Summary table specifies the proposed instructions to the editor. The changes are relative to *IEEE P802.11-REVmd™/D2.3 July 2019 [1] and IEEE P802.11-az/D1.2, July 2019 [2].*

# Document History

r0 – Initial Revision

# References

[1] Draft P802.11REVmd 2.3 - [http://www.ieee802.org/11/private/Draft\_Standards/11md/Draft P802.11REVmd\_D2.3.pdf](http://www.ieee802.org/11/private/Draft_Standards/11md/Draft%20P802.11REVmd_D2.3.pdf)

[2] Draft P802.11az D1.2 - [http://www.ieee802.org/11/private/Draft\_Standards/11az/Draft P802.11az\_D1.2.pdf](http://www.ieee802.org/11/private/Draft_Standards/11az/Draft%20P802.11az_D1.2.pdf)

[3] TGaz Functional Requirements – <https://mentor.ieee.org/802.11/dcn/16/11-16-0424-11-00az-proposed-802-11az-functional-requirements.docx>

[4] TGaz Specification Framework - <https://mentor.ieee.org/802.11/dcn/17/11-17-0462-15-00az-11-az-tg-sfd.doc>

# Document Conventions

Suggested changes are specified as follows

* Red for editorial instructions
* Strikethrough for text to be deleted
* Underlined for any new proposed text
* Figures or changes to existing figures are described black and white or any other color.
* Black for existing text
* Prefix changes with

**<draft> Editor: [Add, Change, Delete, Replace] <description>(<CID>, <section>, p<page>, <line>)**

Where **<draft>** is **TGaz** for TGaz draft [2] changes or empty for base specification [1] changes

Existing clauses are identified by section, page and line numbers.

# Resolution Summary

| **CID** | **Commenter** | **Clause** | **Page.Line** | **Comment** | **Proposed Change** | **Resolution** |
| --- | --- | --- | --- | --- | --- | --- |
| 1853 | Joseph Levy | 4.5.4.2 | 15.7 | Why is PASN authentication necessary, all authentication occurs pre-association. What makes PASN necessary, why can't one of the existing authentication methods be used? | Remove PASN and use an existing authentication capability. | Rejected. PASN establishes a PTKSA before association where as other authentication methods establish a PMKSA. PTKSA is needed for secure ranging before association. |
| 1918 | Mark Hamilton | C.3 | 171.19 | Rename MIB attributes per WG recommendations | Control variables should use Activated (nor Allowed), not Implemented, per naming conventions. Capability variables should be Implemented, not Activated (see P174.26). | Accepted.  TGaz Editor: Replace all occurrences of dot11NoAuthPASNAllowed in the draft with dot11NoAuthPASNActivated |
| 1447 | Daniel Harkins | - | 20.25 | Where is figure 9-51f | Create the figure | Revised.  TGaz Editor: Revise as specified in document 11-19/1402r1 |
| 1107 | Alfred Asterjadhi | 9.4.2.118 | 45.5 | Backward compatibility issue. MIC element for these types of STAs might be variable but not for existing ones. | Ensure that all additions/changes in this amendment do not impact previous amendments (backward compatibility). | Revised.  TGaz Editor: Revise as specified in document 11-19/1402r1 |
| 2016 | Mark RISON | 6.3.5.2.2 | 5.17 | The SAP should not be described in terms of frame contents but in terms of atomic parameters (typically one per frame field/subfield) | As it says in the comment. Ditto in other 6.3.5 subclauses | Revised.  TGaz Editor: Revise as specified in document 11-19/1402r1 |
| D1402-01 | Jonathan Segev | 12.13.7 |  | the statement regarding NNN (“NNN is the Bits required for KCK and TK which is either 384 or 512 depending on the pairwise cipher., 12.13.7 PTKSA derivation with PASN authentication”)  being 384 and 512, does it take the HLTK into account? | Take HLTK into account and also revise J12 test vectors accordingly | Revised. TGaz Editor: Revise as specified in document 11-19/1402r1 |

**---**

# CID 1853 Discussion

Q: Why is PASN authentication necessary, all authentication occurs pre-association. What makes PASN necessary, why can't one of the existing authentication methods be used?

A:

PASN establishes a PTKSA before association where as other authentication methods establish a PMKSA.

PTKSA is needed for secure ranging before association. PTKSA is used to protect management frames used for ranging negotiation and deriving LTF protection key material for ranging measurement.

**---**

**CID 1918 Discussion**

Comment:

Rename MIB attributes per WG recommendations. Control variables should use Activated (nor Allowed), not Implemented, per naming conventions. Capability variables should be Implemented, not Activated (see P174.26).

Agree.

TGaz Editor: Replace all occurrences of dot11NoAuthPASNAllowed in the draft with dot11NoAuthPASNActivated

**---**

**CID 1447 Discussion**

Where is figure 9-51f, create the figure.

I cannot find any references to this figure in D1.2 or even in D1.0. I think it is referring to 9-51d in TGaz D1.0 that is missing. In TGaz D1.2, on page 25 line 30, there is a reference to 9-61d – the figure seems missing. Based on the context, it seems the figure is actually 9-61a (Figure 9-61a Ranging NDP Announcement frame format)

**TGaz Editor: Replace as follows – line 30, page 25 – Draft 1.2**

The STA Info SAC field is used in the secure variant of the non-TB Ranging measurement exchange protocol to carry the sequence authentication code (SAC), see Figure 9-61c. It is included in the Ranging NDP Announcement frame after the other STA Info field(s), see Figure ~~9-61d~~ 9.61a.

**---**

**CID 1107 Discussion**

Comment:

Backward compatibility issue. MIC element for these types of STAs might be variable but not for existing ones. Ensure that all additions/changes in this amendment do not impact previous amendments (backward compatibility).

The MIC field is only used in Mesh related frames.

The length of the MIC is specified by the algorithms used, and it seems redundant to specify it here and prevent the use of this element in contexts where MIC is not exactly 16 octets.

The protocol will anyway reject if this frame has MIC that is invalid (in length or content). However, it might be good clarify that in these frames it can only be 16.

**TGaz Editor: Instruct the Editor to Replace in 9.4.2.118 MIC Element as follows – p1279.l62 – 11md Draft 2.3**

~~The MIC field contains a message integrity code calculated over the~~ In mesh peering Management frame (as

specified in 14.5 (Authenticated mesh peering exchange (AMPE))) and the mesh group key handshake

frame (as specified in 14.6 (Mesh group key handshake))~~.~~, the length of the MIC shall be 16 octets.

In other frames, the MIC field is of variable length, depending on the context that specifies the computation of the MIC.

**CID 2016 Discussion**

Comment:The SAP should not be described in terms of frame contents but in terms of atomic parameters (typically one per frame field/subfield). As it says in the comment. Ditto in other 6.3.5 subclauses

Agree that it would be clearer – as the input is not really the frame itself and information there in. However D2.3 is still not following this convention e.g. 6.3.5.2.2 MLME...authenticate.. Contents of FT Authentication elements – says as defined in.. FT authentication sequence. Anyway…

PASN should include the following

Algorithm and Transaction are generated/specified by procotol. MIC element is computed by the protocol and not needed in the service primitive.

RSNE (xx.xx)

PASN Parameters Element (9.4.2.284)

Wrapped Data Element(9.4.2.187)

Timeout Interval Element (9.4.2.48)

**TGaz Editor: Change the text in the valid range cell for Content of PASN Authentication frame as follows – 6.3.5.2.2 p9.1 – 11azD1.2**

As defined in 12.13.2.2 PASN Frame Construction and Processing. 9.4.2.24 (RSNE), 9.4.2.187 (Wrapped Data element), 9.4.2.284 (PASN Parameters element), 9.4.2.48 (Timeout Interval element)

**TGaz Editor: Change the text in the valid range cell for Content of PASN Authentication frame as follows – 6.3.5.3.3 p10.1 – 11azD1.2**

As defined in 12.13.2.2 PASN Frame Construction and Processing. 9.4.2.24 (RSNE), 9.4.2.187 (Wrapped Data element), 9.4.2.284 (PASN Parameters element), 9.4.2.48 (Timeout Interval element)

**TGaz Editor: Change the text in the valid range cell for Content of PASN Authentication frame as follows – 6.3.5.4.2 p11.1 – 11azD1.2**

As defined in 12.13.2.2 PASN Frame Construction and Processing. 9.4.2.24 (RSNE), 9.4.2.187 (Wrapped Data element), 9.4.2.284 (PASN Parameters element), 9.4.2.48 (Timeout Interval element)

**TGaz Editor: Change the text in the valid range cell for Content of PASN Authentication frame as follows – 6.3.5.5.5 p12.1 – 11azD1.2**

As defined in 12.13.2.2 PASN Frame Construction and Processing. 9.4.2.24 (RSNE), 9.4.2.187 (Wrapped Data element), 9.4.2.284 (PASN Parameters element), 9.4.2.48 (Timeout Interval element)

**---**

**D1402-02 Discussion**

Comment: the statement regarding NNN (“NNN is the Bits required for KCK and TK which is either 384 or 512 depending on the pairwise cipher., 12.13.7 PTKSA derivation with PASN authentication”)

being 384 and 512, does it take the HLTK into account? Take HLTK into account and also revise J.12 test vectors accordingly

Quote from Jonathan’s email.

1.       KCK should be 32 octet long (“KCK is the key confirmation key of length 32 Octets”, 12.13.7 PTKSA derivation with PASN authentication), however only 16 octets are specified.

2.       Start conditions specifies HLTK to be 256 bit long

3.       And TK is 16 octet long.

As a result, KCK || TK || HLTK should be 80 bytes long i.e. 32 + 16 + 256/8 = 80 byte and NNN should be 640, while D1.2 specifies 64 byte long and 512 bit long respectively as shown in the test vector J.12.

Furthermore, the statement regarding NNN (“NNN is the Bits required for KCK and TK which is either 384 or 512 depending on the pairwise cipher., 12.13.7 PTKSA derivation with PASN authentication”)

being 384 and 512, does it take the HLTK into account?

**--**

Agree. HLTK does not seem to be taken into account. KCK should be 32 octets. Needs revision

Also, I realize that the vectors were generated using lengths larger than the maximal length specified for various combinations of BW, NSS etc. in 11.22.6.4.6.3. Since length is bound into KDF, the length has to be exact or the derived bits would be different.

It would be easier for implementations to use a number of bits that is multiple of 8.

Note that 16 bits of SAC are derived as part of DL derivation, and not as part of UL derivation.

**TGaz Editor: Change 12.13.7** PTKSA derivation with PASN authentication **p152.18 – Draft 1.2 – as follows**

DHss is the shared secret derived from the PASN ephemeral key exchange encoded as an octet 16 string (12.4.7.2.2 (Integer to octet string conversion))and NNN is the Bits required for KCK, ~~and~~ TK and HLTK which is either ~~384 or 512~~ 640 or 768 depending on the pairwise cipher.

**TGaz Editor: change the equations 11-yy and 11-zz p123.20, p123.22 as follows**

For each measurement, the maximal numbers of bits in Secure-LTF-bits-DL and Secure-LTF-bits-UL shall be derived by Equations (11-yy, shown including the SAC bits) and (11-zz), respectively.

( (4P’ + 3) x DL\_N’HE-LTF x DL\_N’REP + 16 + 7 ) & ~7), (11-yy)

..

( (4P’ + 3) x UL\_N’HE-LTF x UL\_N’REP + 7 ) & ~7), (11-zz)

--

**TGaz Editor: Change J.12 PASN Test Vectors p193 – Draft 1.2 – as follows**

Hash: SHA-256

NNN: ~~512~~ 640

…

~~KCK: 1b490923d9fd1f7038f9cdf4a615b12c~~

~~TK: adb04dad0d58c1bd26d2138d3db5bc54~~

~~HLTK:~~

~~9fa20d75e738d82f8d85343ecff8dbe2~~

~~8df2e9b7a06e7acd17b88df692ee7a0e~~

KCK:  
7bb821ac0aa5909dd654a56065ad7c77

eb889cbe2905bbf05abb1eeac88ba306

TK:  
673eab46b832d5a80cbc0243016e207e  
  
HLTK:  
2d0f0e82c70dd26b79061a4681e8dbb2

ea83bea399844bd5894eb320f69d7dd6

**---**

**TGaz Editor: Change J.13 HLTK Test Vectors when PASN authentication is not used p194 – Draft 1.2 – as follows**

~~KCK: a1f0472ce8ac7b19613eab92b3158051~~

~~KEK: 0a397c2352e1b662f9f9b994a639340d~~

~~TK: ab691823f08590886f3672b6ea18cf4f~~

~~HLTK:~~

~~2cc5f585c36adcf208f79f31556ecffd~~

~~797ffe6dfbf1b9a26168d449bf182a13~~

KCK: cd7b9e7555362df0b63568484a8112f5

KEK: 99cad3588da0f1e63fd190191039bb4b

TK: 9e2e9377e7532e737a1bc250fe194a03

HLTK:

6c7fb97ceb55b01acff00f070942bdf5

291feb4bee38e0365b25a250bb2ac9ff

**TGaz Editor: Change J.14.1 Secure-LTF-Key-Seed as follows – p195.12**

Hash: SHA-256

HLTK:

~~2cc5f585c36adcf208f79f31556ecffd~~

~~797ffe6dfbf1b9a26168d449bf182a13~~

6c7fb97ceb55b01acff00f070942bdf5

291feb4bee38e0365b25a250bb2ac9ff

Secure-LTF-Key-Seed:

~~4f85993609a54599d900a49cf799eca7~~

~~56a906d1e872f5fefc52444612a41da8~~

07606f7b0d98ca03ec2d61e17c6bdfd3

0e2f2030e3470222551a05ec55d135b9

…

Downlink Secure LTF bits are derived as follows, 176 bits that comprise of 156 bits for 80 MHz Bandwidth, two symbols for repetition and two repetitions, plus 16 bits for SAC rounded to nearest multiple of 8 bits.

SAC || Secure-LTF-DL-bits = KDF-Hash-Length(Secure-LTF-Key-Seed, “Secure LTF Expansion”, Secure-LTF-Counter)

Hash: SHA-256

Length: 176 (bits)

Secure-LTF-Key-Seed:

~~4f85993609a54599d900a49cf799eca7~~

~~56a906d1e872f5fefc52444612a41da8~~

07606f7b0d98ca03ec2d61e17c6bdfd3

0e2f2030e3470222551a05ec55d135b9

Secure-LTF-Counter: 0x000000000005

SAC: ~~83 1f~~  19 a0

Secure-LTF-DL-Bits (plus 4 bits for alignment):

~~b5 e3 0f 3e a8 20 64 cd aa c5 0a 4a 07 ab 17 5d~~

~~d7 d6 09 2b 1d c1 b6 83 66 9b 3f ce 81 39 da 1e~~

~~47 29 8c 09 07 57 d8 3c ce 0f a6 86 8e d3 12 6f~~

~~5e 8a dc 67 77 5c 23 41 6f 79 0f 22 52 71~~

3f28725c316c5e9d12a68f06a9577645

8f2d2833

Uplink Secure LTF bits are derived as follows, 160 bits that comprise of 156 bits for 80 MHz Bandwidth, two symbols for repetition and two repetitions, rounded to nearest multiple of 8 bits.

Secure-LTF-UL~~ISTA~~-bits = KDF-Hash-Length(Secure-LTF-Key-Seed, “Secure LTF Expansion”, SAC || Secure-LTF-Counter)

Hash: SHA-256

Length: 160 (bits)

Secure-LTF-Key-Seed:

~~4f85993609a54599d900a49cf799eca7~~

~~56a906d1e872f5fefc52444612a41da8~~

07606f7b0d98ca03ec2d61e17c6bdfd3

0e2f2030e3470222551a05ec55d135b9

Secure-LTF-Counter: 0x000000000005

SAC: 83 1f

Secure-LTF-UL-Bits (plus 4 bits of alignment):

~~c6 c7 67 f5 53 db 0d cb 7c d3 36 31 04 fc 19 6e~~

~~23 37 a7 5b 01 df 3d 58 5d f7 09 b1 5f 60 97 72~~

~~b3 25 8e 23 e3 98 5e 22 66 a1 9b f5 15 ef dd 60~~

~~be d5 6a ae 59 08 f2 16 1b 58 a2 4b 82 d5 9f fc~~

928e0a8d7a999f8756241ea7c7e652f6

73cc9fc6