**IEEE P802.11  
Wireless LANs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proposed Resolution for a few Security CIDs from LB240 | | | | |
| Date: 2019-03-25 | | | | |
| 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) |
| Thomas Derham | Broadcom Inc. | 16340 West Bernardo, San Diego,California,US,92127 |  | [thomas.derham@broadcom.com](mailto:thomas.derham@broadcom.com) |
| Girish Madpuwar | Broadcom Inc. | S1 SEZ, Electronic City, Bangalore, India, 560100 |  | girish.madpuwar@broadcom.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.

1090, 2222, 2318, 2319, 1906, 1459, 1460, 1461, 1458, 2320,

1457, 1293, 1029, 2221, 1030, 1031, 2313, 2323, 2395, 2017,

1903, 1905, 1726, 2237, 2076, 1032, 1034, 2289, 1751

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™/D2.1 February 2019 [1] and IEEE P802.11-az/D1.0, February 2019 [2].*

# Document History

r0 – Initial Revision

# References

[1] Draft P802.11REVmd 2.1 - <http://www.ieee802.org/11/private/Draft_Standards/11md/Draft%20P802.11REVmd_D2.1.pdf>

[2] Draft P802.11az D1.0 - <http://www.ieee802.org/11/private/Draft_Standards/11az/Draft%20P802.11az_D1.0.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** | **Comment** | **Proposed Change** | **Resolution** |
| --- | --- | --- | --- | --- | --- | --- |
| 1090 | Alecsander Eitan | 1 | 1.01 | Missing test vectors for all the Secure operations | Add test vectors in an Annex | Revised |
| 2222 | Michael Montemurro | 11 | 86.29 | PASN should be added to the state machine in clause 11.3.2 | Add PASN information in 11.3.2 and update Fig 11-16 | Revised |
| 2318 | Solomon Trainin | 12.13.1 | 133.12 | "...but is a non-RSNA protocol when there is no PMKSA and a Base AKM used with it." There is no definition or reference of Base AKM | Provide definition or reference | Revised. |
| 2319 | Solomon Trainin | 12.13.2.2 | 136.24 | "Finite cyclic group from the dot11RSNConfigDLCGroup table" There is no such a table presented | Provide definition or reference | **Accepted.**  TGaz Editor: Please replace  “dot11RSNConfigDLCGroup table”  with dot11RSNAConfigDLCGroupTable |
| 1906 | Mark Hamilton | 12.13.2.2 | 137.31 | It validates the PASN Parameters element. | Insert "element" after "PASN Parameters". Same thing at P140.25. | **Accepted.**  TGaz Editor: Please insert “element” at suggested locations. |
| 1459 | Daniel Harkins | 12.13.3 | 109.01 | How does PASN tunnel FILS shared key and SAE protocol data? Is there a reason that FILS public key is not similarly tunneled? | explain this tunneling better and if one kind of protocol cannot be tunneled explain why. | **Revised.**  Might need some help here |
| 1460 | Daniel Harkins | 12.13.3 | 109.06 | how does the "comeback cookie" work? | explain what this is for and how it works | **Revised**. |
| 1461 | Daniel Harkins | 12.13.3 | 109.16 | how are ECC private keys generated? | randomness recommendations are needed | TBD. Not specific to 11az, but may be needed in 11md |
| 1458 | Daniel Harkins | 12.13.3 |  | PASN authentication does an ephemeral D-H but it's not clear what's done with the resulting secret. The claim is that this exchange produces a PTKSA according to 12.xx (which is not very helpful) but PTKSAs are created as the result of a 4-way handshake (as described in the changes to 12.2) not as the result of an ephemeral key exchange which normally results in a PMKSA | Explain which kind of SA is created, what happens to the D-H shared secret. Be consistent with the rest of the standard-- key exchanges produce PMKSA whose key is used in the 4way handshake to produce PTKSAs. | Rejected. See discussion |
| 2320 | Solomon Trainin | 12.13.3.1 | 136.08 | "The lifetime of the PTKSA is shall be the minimum based on the timeout information exchanged (if any) but shall not exceed the lifetime of the PMKSA for the Base AKM." Awkward sentence "is shall". Lack of definition or reference of "lifetime of the PMKSA". What is the "timeout information exchanged"? The definition is not feasible for implementation and verification. | Provide full definition and references | Revised. |
| 1457 | Daniel Harkins | 12.2 |  | apparently the KDF crank is turned to produce HLTK but which PMK is this? The one from the "base AKM"? | indicate which key is being input to the KDF | Revised. See proposed changes in this document |
| 1293 | Assaf Kasher | 12.2.11 | 128.27 | "Secret Key" - what secret key | refere to where the secret key is defined | Revised: TGaz Editor: Add a reference - (See  Figure 9-619e – Ranging Operation Parameters field format  ) - after “Secret Key” |
| 1029 | Albert Petrick | 12.6.1.1.1 | 129.32 | Fix PTKSA: to be consistent with 802.11REVmd D2.1 | Change: "sequence, or FILS.." to "sequence, FILS.." | Accepted: TGaz Editor: Remove word “or” in the context as suggested |
| 2221 | Michael Montemurro | 12.6.1.1.6 | 130.08 | "the timeout negotiated, if any, during PASN authentication." | For a number of "base" AKMs, the STA has no knowledge of the PMKSA lifetime, so add behavior to describe what a STA does if it does not receive the PMKSA lifetime information. | Rejected. Default seems ½ day (43200 seconds) based on MIB default. Current spec [1] says “The PTK shall not be used longer than the PMK lifetime as determined by the minimum of the PMK lifetime  indicated by the AS, e.g., Session-Timeout + dot1xAuthTxPeriod or from dot11RSNAConfigPMKLifetime.  When RADIUS is used and the Session-Timeout attribute is not in the RADIUS Accept message,  and if the key lifetime is not otherwise specified, then the PMK lifetime is infinite.” |
| 1030 | Albert Petrick | 12.6.1.1.6 | 130.11 | Fix the inclusion of additional information remove "or" | Change "handshake, or FT Protocol authentication, or FILS authentication, or PASN authentication. " to " handshake, FT Protocol authentication, FILS authentication, or PASN authentication. " | Accepted.  TGaz Editor: Please change as follows.  …  handshake, ~~or~~ FT Protocol authentication, ~~or~~ FILS authentication, or PASN authentication. |
| 1031 | Albert Petrick | 12.6.10.1 | 130.23 | Improve sentence structure | Change to; "PASN authentication may be used without (re)association." | Accepted. TGaz Editor: Please change as suggested. |
| 2313 | Rojan Chitrakar | 12.7.1.3 | 131.01 | REVmd\_D2.0 (line 11) defines the PTK lengths as:  Length = KCK\_bits + KEK\_bits + TK\_bits  This needs to be revised. | change line 11 of REVmd\_D2.0 as:  where Length = KCK\_bits + KEK\_bits + TK\_bits + HLTK\_bits | Accepted. See changes in the document. |
| 2323 | Song-Haur An | 4.5.4.2 | 7.14 | How is PASN required or used for this amendment (Enhancements for Positioning)? A high level illustration will be useful. | Please add a brief illustration how this method may enable the enhancements for positioning following the first time introduction of this method. | TBD – Perhaps a separate submission |
| 2395 | Tomoko Adachi | 4.5.4.2 | 7.22 | The content in the sentence starting from "PASN authentication allows ...." is repeated here. Delete the repeated part and just mention what's new. | As in comment. | Rejected. Granted ..allows management frame protection… is repeated from Glossary, but the current text is clearer in the context… |
| 2017 | Mark RISON | 6.3.5.2.2 | 5.15 | [Re-raising this comment from the comment collection, as it is not possible to determine from 18/1544r8 whether/how it was addressed. References are to the CC draft and hence may be wrong against D1.0.]  Missing comma | Add comma after "FILS\_PUBLIC\_KEY". Ditto in other 6.3.5 subclauses | Rejected. Not able to find FILS\_PUBLIC\_KEY in the 11az D1.0 |
| 1903 | Mark Hamilton | 6.3.5.2.2 | 8.11 | There is no PASN authentication frame. There are only Authentication frames that are carrying PASN parameters or element(s). (Yes, I'm putting the same comment into REVmd to fix the similar issues with FILS and SAE.) Secondly, is there more to "PASN Parameters" than what is in the PASN element? It doesn't seem so. | Change "PASN authentication frame" to "PASN element" throughout clause 6. Change "Sequence of elements and fields" in P9.1 to "PASN element", and change the valid range to reference 9.4.2.284 (instead of something in clause 12), and change the Description to start, "The element to be included...". Similarly in the other MLME-AUTHENTICATE primitives. Change the title of 12.13.2.2 to something more appropriate. | Rejected. PASN authentication frame is an authentication frame with algorithm set to PASN authentication algorithm. Would like to see resolution of 11md comment before making additional changes here. |
| 1905 | Mark Hamilton | 9.3.3.12 | 30.04 | The PASN element is not optional in a "PASN authentication frame" (sic). That is the (apparent) definition of "PASN authentication frame" (which needs to be reworded - see my other comment). | Change the Notes column to, "A PASN element is present only in certain Authentication frames as defined in Table 9-43 (Presence of fields and elements in Authentication frames)." | Accepted. Agree – since 9-43 already specifies what those frames are.  TGaz Editor: Please change as suggested. |
| 1726 | Hiroyuki Motozuka | 9.3.3.12 | 38.04 | FILS Wrapped Data element is renamed to Wrapped Data element in 9.4.2.187 of this amendment. The changes need to apply all of the REVmd text. | Add text amendment everywhere "FILS Wrapped Data element" appears in REVmd.  - Table 9-42 Authentication frame body  - Table 9-43 Presence of fields and elements in Authentication frames  - Table 9-94 Element IDs  - In subclauses of 12.12.2.3 Key establishment with FILS Shared Key authentication | Revised. |
| 2237 | Nehru Bhandaru | 9.4.2.1 | 33.05 | Table 9-87 -- Element IDs is missing the row for PASN Parameters | Add a row for PASN Parameters - an extensible, but non-fragmentable element  PASN Parameters 230 <ANA> Yes No | Accepted.  TGaz Editor please make the suggested change to table (Table is 9-94 in 11md D2.1) |
| 2076 | Mark RISON | 9.4.2.251 | 35.30 | [Re-raising this comment from the comment collection, as it is not possible to determine from 18/1544r8 whether/how it was addressed. References are to the CC draft and hence may be wrong against D1.0.]  "The Secure LTF Parameters element contains a set of fields." is impressively useless | Delete | **Accepted**. Cannot find the text in Draft 1.0. No further action needed. |
| 1032 | Albert Petrick | ANNEX C | 134.23 | Missing MIB "dot11RSNAConfigDLCGroupTable" add to ANNEX C | As commented | **Accepted.** A typo.  TGaz Editor: Please replace  dot11RSNConfigDLCGroupTable  with dot11RSNAConfigDLCGroupTable |
| 1034 | Albert Petrick | ANNEX C | 134.23 | Missing MIB "dot11RSNConfigDLCGroup" add to ANNEX C | As commented | **Accepted.** A typo.  TGaz Editor: Please replace dot11RSNConfigDLCGroup with  dot11RSNAConfigDLCGroup. |
| 2289 | Qi Wang |  | 59.11 | "LTF Sequence Generation Information" seems to be synonymous with "secure LTF counter" in the spec. "Secure LTF counter" is a more descriptive name to use than the field name of "LTF sequence generation information". | Replace "LTF sequence generation information" with "secure LTF counter" throughout the spec. | **Accepted.**  TGaz Editor: Please replace “LTF sequence generation information” with “Secure LTF Counter” in the amendment text. Note in many cases LTF is followed by two spaces. |
| 1751 | James Lepp | 12.13.2.2 | 139.28 | Verifies that the public key WHAT as specified in 5.6.2.3 of NIST SP 800-56A R2? Not sure if you need to remove "that" or add something before "as" to make this sentence work. | |  | | --- | | Remove "that" or add to the sentence to clarify what it is verifying | | Accepted.  TGaz Editor: Please remove the word “that” as suggested. |

# CID 1090 Discussion

Test vectors for secure operations are missing. These include

1. DHss derivation
2. PTKSA (including HLTK) derivation (12.13.7) from PMK
3. MIC computation for PASN second frame (12.13.8.1)
4. MIC computation for PASN third frame (12.13.8.2), mislabeled 12.13.7.2 in D1.0
5. LTF Sequence generation

Probably need a PASN protocol implementation for [2-4].

No ANA assignment for PASN authentication algorithm yet.

Detailed derivation using standard algorithms selected for HASH etc. given particular frames (e.g. EAPOL, FT etc) is not provided in the test vectors. Perhaps we can start with test vectors for 1, 2, and 5. The document can be updated with additions for 3 and 4 at a later time.

Also note that some of the test vectors (J.7.2 CCMP…) do not take into account the key descriptor version and/or AKM into consideration or make assumptions regarding.

**TGaz Editor: Add annex J.<mm> with the test vectors for PASN derivation(s) where mm is the next available sub-section in annex J in [1]**

# J.mm PASN Test Vectors

Diffie-Hellman exchange to establish a shared secret used in key derivation for PASN and FILS with PFS is a well-known algorithm. The reader is referred to IETF RFC 5903 (https://www.ietf.org/rfc/rfc5903.txt ) for test vectors when using ECP groups. Only the x-coordinate of the result of scalar multiplication of a private key with the received public key is used as the shared secret.

Sample PTKSA derivation with PASN Authentication

KCK || TK || HLTK = KDF-HASH-NNN (PMK, “PASN PTK Derivation”, SPA || BSSID || DHss)

Hash: SHA-256

NNN: 512

PMK:

def43e5567e01ca6649265f19a290eef

f8bd888f6c1d9cc9d10f04bd378f3cad

SPA: 00904c01c107

BSSID: c0ffd4a8dbc1

DHss: Sample using NIST P256 (Group 19)

f87b208e7ed2b737afdbc2e13eae78da

300123d4d84ba8b0eafe90c48cdf1f93

KCK: 1b490923d9fd1f7038f9cdf4a615b12c

TK: adb04dad0d58c1bd26d2138d3db5bc54

HLTK:

9fa20d75e738d82f8d85343ecff8dbe2

8df2e9b7a06e7acd17b88df692ee7a0e

**TGaz Editor: Add annex J.<pp> with test vectors for HLTK generation as part of PTKSA generation when PASN authentication is not used**

# J.pp HLTK Test Vectors when PASN authentication is not used

KCK || KEK || TK || HLTK = KDF-Hash-Length(PMK, “Pairwise key expansion”, Min(AA, SPA) || Max(AA, SPA) ||

Min(ANonce, SNonce) || Max(ANonce, SNonce)

Hash: SHA-256

Length:640

PMK:

def43e5567e01ca6649265f19a290eef

f8bd888f6c1d9cc9d10f04bd378f3cad

AA: c0ffd4a8dbc1

SPA: 00904c01c107

ANonce:

be7a1ca284347b5bd67dbd2dfdb4d99f

1afae0b88ba18e008718417e4b27ef5f

SNonce:

404b012ffb43ed0fb43ea1f287c91f25,

06d21b4a92d74b5e a50c943350ce8671

KCK: a1f0472ce8ac7b19613eab92b3158051

KEK: 0a397c2352e1b662f9f9b994a639340d

TK: ab691823f08590886f3672b6ea18cf4f

HLTK:

2cc5f585c36adcf208f79f31556ecffd

797ffe6dfbf1b9a26168d449bf182a13

**TGaz Editor: Add annex J.<nn> with test vectors for LTF sequence generation where <nn> is the next available sub-section in annex J in [1].**

# J.nn LTF Sequence Generation Test Vectors

**J.nn.1 Secure-LTF-Key-Seed**

As defined in 11.22.6.3.4, Secure-LTF-Key-Seed is derived from HLTK as follows

Secure-LTF-Key-Seed = HMAC-Hash(HLTK, “Secure LTF key seed”)

Hash: SHA-256

HLTK:

2cc5f585c36adcf208f79f31556ecffd

797ffe6dfbf1b9a26168d449bf182a13

Secure-LTF-Key-Seed:

4f85993609a54599d900a49cf799eca7

56a906d1e872f5fefc52444612a41da8

Downlink Secure LTF bits are derived as follows

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

Hash: SHA-256

Length: 512 (bits)

Secure-LTF-Key-Seed:

4f85993609a54599d900a49cf799eca7

56a906d1e872f5fefc52444612a41da8

Secure-LTF-Counter: 0x000000000005

SAC: 83 1f

Secure-LTF-DL-Bits:

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

Uplink Secure LTF bits are derived as follows

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

Hash: SHA-256

Length: 512 (bits)

Secure-LTF-Key-Seed:

4f85993609a54599d900a49cf799eca7

56a906d1e872f5fefc52444612a41da8

Secure-LTF-Counter: 0x000000000005

SAC: 83 1f

Secure-LTF-UL-Bits:

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

**---**

**CID 1457 Discussion**

The comment is

“apparently the KDF crank is turned to produce HLTK but which PMK is this? The one from the "base AKM"?”

Agreed. It seems to be missing from the description.

**TGaz Editor: Change HLTK derivation with PASN as follows (1457, 12.13.7 p143.29)**

KCK || TK || HLTK = KDF-HASH-NNN (PMK, “PASN PTK Derivation”, SPA || BSSID || DHss)

Where

PMK is the pairwise master key for the base AKM if the AKM is other than PASN AKM (9.4.2.24.3 AKM Suites). Otherwise, if the base AKM is PASN AKM i.e. the PASN PTKSA is being setup without mutual authentication in a non-RSN, the PMK shall be set to “PMKz”||^28 i.e. the string “PMKz” padded with 28 0s. Note that the PMK for the derivation may come from a cached PMKSA for the AKM or from the PMKSA established with PASN by tunneling Wrapped Data or Authentication frames.

KCK is the key confirmation key of length 32 Octets.

...

~~If the PTKSA is being setup without a PMKSA i.e. without mutual authentication in a non-RSN, PMK shall be set to “PMKz”||028 i.e. the string “PMKz” padded with 28 0s.~~

**---**

**CID 1459 Discussion**

The comment was:

How does PASN tunnel FILS shared key and SAE protocol data? Is there a reason that FILS public key is not similarly tunneled?

explain this tunneling better and if one kind of protocol cannot be tunneled explain why.

Discussion:

With respect to FILS shared key:

The tunneling for FILS shared key is described in 12.13.4 PASN authentication with FILS Shared Key. Essentially the rMSK is obtained by tunneling Erp messages in Wrapped Data element as described.

The Wrapped Data element to AP contains EAP-Initiate/Re-auth frame and to STA contains EAP-Finish/Re-auth frame

The Wrapped Data element is fragmentable (9.4.2.187)

There is only the PFS option with PASN.

With respect to SAE:

SAE authentication frames are tunneled in Wrapped Data element. As described

With respect to FILS Public Key:

Needs more machinery and seems complicated.

Uses of public keys, ephemeral keys, Nonces

Need to take into account gSTA/gAP to confirm keys – currently the PTKSA derivation and key confirmation is common to all base AKMs. Altneratively FILS key derivation may need to change to take gSTA/gAP public keys into account

No fundamental reason not to support it

perhaps can be taken up separately – no FILS public key deployments/certification…

Not sure what the purpose of nonces here are – since DHss should provide freshness...

Need more concrete suggestions to improve this.

**TGaz Editor: Change 12.13.4 PASN authentication with FILS Shared Key as follows (1459, 12.13.4 p141.12)**

This subclause specifies aspects of PASN authentication when FILS AKM 00-0F-AC:14 or 00-0F-AC:15 is used as a Base AKM when PMK Caching is not used with PASN authentication. Otherwise, when PMK Caching is used, the PMKSA identified by the PMKID in the first PASN authentication frame is used for the PASN PTKSA derivation.

Where FILS shared key authentication without PFS is desired and there is no cached PMK, the Base AKM data is constructed using Wrapped Data element (9.4.2.187 (Wrapped Data element)).

In the first PASN frame, ~~it~~ the Wrapped Data element contains the EAP-Initiate/Re-auth packet similar to FILS shared key processing (see 12.12.2.3.2 (Non-AP STA construction of FILS Authentication frame)). The EAP-Initiate/Re-auth data is forwarded to the AS by the AP and the EAP-Finish/Re-auth packet received from the AS is forwarded to the non-AP STA encapsulated in the Wrapped Data element in the second PASN frame. The AS returns the associated EAP-RP rMSK along with the EAP-Finish/Re-auth packet to the AP. Wrapped data shall be absent in the third PASN frame.

…

The PMKSA is then used in PTKSA derivation for PASN authentication. A FILS PMKSA so established shall be used only to establish PTKSAs that are negotiated via PASN authentication.

There is only a single FILS session with PASN authentication, and FILS session element is not tunneled.

**TGaz Editor: Change 12.13.5 PASN authentication with SAE as follows (1459, 12.13.5 p141.33)**

This subclause specifies aspects of PASN authentication when SAE AKM 00-0F-AC:8 is used as the Base AKM when PMK caching is not ~~use~~ used.

**---**

**CID 1460 Discussion**

The comment was: Explain comeback cookie

**TGaz Editor: Change 9.4.2.284 PASN Parameters Element as follows (1460, 9.4.2.284, p58.21)**

**~~Figure 9- 1019 PASN Parameters Element Control field~~**

**Figure 9- 1019 PASN Parameters Element Comeback Info field**

where the Comeback After subfield is time in TUs after which the non-AP STA is requested to retry the PASN authentication. ~~It may be present in frames from a non-AP STA that is retrying the authentication (see 12.13.3 PASN Frame Construction and Processing).~~ Comeback After subfield may be 0 indicating that the operation may be retried with the Cookie of non-zero length that shall be present. Comeback After subfield shall not be present (i.e. zero octets) in PASN authentication frames from a non-AP STA.

Cookie Length field is the length of the following Cookie. Cookie length may be 0, indicating that there is no cookie.

Cookie (see 12.13 Pre-Association Security Negotiation)

**TGaz Editor: Change 12.13.1 General as follows (1460, 12.13.1 p133.30)**

A provision to allow APs to request the peer to come back later based on resource constraints or other conditions. See Comeback Cookies (12.13.xx). ~~The comeback cookie can also be used as an anti-clogging token for better denial of service protection on the AP.~~

**TGaz Editor: Add subclause 12.13.xx for a description of Comeback Cookies (1460)**

**12.13.xx Comeback Cookies**

An AP is required to do a considerable amount of work upon receiving first PASN authentication frame and thus be subjected to denial of service attacks by an attacker by spraying such frames, each possibly with a different source MAC address. An AP may also have limited resources and may desire to defer authentication for new client STAs. PASN authentication allows an AP to indicate the deferral time (Comeback After field in PASN Parameters Element) and optionally a Cookie in the Comeback Information field of the PASN parameters element (9.4.2.284 PASN Parameters) and refuse the authentication temporarily.

The Comeback Cookie is used similar to the SAE anti-clogging token (12.4.6 Anti-clogging tokens) for denial of service protection on an SAE peer. When a client STA receives the Comeback Information with a deferral time and a Cookie, the STA can re-initiate PASN authentication by providing the Cookie. The AP would accept or deny authentication based on the received cookie, if one was expected. PASN Comeback Cookies are an opaque sequence of octets to the client. They are generated and validated by the AP by an implementation dependent scheme. An AP may start requiring clients present the Comeback Cookie provided based on resource or other constraints outside the scope of this specification.

**---**

**CID 1461 Discussion**

Comment was “randomness recommendations needed…”

Not sure what we want to add here that is specific to PASN.

Ephemeral keys and DH are used in other places – SAE description, FILS w/ PFS

Spec already has J.5 suggestions for random number generation.

Perhaps something more concrete is needed – or point to an IETF RFC like RFC6090 – fundamental ECC algorithms, and random number generation requirements RFC 1750 – Randomness Recommendations for Security

**---**

**CID 2318 Discussion**

Comment was:

"...but is a non-RSNA protocol when there is no PMKSA and a Base AKM used with it." There is no definition or reference of Base AKM

Perhaps it could be introduced at the beginning of the subsection.

**TGaz Editor: Change 12.13.1 General as follows (2318, 12.13.1 p133.10)**

Pre-Association Security Negotiation (PASN) is an RSNA authentication protocol in all cases where it relies on the existence of a PMKSA for an AKM, termed Base AKM for PASN. It is , ~~but is~~ a non-RSNA protocol when there is no PMKSA and ~~a~~ the corresponding Base AKM used with it.

**---**

**CID 1458 Discussion**

The comment was:

PASN authentication does an ephemeral D-H but it's not clear what's done with the resulting secret. The claim is that this exchange produces a PTKSA according to 12.xx (which is not very helpful) but PTKSAs are created as the result of a 4-way handshake (as described in the changes to 12.2) not as the result of an ephemeral key exchange which normally results in a PMKSA…

Explain which kind of SA is created, what happens to the D-H shared secret. Be consistent with the rest of the standard-- key exchanges produce PMKSA whose key is used in the 4way handshake to produce PTKSAs.

..

The shared secret DHss is used in PTKSA derivation – see 12.13.7 - to derive KCK, TK and HLTK.

Agreed that this is somewhat inconsistent with the current draft in that any DH shared secret is used to derive PMKSA, but DH can be used to provide PFS for any negotiation.

There is already a PMKSA for the Base AKM and we do not want to derive another one and specify behavior for PMKSA caching etc. ECDH (256-bit) is relatively cheap with modern hardware. We just need a PTKSA; rather than use Nonces DHss provides PFS for PTKSAs.

PASN negotiation is a replacement for 4-way handshake to generate a PTKSA; similar to FT authentication being one.

NIST SP800-56A has ECDH key generation requirements – private key requirements being an approved DRBG/RBG and the private key in range [1, n-1] where n is the order of the group.

**---**

**CID 2320 Discussion**

Comment was:

"The lifetime of the PTKSA is shall be the minimum based on the timeout information exchanged (if any) but shall not exceed the lifetime of the PMKSA for the Base AKM." Awkward sentence "is shall". Lack of definition or reference of "lifetime of the PMKSA". What is the "timeout information exchanged"? The definition is not feasible for implementation and verification.

Agreed. Needs some rewording.

PMK Lifetime is an existing notion used in [1]

**TGaz Editor: Change 12.13.3.1 Overview as follows (2320, 12.13.3.1 p136.8)**

~~The lifetime of the PTKSA is shall be the minimum based on the timeout information exchanged (if any) but shall not exceed the lifetime of the PMKSA for the Base AKM.~~

The lifetime of the PTKSA shall be the minimum of PTKSA timeout interval(s) exchanged (if any) but shall not exceed the PMK lifetime for the PMKSA of the Base AKM.

**---**

**CID 2313 Discussion**

Comment was:

REVmd\_D2.0 (line 11) defines the PTK lengths as:

Length = KCK\_bits + KEK\_bits + TK\_bits

This needs to be revised…

change line 11 of REVmd\_D2.0 as:

where Length = KCK\_bits + KEK\_bits + TK\_bits + HLTK\_bits

**Editor: Change 12.7.1.3 Pairwise key hierarchy as follows (2313, 12.7.1.3 p2608.11)**

— The PTK shall be derived from the PMK by

PTK = PRF-Length(PMK, “Pairwise key expansion”, Min(AA,SPA) || Max(AA,SPA) ||

Min(ANonce,SNonce) || Max(ANonce,SNonce))

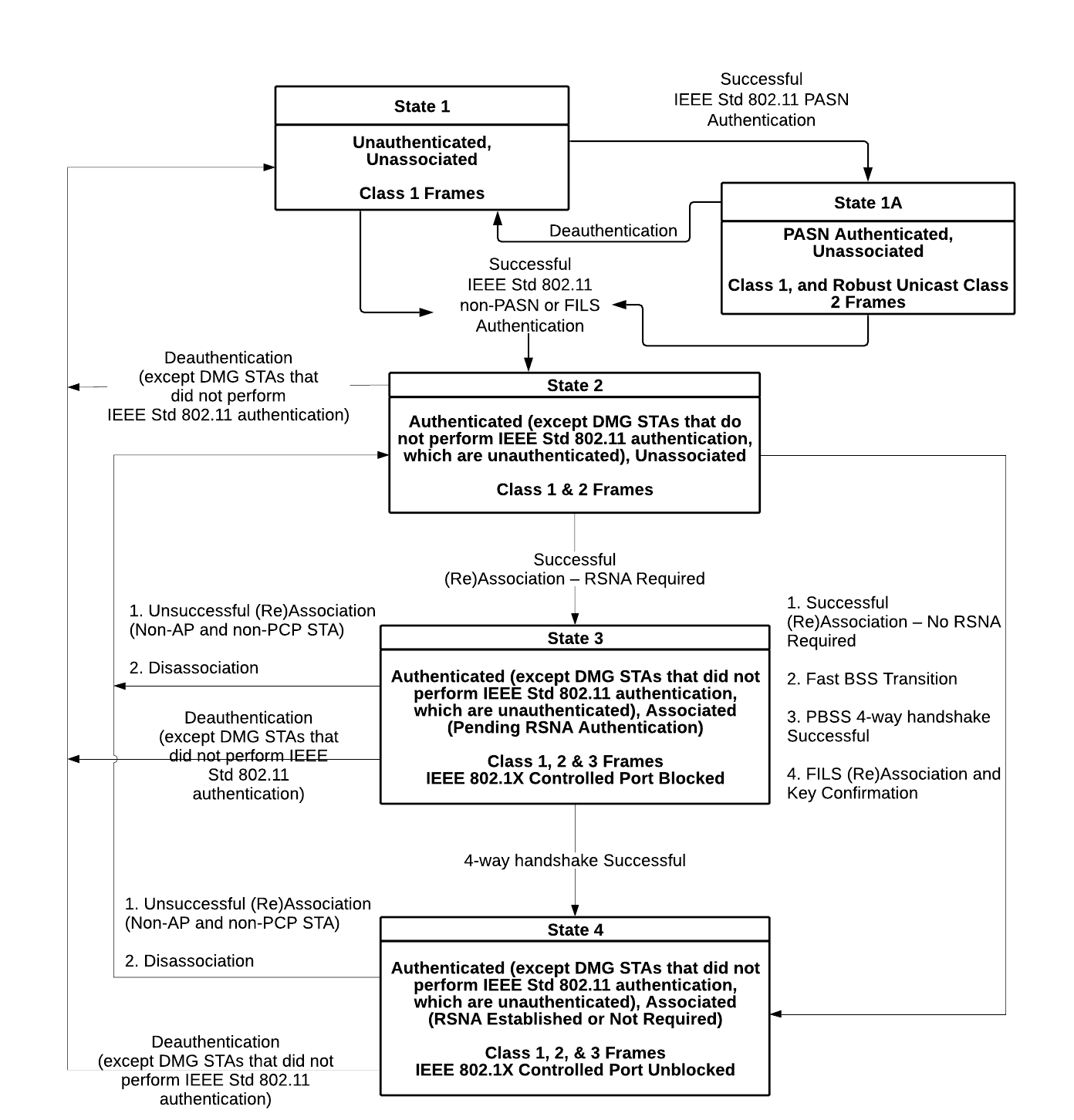
where Length = KCK\_bits + KEK\_bits + TK\_bits + HLTK\_bits.

**---**

**CID 2222 Discussion**

Comment was: PASN should be added to the state machine in clause 11.3.2

**Editor: Replace Figure 11-16...Relationship between state and services… with the following figure (2222, 11.3.2, p2194.3)**



**---**

**CID 1726 Discussion**

Comment was:

FILS Wrapped Data element is renamed to Wrapped Data element in 9.4.2.187 of this amendment. The changes need to apply all of the REVmd text…

Add text amendment everywhere "FILS Wrapped Data element" appears in REVmd.

- Table 9-42 Authentication frame body

- Table 9-43 Presence of fields and elements in Authentication frames

- Table 9-94 Element IDs

- In subclauses of 12.12.2.3 Key establishment with FILS Shared Key authentication

**Editor: Change 4.10.3.6.2 AKM operations using FILS Shared Key auth.. as follows (1726, 4.10.3.6.2 p290.14)**

Authentication Protocol (EAP) reauthentication protocol (EAP-RP) signaling. EAP-RP signaling is

encapsulated using ~~FILS~~ W~~w~~rapped D~~d~~ata element in an Authentication frame

**Editor: Change 12.12.2.3.4 AP construction of Authentication frame as follows (1726, 12.12.2.3.4 p2678.35)**

If PMKSA caching is not used, this frame shall contain the ~~FILS~~ W~~w~~rapped D~~d~~ata element that encapsulates EAP-Finish/Re-auth packet received from the Authentication Server.

**Editor: Change 12.12.2.3.3 AP processing of Authentication frame as follows (1726, 12.12.2.3.3 p2677.60)**

3) If an EAP-Initiate/Re-auth packet is included and PMKSA caching is not used, the AP shall

extract the EAP-Initiate/Re-auth data from the ~~FILS~~ W~~w~~rapped D~~d~~ata element ~~field~~ (see 9.4.2.187 (~~FILS~~

Wrapped Data element(11ai))) and shall forward it to the Authentication Server.

**Editor: Change 12.12.2.3.2 Non-AP STA construction of Authentication frame as follows (1726, 12.12.2.3.2 p2677.16)**

The EAP-Initiate/Re-auth packet, if generated, shall

be copied into the ~~FILS~~ W~~w~~rapped D~~d~~ata element ~~field~~ (see 9.4.2.187 (~~FILS~~

Wrapped Data element(11ai))).

**Editor: Change 12.12.2.3.1 Overview as follows (1726, p2676.5)**

the AP forwards the packet to the STA by encapsulating the

EAP-RP packet in the ~~FILS~~ Wrapped Data element of the Authentication frame.

**Editor: Change 12.12.2.3.1 Overview as follows (1726, p2675.65)**

EAP-RP signaling is encapsulated using

a ~~FILS~~ Wrapped Data element in the Authentication frame.

**Editor: Change tables 9-42 (p874.24), 9-43 (p877.20, p876.32, p876.19 and p876.53), 9-94 (p997.34) and tables in sections 6.3.5.5.3 (p350.12), 6.3.5.4.2(p348.12), 6.3.5.3.2 (p346.39), 6.3.5.2.2(p344.50), B4.2.7 FILS features (p3748.21), by replacing all occurrences of** FILS Wrapped Data **with** Wrapped Data

~~FILS~~ Wrapped Data

**---**