IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proposed revisions to PASN based on WFA Security Review | | | | |
| Date: 2021-10-27 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Nehru Bhandaru | Broadcom | 250 Innovation Drive, San Jose CA 95134 | +1 408 391 2159 | nehru.bhandaru@broadcom.com |
|  |  |  |  |  |

Abstract

Wi-Fi Alliance has requested the IEEE802.11 Working Group to consider the comments on P802.11az 3.1 draft received from an external security review to improve the P802.11az draft (see [11-21-1524r0](https://mentor.ieee.org/802.11/dcn/21/11-21-1524-00-0000-communication-from-wfa-re-p802-11az.docx)). This document captures the comments, discussion, and proposed resolution for the comments. It also includes a proposed set of changes to P802.11 4.0 draft to address the comments.

Note – the original comments specify page and line numbers corresponding to TGaz draft 3.1; the proposed changes provide the mapping that corresponds to TGaz draft 4.0.

**Revision History:**

0 – Initial revision

1 – Update from the Sept 29, 2021, TGaz conf. call

2 – Update after Sep 29, 2021, TGaz conf. call

3 – Update during the Nov meeting 11/09/21

In the text below, the verbatim received comments are in blue, the discussion is in black, and the proposed summary resolution is in red. In terms of process, the summary resolution can turn green when there is consensus from TGaz group.

1. General: Are all the authentication options required? The concern is implementation errors in infrequently used options.

It is always a balance to choose what features to include and what not to include. PASN is dependent on a base AKM for authentication and the options it provides encompasses some of the common AKMs of interest and their options. It could be argued that some of the options, may be comeback cookies etc., might not be used immediately but nevertheless useful. It would have been better if the review identified a few options that might not be required. In general, standards maintenance has a process to deprecate and obsolete options that are not used over time. Devices implementing the standard would, in general, go through industry certification programs that are intended cover all the implemented options which usually do not include infrequently used options. However, we acknowledge the noted risk.

Resolution: No change to the draft.

1. 68.12: In Figure 9-664, simply crossing out “variable” doesn’t give the reader any idea about the sizing of the field. The following text merely says that 12.11 and 12.12 use it. Given that any element includes the first three fields in Figure 9-664, we recommend saying something about the length or deleting the figure altogether.

The comment is referring to figure that renames FILS Wrapped Data element to Wrapped Data element to reflect its used outside of FILS protocol(s). Clause 9 lists all the elements, so it makes sense to keep the figure. It appears that the issue has been addressed by keeping the

‘variable’ label for the variable length Wrapped Data field has been restored in draft 4.0 – and perhaps is sufficient to address this comment.

The usage of the element is specified in other clauses and not in clause 9.

Resolution: No change to the draft.

1. 68.21: For bit 9, it might be worth mentioning what should be done if a STA does not support Secure RTT Measurement exchange.

The comment is valid in that the Notes could have an Otherwise clause to cover the case when describing the bit in RSNXE capabilities



Resolution: Revise

TGaz Editor: Change the text in cell for Bit 9 - p71.20 Table 9-321 as follows

A STA sets the Secure RTT Supported field to 1 if it supports Secure RTT Measurement exchange as defined in 11.21.6.4.2.1.6 (Secure measurement exchange protocol for EDMG STAs). Otherwise, the field is set to 0.

1. 87.20-21: This sentence would seem to indicate that an ephemeral public key and finite cyclic group ID might not be present. This does not align with 12.12.3.2 and other portions of 12.12, which expects an ephemeral key to be present.

In some of the error cases, the group and the public key need not be present in the PASN parameters element. For example, as stated in the draft p219.3

*If the processing status is REFUSED\_TEMPORARILY, the AP constructs and includes in the second PASN frame.*

*— 9.4.2.303 (PASN Parameters Element) with Comeback Info field with time for the peer to retry the operation. The Cookie Length field is set to nonzero if a cookie is being returned to the non-AP STA, otherwise it is set to 0. The Cookie field is optionally set. The Control field in the element is set appropriately to indicate the presence or absence of fields in the element.*

Resolution: No change to the draft.

1. 114.24: It says, “In an infrastructure BSS, association is required.” But this isn’t necessarily true when an infrastructure BSS is only used for location services.

Fair comment, but there is no change proposed by 11az draft here. In the containing subclause, there is mention of being able to send or receive Class 3 frames which require association. There are also other frames and services that can be used in an Infrastructure BSS without requiring association e.g., pre-association discovery services. Perhaps the phrase …required can be qualified to apply to Class 3 frames, so location and other pre-association services can be included.

Resolution: Revise

TGaz Editor: change the text p116.24 as follows

Association is not applicable in an IBSS. In an infrastructure BSS, association is required to exchange Class 3 frames. In a PBSS, association is optional. (#2582) APs and PCPs do not initiate association.

1. 125.5: Clauses 12.6.19 and 12.6.20 make no mention of PTKSA establishment.

Agreed. The section and their numbers need revising as they refer to PMF…

Resolution: Revise

TGaz Editor: replace the paragraph and the Note below p127.3

If the ISTA and the RSTA are associated, the PTKSA is established as defined in 12.6.19 3 (Protection of Robust Management Frames), and 12.6.20 (Robust management frame selection 4 procedure).

NOTE—The MFPC and MFPR capabilities define if the PTKSA between RSTA and ISTA is established or not.

With the following

If the ISTA and the RSTA are associated, a PTKSA is established as defined in 12.6.1.1.6 (PTKSA), 12.7.6 (4-way handshake), 12.11.2 (FILS authentication protocol), and 13.5 (FT Protocol). RSN capabilities MFPC and MFPR determine if the PTKSA between RSTA and ISTA is established or not.

1. 125.8: We suggest moving the note above the previous paragraph and dropping “NOTE—“.

Addressed by changes for item 6 above.

Resolution: Accept. No further changes required.

1. 125.8: s/the PTKSA/a PTKSA/

Addressed by changes for item 6 above.

Resolution: Accept. No further changes required.

1. 174.2: The figure needs some fixing near “Error”. The arrowhead is located between the two lines in the ladder diagram, which is confusing.

The figure has been fixed and updated in draft 4.0. Some of the terminology has also changed. It appears the figure 11-37r (Error recovery non-TB ranging measurement exchange using secure LTF) is clearer now.

Resolution: revised in draft 4.0. No futher changes required.

1. 199.3: s/signals/is set to 1 to signal/

Agree – ‘set to 1’ is missing.

If the bit is not actively being set, then use equal to. Perhaps this should be a revise

Resolution: Revise

TGaz Editor, please replace the paragraph p202.11 by the below text.

In a protected unicast management Action frame, Bit 4 of the Key ID octet equals 1 if the frame is a Protected Fine Timing Frame – see Table 9-51 (Category values). In other protected unicast frames, the bit is reserved.

1. 211.3: This clause could be much clearer in initially describing the options it offers for the different ways that PASN may be employed. Instead, the reader is left to intuit that there are different options from the salient points list on page 211, lines 9-17. For example, lines 9-10 leaves the reader assuming a PMKSA is implied, but lines 15-17 contradict that. We recommend starting with an enumeration of the PASN “modes” and then providing salient points for each of the modes.

There aren’t really that many modes – there are basically a few high-level aspects that could be covered earlier perhaps. Those would be whether mutual authentication is provided, whether PMKSA is established for the base AKM via tunneling. It would also help to clarify that use of PMK caching etc. is optional and that provides for mutual authentication (lines 9-10) – and lines 15-17 indicate that there is no mutual authentication already.

Resolution: Revise.

TGaz Editor: replace the paragraphs p214.3 to p214.9 (one paragraph and second one upto the first bullet, preserving the rest of the bullets, italics are used to highlight the changes) with the following

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 a non-RSNA protocol when there is no PMKSA and the corresponding (#2138) Base AKM used with it. *The protocol supports PTKSA establishment with and without mutual authentication and allows for a PMKSA to be established for certain Base AKMs by tunneling the Base AKM protocol messages.*

PASN is primarily intended for use in Infrastructure networks for a STA and an AP to establish a PTKSA using a three-message authentication frame exchange. Some salient aspects of this protocol are:

— *Where available,* it leverages a cached PMK for a Base AKM or already specified mechanisms for a Base AKM to establish the PMKSA from which the PASN PTKSA is derived. *Such a PTKSA provides mutual authentication.*

1. 211.4: Use of “AKM” here might be better served by “AKM selector value” or “AKM suite type”. See usage in REVmd D5.0, page 1109 or page 2613, respectively. We recognize that the base standard is also inconsistent in its usage.

This point is debatable – as selector value and suite type are too specific and implementation oriented. The description is referring to the AKM protocol (AKMP) and not the specific value assigned to the protocol – which an implementation can use the specificiation to translate into. However, it appears that the currently defined acronym AKMP is better suited in the contexts the acronym AKM is used in the PASN section. Consequently, we should replace the use of abbreviation AKM with AKMP in the PASN section 12.12

Another alterative is to leave this alone as there are many other instances where AKMP is meant where AKM is used etc.

e.g.,

*PMKSA for an AKMP, termed Base AKMP* instead of *PMKSA for an AKM, termed Base AKM*

Resolution: Revise.

TGaz Editor: replace the use of the abbreviation AKM with AKMP in 12.12 (PASN security negotiation) including the figure 12-55a—PASN authentication

1. 211.12: A PMKSA is not itself an RSNA protocol. A PMKSA might be established with an AKMP that has a key management type listed as an RSNA type in Table 9-151.

Agree. There is some confusion between the use of SA and protocol. The sentence ..is an RSNA protocol.. is not adding much – it has already been said in the introduction to 12.12 earlier – that PASN is an RSNA protocol when it relies on a PMKSA.

Resolution: Revise.

TGaz Editor: replace the bullet item p214.10 (also, note the use of AKMP instead of AKM)

— It exchanges ephemeral public keys to provide PFS and derive the PTKSA keys, using a PMKSA if one exists *and is an RSNA protocol*. The PMKSA corresponds to the Base AKM and consequently PASN AKM shall not be used in the RSNE of an (re)association request.

With the following

— It exchanges ephemeral public keys to provide PFS and derive the PTKSA keys, using a PMKSA if one exists. The PMKSA corresponds to the Base AKMP and consequently PASN AKMP shall not be used in the RSNE of an (re)association request.

1. 213.21: Would it make sense to provide a minimum value for a PTKSA lifetime? At some point, a PTKSA must have a lifetime so short that the STA will not have sufficient time to perform location determination with the AP for which the PMK lifetime is close to expiry.

I think min and max for lifetimes would be implementation specific. I am inclined not to specify the lifetimes in the specification – as any such values would need to consider all the uses for the PTKSA; it would be best to leave this to the implementation/higher level entities that understand the purpose of setting up the SA. Certification programs typically test this, at least implicitly.

We don’t have a minimum association lifetime; it is also application dependent.

Resolution: No change to the draft.

1. 214.22: This is the first PASN authentication frame. What validation has been performed that could have failed? Is it the original validation of the second PASN authentication frame that contained the Comeback Info field? If so, please make that clear.

Not sure how the validation got there – there is no validation when the first frame is being composed.

The first PASN Authentication frame (see 9.3.3.11 (Authentication frame format)) of the exchange is constructed as follows:

— …

— Including 9.4.2.303 (PASN Parameters Element) with the wrapped data format, chosen finite cyclic group ID, and the ephemeral public key. Comeback fieldshall either be absent or set to the cookie length and the cookie received from the AP if the authentication is being retried. Comeback After subfield shall not be present in the Comeback field. The Control field in the element is set appropriately to indicate the presence or absence of fields in the element. (#5019) *If the validation fails, unless a processing status of REFUSED\_TEMPORARILY is being returned, the processing status is set to INVALID\_PARAMETERS.* (#5086).

Resolution: Revise

TGaz Editor: Please remove the sentence re: validation in highlighted italics above p217.25 from the draft.

1. 216.5: The statement “The Cookie field is optionally set.” can be implied from the previous sentence, but could be more logically tied to the Cookie Length.

Okay, makes sense. Note: this is the case where the AP is returning a cookie to the non-AP STA.

Resolution: Revise

TGaz Editor: Please replace the bullet item p219.5 as follows

— 9.4.2.303 (PASN Parameters Element) with Comeback Info field with time for the peer to retry the operation. The Cookie Length subfield is set to nonzero if a cookie is being returned to the non-AP STA, otherwise it is set to 0. The Cookie subfield contains *the cookie if the Cookie Length subfield is non-zero*. The Control field in the element is set appropriately to indicate the presence or absence of fields in the element.

1. 216.11: Prior to deriving the PTKSA, the AP should derive the Diffie-Hellman shared secret (DHss) needed in the derivation of the PTKSA. The PTKSA derivation clause (12.12.7) does not actually spell out how this is done, but REVmd D5.0 on page 2702 shows an example of suitable text.

Agree, that is implied as DHss is used in the derivation of PTKSA (…) and it is a standard operation but does not hurt to specify the step.

Resolution: Revise

TGaz Editor: Please add the following bullet below before p219.13 before ..deriving the PTKSA

— perform the group’s scalar-op (see 12.4.4.1 (General)) with the non-AP STA’s ephemeral public key and its own ephemeral private key to produce an ephemeral Diffie-Hellman shared secret, DHss.

1. 216.18: The text says “PMKID count of 1”, and this seems correct, but Figure 12-55a (212.17) shows “PMKID[0..n]”. The figure should show “PMKID[0..1]” for authentication step 2.

Agreed. There would only be one PMKID in the response from the AP – authentication step 2 – p215 Figure 12-55a PASN authentication

Resolution: Accept.

TGaz Editor: Please change the figure as suggested.

1. 217.6: An active attacker could set the Status Code field to FAIL and cause early termination of the protocol. This would be similar to mangling the MIC except that the MIC is not checked until 217.32.

This needs some thought, but in general this is related to processing of the second PASN authentication frame by the non-AP STA. There is no real security state to validate the message and protection against denial of service is nearly impossible.

This is not an issue with PASN per se, but a general issue. For example, in the base spec (11me 0.2), SAE protocol processing in the committed state is similar. It needs to process the received commit message – the alleged response to its commit message – and there is no way to validate it. Ditto w/ 802.11 association that happens prior to security being available to protect the association messages. SAE can be attacked with a burst of SAE commit messages (with specific invalid status codes) in response to a SAE commit

In the case of PASN, if the status code is not a success, there is no requirement for the AP to engage in key derivation and include a proper MIC in the message.

In general, protection against active DoS attacks is not a security goal for PASN or many other mechanisms currently in the 802.11 standard. However, the general approach to handling an error status that cannot be securely confirmed is to drop without futher effect and let the protocol timeout.

There is a standard ‘AuthenticateFailureTimeout’ input to MLME that determines if the authentication fails. Perhaps we can drop the error frame without further effect and add a statement that PASN will be terminated if the authentication does not complete within the timeout.

It is not clear there is any point sending a failure PASN Auth Frame 3 from non-AP STA if validation of Auth Frame 2 fails. The AP expects a MIC and cannot do much with it.

Resolution: Revise

TGaz Editor: please change the paragraphs starting from p220.9 – p221.13 as follows

Once the processing completes, the AP sends the second PASN frame to the non-AP STA. If the processing status returned in the frame was not SUCCESS, the AP shall terminate PASN authentication protocol. (#5086) Upon receiving the second PASN frame, the non-AP STA

— Validates the Status Code field is SUCCESS~~. Otherwise, if the field is not REFUSED\_TEMPORARILY, PASN authentication shall be terminated.~~

— Validates the PASN Parameters element. ~~If the Parameters are not valid, PASN authentication shall be terminated.~~ If Status Code field was REFUSED\_TEMPORARILY and Comeback Info field in the parameters specifies a comeback time, the STA may retry PASN authentication after the specified time with the specified cookie (if any).

— Validates RSNE to ensure it is well formed and the AKM, Pairwise Cipher, MFPC and MFPR capabilities, No Pairwise bit, and Group Ciphers fields (of length 0) are set as expected. ~~Otherwise the STA shall terminate the PASN authentication protocol exchange.~~

— Validates that finite cyclic group indicated in PASN Parameters element is supported (present in dot11RSNAConfigDLCGroupTable). (#1032)

— Verifies that the public key as specified in 5.6.2.3 of NIST SP 800-56A R2.

— Verifies that a PMKSA named via a PMKID in the RSNE exists for the specified Base AKM, or the Base AKM is set PASN AKM or Base AKM data exists in the frame to allow a PMK to be established. If Base AKM is equal to PASN AKM, verifies that dot11NoAuthPASNActivated is set to true.

— Extracts any Base AKM specific data and processes it according to the behaviour described in a later subclause specific to the AKM; e.g., 12.12.4 (PASN authentication with FILS shared key).

— Locates the PMKSA, which might include waiting for Base AKM specific processing to complete.

— Derives the PTKSA; see 12.12.7 (PTKSA derivation with PASN authentication)

— If dot11RSNAOperatingChannelValidationActivated is true and the peer STA’s RSNE indicated OCVC capability, it validates that an OCI element is present and the Channel information in the element matches current operating channel parameters (see 12.2.9 (Requirements for Operating Channel Validation)). Otherwise, if there is a mismatch, processing status is set to OCI\_MISMATCH. (#5374, #TC1030r1)

— Computes the MIC as specified in 12.12.7.1 (MIC computation for PASN second frame) and verifies it to be the same as the MIC provided in the MIC element.

If ~~MIC~~ validation fails, the non-AP STA shall discard the frame and terminate ~~the processing of the frame with no further effect~~ further protocol processing but may retry PASN authentication later.

~~If other validation fails, the STA begins the construction the third PASN frame as follows:~~

~~— 9.4.1.1 (Authentication Algorithm Number field) set to 7 (PASN Authentication)~~

~~— 9.4.1.2 (Authentication Transaction Sequence Number field) set to 3~~

~~— Status code indicating the failure~~

~~— 9.4.2.118 (A MIC element) with MIC computed as specified in 12.12.8 (MIC computation for PASN third frame) . (#5370)~~

Otherwise…

TGaz Editor: Please add the following text at the end of §12.12.3 - p222.25

If the authentication is not successful within the AuthenticateFailureTimeout, the STA shall terminate PASN authentication and delete the PASN authentication state.

1. 217.27: The STA should compute the Diffie-Hellman shared secret (DHss) prior to deriving the PTKSA, as discussed above (see 216.11).

Agreed, similar to the AP case.

Resolution: Revise

TGaz Editor: Please add the following bullet before p220.33 (the corresponding non-AP STA case)

— perform the group’s scalar-op (see 12.4.4.1 (General)) with the AP’s ephemeral public key and its own ephemeral private key to produce an ephemeral Diffie-Hellman shared secret, DHss.

1. 218.21: The second half of this sentence (key installation) should only occur if the status code being sent was success.

Agreed, needs clarification. The keys are not installed in the failure case.

Resolution: Revise

TGaz Editor: modify paragraph p221.29-30 as follows

Once the processing complete, the non-AP STA sends the third PASN frame to the AP. *If a success status was indicated*, it installs the temporal key derived using the MLME-SETKEYS.request primitive.

1. 218.23: The STA inserts a status code into the third PASN frame indicating success or failure. Upon receipt of the third PASN frame, the AP, however, does not check the status code.

Agreed. Status code needs to be checked.

Resolution: Revise

TGaz Editor: Add the following bullet before p221.32

— Validates that the Status Code field is SUCCESS.

1. 220.29-31: Since these are called as out as little endian are the others big endian?

There is no such implication. Certain fields are called out in the spec, although 802.11 follows little endian conventions (§ 9.2.2 Conventions) in general. However, certain fields (esp. related to IETF protocols, SAE exchanges – integer encoding etc.) are big endian in the spec. A bit of redundancy does not hurt.

Resolution: No change to the draft.

1. 222.6: If computational costs are a concern, in the case that the AKM suite selector value is other than
   1. PASN, which employs a fixed PMK, it might be worth considering sacrificing the PFS. In place of the DHss, S-Ephemeral Pub || A-Ephemeral Pub would serve a similar purpose to the nonces in other versions of PTK generation within the base specification.

I think technology and the devices have evolved to a point where using public key mechanisms for authentication protocols and providing protection such as PFS is not a big concern.

If we were to allow no-PFS option – that would be yet another mode – in general, fewer modes is better.

We prefer not to introduce this additional option.

Resolution: No change to the draft.

1. 222.6, 222.20-27: The meaning of KDF-HASH-NNN would be more approachable if it were simply written as is done on page 2650 of REVmd D5.0. Another similar approach is shown in IEEE 802.11az D3.1 on pages 207, lines 6-22.

No change to semantics but would not hurt to be consistent.

Perhaps also some reorganization to describe function arguments first, then the function and then the output…

Resolution: Revise

TGaz Editor: replace p225 clause 12.2.7 PTKSA derivation with PASN authentication as follows

**12.12.7 PTKSA derivation with PASN authentication**

For PTKSA key derivation, the inputs to the PRF are the PMK of the PMKSA, a constant label and a concatenation of non-AP STA’s MAC address, AP’s BSSID and the DH shared secret from the ephemeral exchange.

*PTK* = 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; see 8 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 the string “PMKz” padded with 28 0s.

NOTE—The PMK for the derivation can come from a cached PMKSA for the AKM or from the PMKSA established with PASN by tunneling Wrapped Data or Authentication frames. (#3825)

DHss is the shared secret derived from the PASN ephemeral key exchange encoded as an octet string (12.4.7.2.2 (Integer to octet string conversion)).

KDF-HASH-NNN is the key derivation function defined in 12.7.1.6.2 (Key derivation function (KDF)) using the hash algorithm defined for the Base AKM; see Table 9-151 (AKM suite selectors). When there is no Base AKM, the hash algorithm is selected based on the pairwise Cipher Suite provided in the RSNE provided by the AP in the second PASN frame. SHA-256 is used as the hash algorithm, except for the ciphers 00-0F-AC:9 and 00-0F-AC:10 for which SHA-384 is used.

*NNN is the Bits required for KCK, TK and KDK depending on the pairwise cipher and whether a KDK is derived.*

*PTK is composed of the Key Confirmation Key (KCK), Temporal Key (TK) and the Key Derivation Key (KDK) which are derived as follows*

*KCK = L(PTK, 0, 256)*

*KCK is the first 256 bits of the PTK*

*TK = L(PTK, 256, TK\_Length\_Bits)*

*TK is the transient key whose length is the same as a key for the pairwise cipher in RSNE provided by the AP in the second PASN frame. This length is 16 octets for all ciphers, except for the ciphers 00-0F-AC:9 and 00-0F-AC:10 for which it is 32 octets.*

*KDK = L(PTK, 256 + TK\_Length\_Bits, KDK\_bits)*

*The KDK is of bit length KDK\_bits which has the value 256 if a KDK is derived (see 12.7.1.3 (Pairwise Key Hierarchy)) or 0 otherwise.*

KDK shall be derived if dot11SecureLTFImplemented is true and the peer STA has indicated Secure LTF support capability in its advertised Extended Capabilities.

The Key ID in the PTKSA (see 12.6.1.1.6 (PTKSA)) resulting from PASN authentication shall be 3 0. (#1457)

1. 222.18: clarify the meaning of “higher layer security”.

Okay. Revised part of item 25; higher layer security is replaced by whether KDK is derived or not; whether or not KDK is derived is described in a related PTKSA section.

Resolution: Revise. No further changes required.

1. 222.27: If a KCK is 32 octets and a TK is 16 or 32 octets, then with a KDK of either 256 bits or 0 bits, totals of 640 and 768 are not always guaranteed.

Okay. Revised part of item 25; removed the numbers that can easily be incorrect...

Resolution: Revise. No further changes required.

1. 222.28-29: It might make more sense to move this sentence to 222.19 if this is what is meant by higher layer security, otherwise there’s a contradiction between the two statements.

Okay. Agreed. Moved the KDK related text to near where KDK is described. Revised as part of item 25

Resolution: Revise. No further changes required.