IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 802.11  Pre-Association Security Negotiation for 802.11az | | | | |
| Date: 2018-03-07 | | | | |
| Author(s): | | | | |
| Name | Company | Address | Phone | Email |
| Nehru Bhandaru | Broadcom Ltd. | 250 Innovation Drive, San Jose CA 95134 USA | 408-391-2159 | [nehru.bhandaru@broadcom.com](mailto:ganesh.venkatesan@intel.com) |
| Thomas Derham | Broadcom Ltd. | 16340 West Bernardo, San Diego,California,US,92127 |  | thomas.derham@broadcom.com |

**Abstract**

This submission contains an authentication protocol, termed **PASN**, that allows a PTKSA to be established before 802.11 association exchange. The motivation for such a protocol originates from TGaz functional requirements [2] (see TGaz R38), and a high level scheme outlined in TGaz SFD [3] (see Security section 6, item 13). Such a protocol can be used to derive protection for 11az Negotiation and Measurement exchanges, leveraging Protected Management Frames (PMF) and other mechanisms much like the case where there is an 802.11 association followed by a PTKSA derivation. Once PTKSA is available, how it is applied to achieve protection of 11az specific exchanges is not the subject of this proposal.

Proposed changes are relative to *IEEE P802.11-REVmd™/D1.0, February 2018 [1].*

# Document History

**r00** – Initial Revision

r01 – Update with feedback from presentation at march IEEE

r02 – Update with additional feedback – see discussion

r03 – Additional review and update – see discussion

# References

[1] Draft P802.11REVmd D1.0 - <http://www.ieee802.org/11/private/Draft_Standards/11md/Draft%20P802.11REVmd_D1.0.pdf>

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

[3] TGaz Specification Framework - <https://mentor.ieee.org/802.11/dcn/17/11-17-0462-13-00az-11-az-tg-sfd.doc>

[4] Pre-Association Security Negotiation for 11az - <https://mentor.ieee.org/802.11/dcn/17/11-17-1737-00-00az-pre-association-security-negotiation-for-11az.pptx>

[5] Frame protection for 11az - <https://mentor.ieee.org/802.11/dcn/17/11-17-1776-01-00az-frame-protection-for-11az.pptx>

[6] Defense against multi-channel MITM attacks via Operating Channel Validation - <https://mentor.ieee.org/802.11/dcn/17/11-17-1807-03-000m-defense-against-multi-channel-mitm-attacks-via-operating-channel-validation.docx>

# Document Conventions

Suggested changes are specified as follows

* Red for editorial instructions
* Strikethrough for text to be deleted
* Blue for new proposed text
* Figures or changes to existing figures are described black and white or any other color.
* Black for existing text
* Green for reviewer or other notes

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

# Discussion - Topics to Cover

* Clause for PASN
  + Protocol description
    - Base AKM
    - With existing PMK and PMKID (any AKM)
    - PMK derivation with FILS and SAE
  + Protocol figure
  + Authentication frame fields/information elements
* PASN AKM indication in RSNIE
* Frame filtering
* 12.2.4 – RSNA establishment
* PASN parameters element
* Auth frame format
* Ephemeral public key encoding
* MLME authenticate request, confirm, indication, response
* 12.2.4 says – only Open system authentication and FT in an RSNA..
* Whether association can proceed with PASN authentication – perhaps related to above.
* AP and STA may choose to drop the frame or respond with a status code
* Including PASN frame 1 contents when generating MIC for frame 2 and verifying
* Update to integrity and encryption algorithms
* MIB – activated, supported
* Use only one AKM and derive keys
* RSNE validation in M2, M1 validation in M3
* PASN AKM may not be used in association messages.
* PASN Test Vectors
* SAE Support
* FT Support
* Operating Channel Information confirmation in PASN (Future)
* SA Query

# Discussion - Background

As noted earlier, related to PASN, TGaz requirements [2] include the following

*“TGaz R38The 11az positioning protocol shall have at least one secured mode that meets all of the following security requirements in the unassociated state*

1. *Authentication - Mutual authentication of initiator and responder (provided there is a prior security context established).*
2. *Encryption Algorithm - The cryptographic cipher combined with various methods for encrypting the message\* used in 11az-positing protocol.*
3. *Key Management - Create, distribute and maintain the keys.*
4. *Message Integrity - Ensures that the encrypted message\* has not been tampered with.*

*(\* Message refers to frame and/or field(s) within the frame.)*

*“*

*“TGaz R40 - 11az protocol shall support a mode where range integrity can be obtained without authentication and encryption protecting against type A adversaries. “*

*“TGaz R41 - The 11az protocol shall support shared key generation between Responding-Station and Initiating-Station when no previous shared secret has been pre-configured.”*

In support of the above requirement, TGaz agreed to the following in the SFD [3].

1. *The Pre-Association Security Negotiation (PASN) authentication shall allows authentication, encryption, and message integrity to be provided for selected 802.11 frames that require such protection.*
   1. *Whether such protection is required for a frame is determined by the security parameters negotiated for the exchange (e.g. 11az Protocol Negotiation) to which the frame belongs.*
      1. *An AP indicates PASN support by advertising a [TBD] PASN AKM in RSNIE that is included in Beacons and Probe Responses, and also in neighbor reports and reduced neighbor reports where supported.*
      2. *A non-AP STA selects use of PASN authentication based on the security requirements of features that need pre-association security. 11az protocol security for an un-associated STA requires PASN.*
   2. *A non-AP STA and an AP use 802.11 authentication frames with the Authentication algorithm number set to [TBD] (PASN Authentication) for the protocol exchange.*
      1. *A non-AP STA optionally, via PASN protocol, proposes to an AP a base AKM and PMKID(s) used to identify the PMK used for derivation of PTK for key confirmation and frame protection.*
      2. *An AP optionally, via PASN protocol, indicates to the non-AP STA, a base AKM and PMKID corresponding to the PMK used for derivation of PTK for key confirmation and frame protection.*
   3. *PASN protocol allows a FILS base AKM using shared key (See 12.12.2.3 of [1])*
      1. *PASN protocol allows any base protocol with a PMK – e.g. FT base AKM using PMKr1*
      2. *A non-AP STA and AP exchange ephemeral public keys to derive protection keys via PASN.*
   4. *The PTK for the exchange is derived from PMK, if any, and the shared secret from the ephemeral key exchange.*
   5. *If 11az measurement security for type A or type B attackers is required, the IEEE 802.11az negotiation protocol and measurement reports shall be integrity protected and encrypted for privacy.*
   6. *If 11az measurement security for type B attacker is required, the fields over which measurements are performed shall be protected (e.g. LTF sequence derived using the keys from PASN protocol).*

The requirement *e* above can be met by using MFP (Management Frame Protection) derived from PTKSA established using PASN.

Requirement *f* can be met by deriving an LTF protection key from PTKSA or part of PTKSA. While not specified here, one possibility derive PAIKM (Pairwise Application Input Key Material) along with PTKSA and use it as initial keying material (IKM in RFC5869) to derive the key for FTM or any other application. But this would require an extension to Pairwise Key Hierarchy.

Alternatively, a random key can be generated for the session and used as a PRNG seed along with SAC to derive the LTF sequence/protection.

Other requirements are (hopefully) addressed by this proposal.

# Discussion – MIB

Two MIB variables would be needed, at least – one to control whether PASN is allowed, and another to control whether PASN protocol without mutual authentication i.e. PMKSA is allowed.

# Discussion – Security Considerations

Note that an AP may need to perform non-trivial cryptographic operations (RNG generation, ECC multiply) when it receives a PASN authentication request. These only make things a bit harder as an attacker needs to capture the response and send another request with the token. Independent of PASN, an AP may have a mechanism to limit the number of concurrent authentications – as a resource limitation or protection against denial of service. Some support to address this might be useful.

There is no PMKSA setup for PASN AKM – the PMKSA is for the Base AKM; consequently, PASN AKM can not be used in RSNE used in the association request. It could be used to protect the association, if there is such a proposal in the future, perhaps.

PMKSA for a given Base AKM is used to derive PTKSA. I believe this should not violate security expected of those AKMs. Multiple PTKSAs are allowed to be derived from a given PMKSA.

PASN allows PMKSA to to be setup with SAE and FILS shared key.

Need a cryptographic RNG for generating private keys. Need at least one common group NIST P256/Group ID 19

Deauthentication deletes a PTKSA (4.5.4.3) established by PASN authentication

Denial or service and resource considerations dictate support for allowing an AP to tell the non-AP STA to comeback at a later time, and possibly provide a cookie to come back with (anti-clogging token)

PTKSA setup by PASN authentication

SA Query needs to be allowed – as are other Robust Action frames with PASN PTKSA

# Discussion – FT Support

Support only when FT hierarchy is already established. PMKR1Name is used as PMKID during PASN. Tunnelling FT authentication frame data (FTE, MDE) so that FT association can follow PASN authentication is cumbersome – from a protocol and specification standpoint - and probably not needed in PASN. This would need defining a FT authentication over PASN or something like that.

# Discussion – FILS Support

Either FILS PMKSA exists or PMKSA is setup for FILS Shared Key using Wrapped Data that is tunnelled to and from the authentication server. If FILS association is desired, FILS 802.11 authentication should be used, followed by a FILS association.

Tunnelling FILS authentication frame data (FILS Nonce, FILS Session…) so that FILS association can follow PASN authentication is cumbersome – from a protocol and specification standpoint - and probably not needed in PASN. This would need defining a FILS authentication over PASN or something like that.

# Discussion – SAE Support

SAE commit and confirm elements are tunnelled through wrapped data. Since the receiver of initial Commit message sends Commit and Confirm messages in back-to-back Authentication frames and transitions to Confirmed state, they can be combined and delivered as part of PASN second frame.

Not sure about supporting SAE anti-clogging tokens support when PASN uses SAE. It might reasonable to expect normal SAE authentication to be used followed by PASN authentication if it is desired to establish pre-association security.

# Discussion – RSN PSK (non-SAE) Support

PMKID corresponds to the PMK computed using SSID and Passphrase or the binary PMK without passphrase.

# Discussion – Mesh

No support for PASN authentication in a Mesh BSS (MBSS)

# Discussion – Additional Futures

It might be possible to complete association using a Base AKM without doing additional 802.11 authentication corresponding to Base AKM. That would be Base AKM over PASN – like FT over SAE etc.

Association frames can possibly be protected – guaranteeing integrity of other information – using PASN.

Work to confirm Operating Channel Information in security handshakes [6] could be integrated if and when accepted into 802.11md

Should supported ECC groups from dot11RSNAConfigDLCGroupTable be advertised by AP; currently seems to be trial and error with SAE, FILS and now PASN

FILS public key support may be added, but FILS shared key is going to be deployed first.

# Discussion – Feedback after r00

Is use of PTKSA confusing? Perhaps need a different name?

PTKSA for PASN needs a lifetime – addressed in r01 with a MIB variable dot11RSNAConfigPASNPTKSATimeout

# Discussion – Feedback after r01

Limit the unicast frames to the ones used for 11az such as negotiation, measurement reporting and perhaps PASN key update, etc. – This requires more thought. Perhaps we can leave this TBD for the initial draft and refine it later.

* Pros to protecting all of the robust frames
  + Since we have protection, it would be good to protect all such frames. One thing not clear is how the consumer (say using MLME) would know when to select the protection, but MLME use is device/implementation dependent typically. In this case we might need something to indicate there is protection possible – what is that? – so the application can select protecting using ‘protected’ option in MLME.
* Cons to protecting all of the frames
  + Frames are wide and varied – they may need or not need protection.
  + Risk of breaking some application or spec.
  + Are there issues due confusion with PTKSA with peers getting out of sync
    - At any time there can be only one PTKSA
    - PTKSA is deleted on association for non-AP STA and on association success on AP. (see p2023.42 11.3.5.3 AP or PCP association receipt procedures, 11.3.5.2 Non-AP and non-PCP STA association initiation procedures p2020.50
      * For FT, FT authentication derives a PTK and is used to protect the association.
    - Both sides has PASN PTKSA, or non-PTKSA or one side has a PTKSA and other has none
      * With PASN or non-PSANPTKSA on both sides SA query will verify the SA
      * If one side has none, then too
      * PASN on one side w/ non-PASN on other should not be possible – but an SA query will resolve that also.

Change the wording to include use of AKM 00-0F-AC, group addressed not allowed instead of cipher suite of length 0 because group addressing is not allowed. Agreed.

PMKSA for FILS base AKM and SAE base AKM established should be used only with PASN – might be good to restrict for now – this may be removed later if there are no security issues.

It would be easier to use the F1 Auth full body as oppose to the hash value?

After F1 is sent, only the hash needs to be stored. There is no need to keep the entire body. Simpler implementation.

Add note that MLME-SETKEYS is used to install keys after third frame is sent/received.

# Discussion - Feedback after r02

PTKSA section 12.6.1.1.6 needs to be updated to include PASN authentication

Timeout Interval element needs to be added to PASN authentication frames. It is already present in certain authentication frames – need to add it to PASN authentication related table

What is the key ID negotiated with PASN – set to 0 in PTKSA and also in RSNE exchanged in PASN authentication frames.

**Discussion - Feedback after r03**

Add PASN for Authentication Type in tables im MLME-Authenticate.xxxx

Add some rationale for frame filtering rule – leave it TBD for now.

PASN description uses AP-STA terminology, rather than ISTA and RSTA.

* ISTA and RSTA are terminology specific to FTM/11az
* PASN may be used in another context
* Infrastructure networks are the primary use case for 11az
* IBSS etc. have already security mechanisms, and there is no notion of pre-association
* Future public key (e.g. certificates) mechanisms may be added to PASN to support additional STA-STA scenarios if such use cases arise (separate submission)
* AP does not initiate 802.11 authentication protocol (but initiates 4-way handshake etc). Relaxing this causes more complications (fit in 802.11 architecture). AP’s typically do not initiate anything – perhaps they can send a trigger to have STA initiate (e.g. 11k/v)

--

Modify the definitions and acronyms to include PASN definition (3 p151.48)

pre-association security negotiation (PASN): A mechanism to establish security association and allow management frame protection prior to 802.11 association.

pre-association security negotiation STA: A station that implements pre-association security negotiation (PASN) and for which *dot11PASNActivated* is true

--

Modify 4.5.4.2 Authentication (p242.36) as follows

IEEE Std 802.11 defines ~~five~~(11ai) several IEEE 802.11 authentication methods: Open System authentication,

Shared Key authentication, FT authentication, (11ai) simultaneous authentication of equals (SAE), ~~and~~ FILS

authentication(11ai), and PASN authentication. Open System authentication admits any STA to the DS. Shared Key authentication relies on WEP to demonstrate knowledge of a WEP encryption key. FT authentication relies on keys derived during the initial mobility domain association to authenticate the stations as defined in Clause 13 (Fast BSS transition). SAE authentication uses finite field cryptography to prove knowledge of a shared password.

FILS authentication allows for faster connection to the network for FILS non-AP STAs by providing

authentication, association, and key confirmation information in an efficient number of frame exchanges

(see 4.10.3.6 (AKM operations using FILS authentication(11ai))).(11ai) PASN authentication allows management frame protection prior to 802.11 association by establishing a PTKSA using authentication frames. The IEEE 802.11 authentication mechanism also allows definition of new authentication methods.

An RSNA might support one or more of SAE authentication, FILS authentication, or PASN authentication ~~both(11ai)~~ . An RSNA also supports authentication based on IEEE Std 802.1X-2010, or preshared keys (PSKs) after Open System authentication. IEEE 802.1X authentication utilizes the EAP to authenticate STAs and the AS with one another. This standard does not specify an EAP method that is mandatory to implement. See 12.6.5 (RSNA policy selection in an IBSS(#59)) for a description of the IEEE 802.1X authentication and PSK usage within an IEEE 802.11 IBSS.

…

p243.23

SAE authentication is performed prior to association and a STA can take advantage of the fact that it can be

IEEE 802.11 authenticated to many APs simultaneously by completing the SAE protocol with any number

of APs while still being associated to another AP. RSNA security can be established after association using

the resulting shared key.

PASN authentication is also performed prior to association. It results in establishing a security association (PTKSA) that allows management frames to be protected prior to association. PASN authentication is used in an RSN for an infrastructure BSS when it is based on a PMKSA established by another RSN authentication protocol. Otherwise, it does not guarantee mutual authentication, and can be used as a non-RSN protocol in an infrastructure BSS.

--

Modify MLME-AUTHENTICATE.request (6.3.5.2 p310.2), add PASN to Authentication Type and add an entry for PASN authentication before VendorSpecificInfo

Content of FILS Authentication frame,(11ai)

Contents of PASN authentication frame,

VendorSpecificInfo

)



|  |  |  |  |
| --- | --- | --- | --- |
| Authentication Type | Enumeration | OPEN\_SYSTEM,  SHARED\_KEY,  FAST\_BSS\_TRANSITION,  SAE, FILS\_SHARED\_KEY  WITHOUT\_PFS,  FILS\_SHARED  KEY\_WITH\_PFS,  FILS\_PUBLIC\_KEY(11ai)  PASN | Specifies the type of authentication  algorithm to use during the  authentication process. |



|  |  |  |  |
| --- | --- | --- | --- |
| Contents of PASN authentication frame | Sequence of elements and fields | As defined in 12.xx.2.2 PASN Frame Construction and Processing | The set of elements and fields to be included in PASN authentication frames. Present if AuthenticationType indicates PASN authentication and *dot11PASNActivated* is true, otherwise not present. |

--

Modify MLME-AUTHENTICATE.confirm (6.3.5.3 p311), add PASN to Authentication Type and add an entry for PASN authentication before VendorSpecificInfo

Same instructions as the previous

--

Modify MLME-AUTHENTICATE.indication (6.3.5.4 p313) and add an entry for PASN authentication before VendorSpecificInfo

Same instructions as the previous

--

Modify MLME-AUTHENTICATE.response (6.3.5.4 p314), add PASN to Authentication Type and add an entry for PASN authentication before VendorSpecificInfo

Same instructions as the previous

--

Add PASN Parameters element to Authentication frame body in Table 9-39 (9.3.3.12 p818.41) before Last



|  |  |  |
| --- | --- | --- |
| ANA-PASN-Order | PASN Parameters | The PASN Parameters element is optionally present in PASN authentication frames as defined in Table 9-40 (Presence of fields and elements in Authentication frames |

--

Add PASN authentication to Table 9-40 Presence of fields and element in Authentication frames (p822.61)



|  |  |  |  |
| --- | --- | --- | --- |
| **Authentication algorithm** | Authentication transaction sequence number | Status code | Presence of fields 4 onwards |
| PASN authentication | 1 | Reserved | RSNE is present.  PASN Parameters element is present.  Timeout Interval element may be present.  Wrapped Data element is present if wrapped data format in PASN parameters element is non-zero and not reserved.  Fragment element may be present if any of the elements are fragmented. |
| PASN authentication | 2 | Status | If Status Code filed 0, then   * RSNE is present * PASN Parameters element is present. * Timeout Interval element may be present. * Wrapped data element is present if wrapped data format in PASN parameters element is non-zero and not reserved. * MIC element is present   Fragment element may be present if any of the elements are fragmented. |
| PASN authentication | 3 | Status | If Status Code field is 0, then   * PASN Parameters element is present. * Wrapped data element is present if wrapped data format in PASN parameters element is non-zero and not reserved. * MIC element is present   Fragment element may be present if any of the elements are fragmented. |

--

Add PASN to Authentication Algorithm Number field (9.4.1.1 p834.21)

Authentication algorithm number = ANA-AUTH-ALGO-1: PASN authentication

--

Modify table 9-144 AKM suite selectors by adding the PASN AKM (9.4.2.24.3 p1025.9)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 00-0F-AC | ANA-AKM-1 | PASN | PASN key management defined in 12.xx.yy | Defined in 12.xx.zz (Key establishment with PASN authentication) |

--

Modify Figure 9-519 to allow variable octet count for MIC field (p1198.54)



|  |  |  |
| --- | --- | --- |
| Element ID | Length | MIC |

Octets: 1 1 Variable

--

Change the name of FILS Wrapped Data element to just Wrapped Data element that could be used by non-FILS functionality (p1286.36)

9.4.2.186 ~~FILS~~ Wrapped Data element(11ai)

The ~~FILS~~ Wrapped Data element is used for the STA and AP to communicate data used by the ~~FILS~~

RSNA protocols ~~authentication algorithm~~. The format of the ~~FILS~~ Wrapped Data element is defined in Figure 9-645 (~~FILS~~

Wrapped Data element format(11ai)).

The Element ID, Length, and Element ID Extension fields are defined in 9.4.2.1 (General).

The ~~FILS~~ Wrapped Data field is the data used by the FILS authentication algorithm (see 12.12

(Authentication for FILS(11ai))) and PASN authentication algorithm (see 12.xx (Pre-Association Security Negotiation)).

--

Change FILS Wrapped Data to Wrapped Data in Figure 9-645 (p1286.45)

--

Change references to ‘FILS Wrapped Data’ to ‘Wrapped Data’ in the document

--

Add PASN Parameters Element at the end of 9.4.2 p1337.36

9.4.2.xx PASN Parameters Element

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Element ID | Length | Element ID Extension | Control | Wrapped Data Format | Comeback Info | Finite Cyclic Group ID | Ephemeral Public Key Length | Ephemeral Public Key |

Octets: 1 1 1 1 1 Variable 0 or 2 0 or 1 Variable

**Figure 9- XXE PASN Parameters Element format**

The Element ID, Length and Element ID Extension fields are defined in 9.4.2.1 (General).

The Control field indicates the presence of PASN Parameters Element subfields as follows

B0 B1 B2...B7

|  |  |  |
| --- | --- | --- |
| Comeback Info Present | Group and Key Present | Reserved |

Bits: 1 1 6

**Figure 9- XXCI PASN Parameters Element Control field**

Wrapped Data Format field indicates the format of data in Wrapped Data element included along with the PASN parameters element. The values defined for this format are:

0: No wrapped data

1: Fast BSS Transition Wrapped Data (See 12.xx.6 PASN Authentication with FT)

2: FILS Shared Key authentication without PFS Wrapped Data (See 12.xx.4 PASN authentication with FILS Shared Key)

3: SAE Wrapped Data (see 12.xx.5 PASN authentication with SAE)

Other: Reserved

The Comeback Info field, present only when indicated by the corresponding bit in the Control field, is of variable length and is formatted as follows

|  |  |  |
| --- | --- | --- |
| Comeback After | Cookie Length | Cookie |

Octets: 0 or 2 1 Variable

**Figure 9- XXCI PASN Parameters Element Control 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.xx.2.2 PASN Frame Construction and Processing).
* Cookie Length field is the length of the following Cookie
* Cookie (see 12.xx Pre-Association Security Negotiation)

The finite cyclic group field is used to indicate the group used in PASN authentication. It has the same semantics as the field Finite Cyclic Group field (9.4.1.42).

Ephemeral public key length field is the length in octets of the ephemeral public key.

Ephemeral public key field is the public key encoded using RFC 5480 conventions. Both compressed and uncompressed forms may be used.

The finite cyclic group, Ephemeral public key length and Ephemeral public key length fields are present if the corresponding bit is set in the Control field. Otherwise they are absent.

--

Modify the frame filtering rules to include robust management frames when PASN is used (11.3.3 p2016.22)

b) Class 2 frames

1) Management frames

i) Association Request/Response

ii) Reassociation Request/Response

iii) Disassociation

iv) TBD (subset or all) Unicast Robust Management Frames when PTKSA from PASN authentication exists

--

Allow PASN for Authentication at Originating STA (11.3.4.2 p2017.58)

4) For FILS authentication, the authentication mechanism described in 12.12 (Authentication for

FILS(11ai)). An AP may provide estimated association response latency to a non-AP STA

using the Association Delay Info field in the Association Delay Info element (9.4.2.173 (Future

Channel Guidance element)). The value of the Association Delay Info field shall be larger than

dot11HLPWaitTime(11ai) .

5) For PASN authentication, the authentication mechanism described in 12.xx (Pre-Association Security Negotiation) in an infrastructure BSS.

--

Allow PASN for Authentication at Destination STA (11.3.4.3 p2018.44)

g) Upon receipt of an MLME-AUTHENTICATE.response primitive, if the ResultCode is SUCCESS,

the MLME shall transmit an Authentication frame that is constructed using the appropriate

procedure in 12.3.3.2 (Open System authentication), 12.3.3.3 (Shared Key authentication), 13.5 (FT

protocol) or 13.6 (FT resource request protocol), with a status code of SUCCESS, and the state for

the originating STA shall be set to State 2 if it was in State 1.

h) If PASN authentication is being used, the MLME shall issue an MLMEAUTHENTICATE.indication primitive to inform the SME of the authentication request, and the SME shall execute the procedure described in 12.xx (Pre-Association Security Negotiation)

--

Modify SA Query procedures to allow for SA Queries with PASN PTKSA exists (11.13 SA Query Procedures p2117.20)

A STA that supports the SA Query procedure and receives an SA Query Request frame shall respond with

an SA Query Response frame unless either of the following are true:

— the STA is not currently associated to the STA that sent the SA Query Request frame and there is no PTKSA established by PASN authentication.

— the STA has sent a (Re)Association Request frame within dot11AssociationResponseTimeOut but

has not received a corresponding (Re)Association Response frame

--

Modify the following in the types of security associations (12.6.1.1.1 General, p2371.33)

PTKSA: A result of a successful 4-way handshake, FT 4-way handshake, (11ai) FT authentication

sequence, ~~or~~ FILS authentication, or PASN authentication.

--

Modify 12.6.1.1.6 PTKSA as follows (p2373.30)

The PTKSA results from a successful(11ai) 4-way handshake, FT 4-way handshake, FT protocol, (11ai) FT

resource request protocol, ~~or~~ FILS authentication(11ai), or PASN authentication . This security association is also bidirectional. PTKSAs, except those established using PASN authentication, have the same lifetime as the PMKSA or PMK-R1 security Association, whichever comes first. PTKSAs for PASN authentication have a the minimum of the lifetime of the PMKSA used and the timeout negotiated, if any, during PASN authentication.

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,(11ai) or FT Protocol authentication, or FILS authentication(11ai), or PASN authentication .

…

--

Modify the Integrity and Key-wrap algorithms table as follows (p2414) and add the note after the table



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 00-0F-AC:ANA-AKM-1 | *See note below* | - | - | - | - | - | - |

Note: Keys derived using PASN AKM are not used to protect EAPOL frames

--

Add the following text to allow PASN as an RSNA authentication method (12.2.4 p2308.49)

e) In order to associate with a PCP in a PBSS, an (#136) RSNA capable STA’s SME establishes an

RSNA with the PCP following the RSNA establishment steps in an infrastructure BSS in

accordance with method a) and b) above, as appropriate, with the PCP taking the role of the AP.

f) When an (#136) RSNA capable STA chooses not to associate with a peer in a PBSS, its SME

establishes an RSNA with the peer following the RSNA establishment steps in an IBSS in

accordance with method c) or d) above, as appropriate, with the caveat that the RSNA authentication

and key management algorithm is executed only once between the peers.

g) If an RSNA uses PASN authentication, an RSNA capable STA’s SME establishes an RSNA as described in 12.xx (Pre-Association Security Negotiation)

…

--

Modify text to allow PASN authentication algorithm in an RSNA (p2308.61)

Only Authentication frames with the authentication algorithm equal to Open System authentication ~~or~~ , FT

authentication, or PASN authentication may be used within an RSNA. An RSNA STA shall not associate if Shared Key authentication was invoked prior to RSN association.(#59)

--

Modify the text as follows to allow PASN authentication for an infrastructure BSS (12.6.10 p2383.35)

12.6.10.1 General

When establishing an RSNA in a non-FT environment or during an FT initial mobility domain association, a

STA shall use IEEE 802.11 SAE authentication, FILS authentication(11ai), or Open System authentication

prior to (re)association. It may also use PASN authentication without (re)association.

--

Add a subclause 12.xx to section 12. Security with the following text. (s12.xx, before p2480)

# 12.xx Pre-Association Security Negotiation

# 12.xx.1 General

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

It is primarily intended for use in Infrastructure networks for a STA and an AP to establish a PTKSA using a three message 802.11 authentication frame exchange. Some salient aspects of this protocol are

* 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.
* 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.
* A PTKSA may be derived using ephemeral keys without mutual authentication at the discretion of the protocol peers i.e. it may derive a PTKSA without a PMKSA and is thus a non-RSNA protocol.
* Key establishment and key confirmation is accomplished using 802.11 authentication frames.
* It allows a PMKSA to be established by tunnelling FILS shared key and SAE protocol data as part of PASN authentication frame contents.
* Authentication is initiated by a non-AP STA
* PASN authentication does not derive GTKSA or IGTKSA
* A provision to allow APs to request the peer to come back later based on resource constraints or other conditions. The comeback cookie can also be used as an anti-clogging token for better denial of service protection on the AP.

The security of PASN depends on assumptions similar to those for FILS and SAE authentication. In particular

* It assumed that both the STAs have in common at least one cyclic group from the dot11RSNAConfigDLCGroupTable that is used to select the ephemeral key exchange parameters.
* Protection is provided against downgrade attacks that may alter security parameters (e.g. RSNE, Group, Hash Function) used in the negotiation.
* Denial of service protection is not guaranteed.
* A cryptographic RNG is available to generate FFC private keys

There is no protection for group addressed frames using PASN authentication. Like any PTKSA, the PTKSA established via PASN authentication shall be deleted when the Base PMKSA expires. Similarly, Deauthentication (4.5.4.3) results in deletion of PASN PTKSA. For the purpose of interoperability, a STA shall support group 19, an ECC group defined over a 256-bit prime order field.

# 12.xx.1 Discovery of PASN capable AP

An AP indicates it is capable of performing PASN authentication by including the PASN AKM as part of the RSNE included in Beacon and Probe Response frames. When PASN AKM is advertised, the AP shall also include at least one additional AKM in the RSNE unless it allows PTKSA derivation without authentication using the ephemeral keys exchanged during PASN authentication.

# 12.xx.2 Key establishment with PASN authentication

# 12.xx.2.1 Overview

This subclause defines the procedure for establishing a PTKSA and the corresponding shared keys between a PASN capable STA and AP. The PASN frame sequence used is depicted in Figure 12-xx 1 (PASN Authentication). It consists of three Authentication frames with Authentication Algorithm Number field (9.4.1.1) set to ANA-AUTH-ALGO-1 (PASN Authentication) and the corresponding Authentication Transaction Sequence Number field (9.4.1.2) set to 1, 2, and 3 respectively.

AP/

Authenticator

STA/

Supplicant

Beacon (RSNE(PASN AKM, BASE AKM))

802.11 Authentication (1, PASN, RSNE(Base AKM, PMKID[0..n]), S-Ephemeral Pub, PASN Parameters, Base AKM Data-1)

802.11 Authentication (2, PASN, RSNE(Base AKM, PMKID[0..n]), A-Ephemeral Pub,PASN Parameters, Base AKM Data-2, MIC)

802.11 Authentication (3, Base AKM Data-3, MIC)

**Figure 12-xx 1- PASN Authentication**

A

A non-AP STA shall use the first PASN frame to initiate PASN authentication with a PASN capable AP by providing RSNE that contains Base AKM parameters including a list of PMKIDs (possibly empty), STA’s ephemeral public key, PASN parameters, and additional Base AKM specific data, if any.

The PASN capable AP processes the first PASN frame and responds with a second PASN frame. The processing depends on the Base AKM as specified in a later section of the document. If the processing is successful, the AP includes RSNE that contains chosen Base AKM parameters including the chosen PMKID (possibly none), AP’s ephemeral public key, PASN parameters, and additional Base AKM specific data, if any. The frame is also integrity protected with a MIC using the Key Confirmation key (KCK) as described in 12.xx.7 (MIC Computation for PASN Second Frame). If the processing fails, the AP may return an appropriate status in the Status Code field (9.4.1.9) of the frame without including additional information or silently drop the frame.

The non-AP STA receives and processes the second PASN frame and responds with the third PASN frame. Again the processing depends on the Base AKM as specified later in the document. If the processing successful, the STA responds with an integrity protected PASN third frame; Otherwise it may return an appropriate status in the Status code field (9.4.1.9) of the frame or silently drop the frame. The frame is integrity protected with a MIC using the Key Confirmation key (KCK) as described in 12.xx.7.2 (MIC Computation for PASN Third Frame).

A successful PASN exchange results in establishment of PTKSA using the ephemeral keys and PMK from the Base AKM (if any). 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 PASN PTKSA will be deleted by the SME of a non-AP STA or an AP when

* its lifetime expires
* Upon initiation of (re)association on a non-AP STA or successful association by the peer on an AP (see 11.3.5.2 (Non-AP and non-PCP STA association initiation procedures) and 11.3.5.3 (AP or PCP association receipt procedures)).

# 12.xx.2.2 PASN Frame Construction and Processing

If non-AP STA chooses to initiate PASN authentication, it first selects the following authentication parameters

* Base AKM from among AKMs advertised by the AP if RSNA authentication is desired. Otherwise, if *dot11NoAuthPASNAllowed* is *true,* Base AKM chosen is the PASN AKM, indicating that PTKSA is to be established without mutual authentication i.e. without a corresponding PMKSA.
* Pairwise Cipher Suite to use for the PTKSA that is being setup
* Finite cyclic group from the dot11RSNConfigDLCGroup table that is at least of the security strength provided by the Base AKM and Cipher Suites.

With the chosen finite cyclic group, the STA generates an ephemeral (random) private key, and uses the selected group’s scalar operation (see 12.4.4.1 (General)) with the private key to generate its ephemeral public key. If PMKSA caching is used, it generates a list of PMKSA identifiers, and then obtains Base AKM specific data.

It then composes an RSNE in which

* Chosen AKM and Pairwise cipher included
* MFPC and MFPR (see 9.4.2.24.4 RSN capabilities) in the RSN capabilities field are set to 1.
* No Pairwise bit is set to 0
* PMKID count and PMKID list set corresponding to PMKSA identifiers
* Group Data Cipher Suite and Group Management Cipher Suite set to 00-0F-AC:7, indicating that group addressed traffic is not allowed.
* Extended key ID for individually addressed frames (bit B13) shall be 0.

The first PASN Authentication frame (see 9.3.3.12 (Authentication frame format)) of the exchange is constructed as follows

* Authentication Algorithm Number field (9.4.1.1) set to ANA-AUTH-ALGO-1 (PASN Authentication)
* Authentication Transaction Sequence Number field (9.4.1.2) set to 1
* Including the constructed RSNE
* Including PASN Parameters Element (9.4.2.xx) with the wrapped data format, chosen finite cyclic group ID, and the ephemeral public key. Comeback information included and is set to the cookie length and the cookie received from the AP if the authentication is being retried.
* Including Wrapped Data Element (9.4.2.186) corresponding to Base AKM data, if any. This element may be fragmented using mechanism specified in 10.28.11 (Element fragmentation) as necessary.
* Optionally including Timeout Interval element (TIE) (9.4.2.48) with Timeout Interval Value set to dot11RSNAConfigPASNPTKSATimeout and Timeout Interval Type set to 2 (key lifetime interval).

The non-AP STA sends the first PASN frame to the AP.

Upon receiving the first PASN frame, the AP

* 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 processing status is set appropriately (see Table 9-52 Status Codes - to INVALID\_RSNE, INVALID\_RSNE\_CAPABILITIES, INVALID\_AKMP, INVALID\_GROUP\_CIPHER, INVALID\_PAIRWISE\_CIPHER, UNSUPPORTED\_RSNE\_VERSION)
* Validates that finite cyclic group indicated in PASN parameters element is supported (present in dot11RSNConfigDLCGroup table). Otherwise processing status is set to UNSUPPORTED\_FINITE\_CYCLIC\_GROUP
* Validates PASN Parameters. If resources or other policy does not permit PASN authentication (in general or for the peer) at this time, but will be permitted later, processing status is set to REFUSED\_TEMPORARILY. Otherwise if Comeback Info field is present and AP is expecting the field, it shall validate the field. It is outside the scope of this specification as to the validation mechanism used.
* Verifies that the public key as specified in 5.6.2.3 of NIST SP 800-56A R2. If verification fails, the AP shall terminate PASN authentication protocol.
* Verifies that a PMKSA named via a PMKID in the RSNE exists for the specified Base AKM, or the Base AKM is set to PASN AKM or Base AKM data exists in the frame to allow a PMK to be established. If Base AKM is set to PASN AKM, verifies that *dot11NoAuthPASNAllowed* is set to *true*. Otherwise processing status is set to REFUSED.
* Extracts any Base AKM specific data, including reassembly (10.28.12 Element defragmentation) as necessary, and processes it according to the behavior described in a later subclause specific to the AKM (e.g. 12.xx.4 PASN authentication with FILS shared key). This may include waiting for Base AKM related processing to complete before advancing the PASN protocol state. If the Base AKM related processing indicates an error or there is no PMKSA at the end of such processing, processing status is set to REFUSED

The AP begins the construction the second PASN frame as follows

* Authentication Algorithm Number field (9.4.1.1) set to ANA-AUTH-ALGO-1 (PASN Authentication)
* Authentication Transaction Sequence Number field (9.4.1.2) set to 2
* Status code indicating the processing status

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

* PASN Parameters Element (9.4.2.xx) with Comeback Info field with time for the peer to retry the operation. The Cookie Length field is set to non-zero 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.

Otherwise, if the validation is successful, the AP

* With the chosen finite cyclic group, generates an ephemeral (random) private key, and uses the selected group’s scalar operation with the private key to generate its ephemeral public key.
* Derives the PTKSA (see 12.xx.7 PTKSA derivation with PASN authentication)
* Constructs and includes in the second PASN frame
  + An RSNE that contains
    - Chosen Base AKM and Pairwise cipher
    - MFPC and MFPR (see 9.4.2.24.4 RSN capabilities) in the RSN capabilities field are set to 1.
    - No Pairwise bit is set to 0
    - PMKID count of 1 and PMKID list corresponding to the PMKID from the PMKSA
    - Group Data Cipher Suite and Group Management Cipher Suite set to 00-0F-AC:7, indicating that group addressed traffic is not allowed.
    - Extended key ID for individually addressed frames (bit B13) shall be 0.
  + PASN Parameters Element (9.4.2.xx) with the wrapped data format, chosen finite cyclic group ID, and the AP’s ephemeral public key. The Control field in the element is set appropriately to indicate the presence or absence of fields in the element.
  + Wrapped Data Element (9.4.2.186) corresponding to Base AKM data to be returned to the non-AP STA, if any. This element may be fragmented using mechanism specified in 10.28.11 (Element fragmentation) as necessary.
  + Optionally including Timeout Interval element (TIE) (9.4.2.48) with Timeout Interval Value set to dot11RSNAConfigPASNPTKSATimeout and Timeout Interval Type set to 2 (key lifetime interval).
  + A MIC element (9.4.2.118) with MIC computed as specified in 12.xx.7.1 (MIC computation for PASN second frame)

Once the processing complete, the AP sends the second PASN frame to the non-AP STA.

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 Comeback Info field in the parameters specifies a come back 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 dot11RSNConfigDLCGroup table).
* 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 set to PASN AKM, verifies that *dot11NoAuthPASNAllowed* is set to *true*.
* Extracts any Base AKM specific data and processes it according to the behavior described in a later subclause specific to the AKM (e.g. 12.xx.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.xx.7 PTKSA derivation with PASN authentication)
* Computes the MIC as specified in 12.xx.7.1 (MIC computation for PASN second frame) and verifies it to be the same as the MIC provided in the MIC element

If the validation fails, the non-AP STA shall terminate the PASN authentication protocol. Otherwise the STA begins the construction the third PASN frame as follows

* Authentication Algorithm Number field (9.4.1.1) set to ANA-AUTH-ALGO-1 (PASN Authentication)
* Authentication Transaction Sequence Number field (9.4.1.2) set to 3
* Status code indicating success
* Constructs and includes in the third PASN frame
  + PASN Parameters Element (9.4.2.xx) with the wrapped data format. Public Key information and Comeback Info fields shall not be present. The Control field in the element is set appropriately to indicate the presence or absence of fields in the element.
  + Wrapped Data Element (9.4.2.186) corresponding to Base AKM data to be returned to the non-AP STA, if any. This element may be fragmented using mechanism specified in 10.28.11 (Element fragmentation) as necessary.
  + A MIC element (9.4.2.118) with MIC computed as specified in 12.xx.7.2 (MIC computation for PASN third frame)

Once the processing complete, the non-AP STA sends the third PASN frame to the AP and installs the temporal key derived using MLME-SETKEYS.request primitive to install the new key.

Upon receiving the third PASN frame, the AP

* Validates PASN Parameters
* Computes the MIC as specified in 12.xx.7.1 (MIC computation for PASN third frame) and verifies it to be the same as the MIC provided in the MIC element
* Extracts any Base AKM specific data and processes it according to the behavior described in a later subclause specific to the AKM (e.g. 12.xx.4 PASN authentication with FILS shared key). Base AKM processing may indicate an error

For the third PASN frame, if validation fails, or Base AKM processing indicates an error, the AP shall terminate the PASN authentication. Otherwise the AP installs the temporal key derived using MLME-SETKEYS.request primitive to install the new key.

If the AP terminated the PASN authentication silently or by sending a failure Status Code to the non-AP STA, other than a temporary refusal, it shall delete all of the PASN authentication state.

If the non-AP STA receives an Authentication frame that does not contain Authentication Algorithm Number field set to ANA-AUTH-ALGO-1 (PASN Authentication) once it has sent the first PASN frame, it shall abandon the PASN authentication in progress. Likewise, if the AP receives an Authentication frame that does not contain Authentication Algorithm Number field set to ANA-AUTH-ALGO-1 (PASN Authentication) once it has received the first PASN frame, it shall abandon the PASN authentication in progress.

Note that, because of frame filtering rules (11.3.3 Frame filtering based on STA state), the protection using the key installed by PASN authentication is only used to protect management frames, as applicable.

# 12.xx.4 PASN authentication with FILS Shared Key

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.

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 (9.4.2.186 (Wrapped Data element)).

In the first PASN frame, it 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. Wrapped data shall be absent in the third PASN frame.

The resulting rMSK is used in PMKSA derivation (12.12.2.5.2 (PMKSA key derivation with FILS authentication)) where

* SNONCE is set to the non-AP STA’s ephemeral public key encoded as an octet string (12.4.7.2.4 (Element to octet string conversion))
* ANONCE is set to AP’s ephemeral public key encoded as an octet string (12.4.7.2.4 (Element to octet string conversion))
* 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))

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.

# 12.xx.5 PASN authentication with SAE

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. When PMK caching is used PASN authentication relies on the PMKSA already established by SAE protocol. Without PMK caching, wrapped data shall be present in first, second and third PASN frames and is constructed as follows.

The non-AP STA instantiates an SAE protocol instance (12.4.8 SAE finite state machine) and provides an *Initiate* event. The PASN authentication process receives an Authentication frame with a Commit message (12.4.7.4 Encoding and decoding of SAE commit messages) from the SAE protocol instance. It then sets the Wrapped Data in the first PASN frame to be the Authentication frame body.

The AP upon receiving the first PASN frame, delivers the frame to SAE for processing via the SAE parent process managed by the SME. If the PASN authentication process receives, from the SAE protocol instance, a failure indication, PASN authentication shall be abandoned. Otherwise PASN authentication process receives Authentication frames with a Commit message ((12.4.7.4) and a Confirm message (12.4.7.5 Encoding and decoding of SAE Confirm messages), in that order, as the SAE protocol instance transitions to the *Confirmed* state.

The AP sets the Wrapped Data in the second PASN frame, as shown in the figure below, to be the concatenation of

* Two octet length of first Authentication frame body in little endian order
* First Authentication frame body with a Commit message.
* Two octet length of second Authentication frame body in little endian order
* Second Authentication frame body with a Confirm message.

|  |  |  |  |
| --- | --- | --- | --- |
| First Authentication Frame Length | First Authentication Frame Body | Second Authentication Frame Length | Second Authentication Frame Body |

Octets: 2 Variable 2 Variable

**Figure 12-zzz PASN Wrapped Data format with SAE**

The non-AP STA upon receiving the second PASN frame, delivers the Commit and Confirm messages from the AP to the SAE protocol instance. If the PASN authentication process receives, from the SAE protocol instance, a failure indication, PASN authentication shall be abandoned. Otherwise PASN authentication process receives an Authentication frame with a Confirm message (12.4.7.5 Encoding and decoding of SAE Confirm messages) as the SAE protocol instance transitions to the *Confirmed* state followed by a transition to the *Accepted* state (Since it has received both the Commit and Confirm messages from the peer).

The non-AP STA sets the Wrapped Data in the third PASN frame to be the Authentication frame body with a Confirm message. The AP upon receiving the third PASN frame, delivers the Authentication frame from the Wrapped Data to the SAE protocol instance. Either the instance accepts the frame and transitions to the *Accepted* state or PASN authentication shall be abandoned and the SAE protocol instance destroyed.

Not all aspects of SAE protocol are supported by the above tunnelling mechanism. In particular, the tunnelled protocol may fail if the SAE state machine desire

* Anti-Clogging Tokens
* Retries of Commit and Confirm messages

In such cases, SAE authentication may be first used to establish an SAE PMKSA.

The PMKSA is then used in PTKSA derivation for PASN authentication. If establishment of SAE PMKSA fails, PASN authentication shall be abandoned. An SAE PMKSA so established shall be used only to establish PTKSAs that are negotiated via PASN authentication.

# 12.xx.6 PASN authentication with FT

This subclause specifies aspects of PASN authentication when one of FT AKMs 00-0F-AC: [3, 4, 13, 19] is used as the Base AKM.

PASN authentication with FT relies on FT key hierarchy already being established via the FT initial mobility domain association (13.4.2). PASN protocol messages carry FT PMKR1Name as the PMKID, and the PASN PTKSA is established like any other Base AKM.

Wrapped data shall be optionally present in PASN first frame. When present it shall contain a set of elements that include RSNE (9.4.2.24), MDE (9.4.2.46), and FTE (9.4.2.47) as specified for the first message of FT authentication sequence (13.8.2 FT authentication sequence: contents of the first message)

Wrapped data shall be optionally present in the second PASN frames but shall be present if wrapped data was present in the first PASN frame. When present it shall contain a set of elements that include RSNE (9.4.2.24), MDE (9.4.2.46), and FTE (9.4.2.47) as specified for the second message of FT authentication sequence (13.8.3 FT authentication sequence: contents of the second message).

Wrapped data shall be absent in the third PASN frame. The elements in the wrapped data are used for additional validation FT security parameters as being used in PASN authentication.

# 12.xx.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.

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

where

KCK is the key confirmation key of length 32 octets.

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.

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 9-144 (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.

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))

and NNN is the bits required for KCK and TK which is either 384 or 512 depending on the pairwise cipher.

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.

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

# 12.xx.7 MIC Computation with PASN authentication

# 12.xx.7.1 MIC Computation for PASN second frame

The MIC field of the MIC element in the PASN second frame is set by the AP to first MMM octets of

HMAC-HASH (KCK, BSSID||SPA||*Beacon RSNE*||*Frame Data*)

where

HASH is the hash algorithm used in PTKSA derivation (12.xx.7 PTKSA derivation with PASN authentication)

KCK is the key confirmation key for the PASN PTKSA

BSSID is the BSSID for the AP’s BSS.

SPA is the MAC address of the non-AP STA, the transmitter of the first PASN frame.

*Beacon RSNE* is the RSN element sent in the Beacons transmitted by the AP.

*Frame Data* is the body of the PASN second frame including the MIC element with the octets in the MIC field of the MIC element set to 0.

MMM is half of the output length in octets for the hash function used i.e. 16 or 24 octets for SHA-256 and SHA-384 respectively.

*Beacon RSNE* is included in the MIC computation so that a downgrade attack with forged RSNE Beacons will result in a MIC mismatch and thus PASN authentication failure.

# 12.xx.7.2 MIC Computation for PASN third frame

The MIC field of the MIC element in the PASN third frame is set by the non-AP STA to first MMM octets of

HMAC-HASH (KCK, SPA||BSSID||*F1 Auth*||*Frame Data*)

where

HASH is the hash algorithm used in PTKSA derivation (12.xx.7 PTKSA derivation with PASN authentication)

KCK is the key confirmation key for the PASN PTKSA

SPA is the MAC address of the non-AP STA, the transmitter of the first PASN frame

BSSID is the BSSID for the AP’s BSS

*F1 Auth* is the HASH of body of the PASN first frame

*Frame Data* is the body of the PASN third frame including the MIC element with the octets in the MIC field of the MIC element set to 0.

MMM is half of the output length in octets for the hash function used i.e. 16 or 24 octets for SHA-256 and SHA-384 respectively.

--

Add PICS entry for PASN protocol p3242.52



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PCxx | Pre-Association Security Negotiation | 12.xx (Pre-Association Security Negotiation) | CFInfraSTA:O | Yes - No - N/A - |

--

Add MIB support for PASN authentication as follows (p3449.59)

dot11VHTExtendedNSSBWCapable TruthValue,

dot11FutureChannelGuidanceActivated TruthValue,

dot11FILSActivated(11ai) TruthValue(11ai),

dot11S1GOptionImplemented(11ah) TruthValue(11ah),

dot11PASNActivated,

dot11NoAuthPASNAllowed

--

Add MIB definition for dot11PASNActivated as follows (p3487.4)

*dot11PASNActivated* OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This is a control variable.

It is written by an external management entity or the SME. Changes

take effect for the next MLME-START.request primitive or MLME JOIN.

request primitive.

This attribute indicates whether or not PASN authentication is enabled."

DEFVAL {false}

::= { dot11StationConfigEntry ANA-MIB-1 }

*dot11NoAuthPASNAllowed* OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This is a control variable.

It is written by an external management entity or the SME. Changes

take effect for the next MLME-START.request primitive or MLME JOIN.

request primitive.

This attribute indicates whether or not PASN without mutual authentication is allowed."

DEFVAL {false}

::= { dot11StationConfigEntry ANA-MIB-2 }

Add support for PASN PTKSA timeout at the end of dot11RSNAConfig Table p3498.32

dot11RSNAConfigNumberOfGTKSAReplayCounters Unsigned32,(#59)

dot11RSNASAERetransPeriod Unsigned32,

dot11RSNASAEAntiCloggingThreshold Unsigned32,

dot11RSNASAESync Unsigned32 ~~,~~

dot11RSNAConfigPASNPTKSATimeout}

Add MIB support for PASN PTKSA Timeout at the end of dot11RSNAConfig table p3504.53

dot11RSNAConfigPASNPTKSATimeout OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)

UNITS "seconds"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This is a control variable.

It is written by an external management entity.

Changes take effect as soon as practical in the implementation.

The time for which the derived PTKSA derived from PASN authentication is valid”

DEFVAL { 3600 }

::= { dot11RSNAConfigEntry XX43 }

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- \* End of dot11RSNAConfig TABLE

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--

Add placeholder for PASN test vectors in Annex J (p4131.65)

**J.yy PASN Test Vectors**

TBD