IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assortment of SA ballot comments | | | | |
| Date: 2023-11-14 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Jouni Malinen | Qualcomm Technologies, Inc. |  |  | jouni@qca.qualcomm.com |
|  |  |  |  |  |

Abstract

This document proposes comment resolutions to the following REVme/D4.0 initial SA ballot comments:

CID 6085, 6088, 6419, 6483

This also includes discussion for follow comments with the expectation of additional TG feedback being needed before a resolution can be proposed:

CID 6086, 6087

# CID 6419 (GEN)

Clause Number: 2

Comment:

It is not clear whether the section 5.3 fix in https://theory.stanford.edu/~jcm/papers/fp09-he.pdf actually been included in the spec (the intro suggests there was agreement it should/would be)

Proposed Change:

Add "Analysis of the 802.11i 4-Way Handshake" to Clause 2

Proposed Resolution:

REVISED - Incorporate changes under the “Proposed changes for CID 6419” section of <this doc>.

### Discussion

The comment is apparently referring to this part in the paper’s introduction:

We provided these basic attacks and repairs to the 802.11 TGi; the third repair in Section 5.3 was adopted. Although the 802.11i documentation was left unchanged so that the ratification would not be delayed, the repair will be added once the documentation is updated [Walker, email communication, June 30, 2004].

Section 5.3 of the paper (“Nonce Re-use”) describes this repair as follows:

The third repair is to eliminate the intermediate states on the supplicant side. Specifically, the supplicant can re-use the values of SNonce until a legitimate handshake is completed and a shared PTK is achieved between the supplicant and the authenticator. In other words, the supplicant does not update its nonce responding to each received *Message 1* until *Message 3* is received and verified.

IEEE P802.11-REVme/D4.0 shows RSNA Supplicant key management state machine to assign “SNonce = Random” in the AUTHENTICATION state which is entered when “an SSID is to be authenticated”. In other words, this is not when receiving any 4-way handshake message 1, but at the start of the exchange. That would seem to imply that the SNonce value is to be re-used over multiple message 1 retries. However, 12.7.6.2 (4-way handshake message 1) could be interpreted to indicate that a new SNonce would be generated for each received message 1.

There does not seem to be any clear justification for adding a reference to the paper into Clause 2 as suggested in the comment. However, it would seem reasonable to make the standard clearer in when the SNonce value is updated.

### Proposed changes for CID 6419

**12.7.6.2 4-way handshake message 1**

*Modify 12.7.6.2 as shown (REVme-D4.1 P3036 L9):*

Processing for PTK generation is as follows:

The Authenticator sends message 1 to the Supplicant at the end of a successful IEEE 802.1X authentication, after (re)association completes for a STA that has authenticated with SAE or for which PSK authentication is negotiated, when a cached PMKSA is used, or after a STA requests a new key. On reception of message 1, the Supplicant determines whether the Key Replay Counter field value has been used before with the current PMKSA. If the Key Replay Counter field value is less than or equal to the current local value, the Supplicant discards the message. Otherwise, the Supplicant:

a) Generates a new nonce SNonce if no SNonce has yet been generated for this 4-way handshake. The same SNonce is reused within this 4-way handshake until a valid message 3 has been received.

b) Derives PTK, the derived PTK including the Key derivation key (KDK) if WUR frame protection is being negotiated.

c) Constructs message 2.

# CID 6483 (SEC)

Clause Number: 12

Comment:

The same SNonce should be reused until a valid M3 has been received, to avoid DoS attacks (see https://theory.stanford.edu/~jcm/papers/fp09-he.pdf )

Proposed Change:

Make this recommendation (as a "should")

Proposed Resolution:

REVISED - Incorporate changes under the “Proposed changes for CID 6419” section of <this doc>.

### Discussion

Practically same comment as CID 6419 except with a different proposed change. 12.7.6.2 does not use shall/should language, so descriptive language seems more appropriate in that context. The Supplicant state machines already imply this behavior, so there does not seem to be justification for having to make this any more complex than that. The resolution text for CID 6419 can be used to address this comment as well.

# CID 6085 (SEC)

Clause Number: 12.4.7.3 Page: 2834 Line: 6

Comment:

Use of a plaintext SAE password identifier can reveal personal identifier of a STA (or user) when per-device or per-user passwords are used even if the STA is using MAC address randomization. This is undesired and can reduce likelihood of deployment of SAE password identifiers which can result in deployment issues for future IEEE 802.11 deployments like EHT/MLO where PSK cannot be used, but multiple passwords might be needed.

Proposed Change:

Incorporate changes from the latest revision of https://mentor.ieee.org/802.11/dcn/23/11-23-1236-01-000m-hpke-protected-sae-password-identifiers-for-privacy.docx

Proposed Resolution:

REVISED - Incorporate changes under the “Proposed changes” section of https://mentor.ieee.org/802.11/dcn/23/11-23-1236-02-000m-hpke-protected-sae-password-identifiers-for-privacy.docx.

### Discussion

1236r2 was updated to be on top of REVme/D4.1. Based on earlier SPs, this does not introduce a new AKM suite or use ANQP. Instead, a protected mechanism was added to manage public key changes.

# CID 6086 (SEC)

Clause Number:

Comment:

There are various use cases where multiple ESSs (i.e., different SSIDs) are used to connect to the same IEEE 802 LAN. This could be needed, e.g., when setting up a separate ESS with older RSN parameters to allow misbehaving deployed STAs to get connected. It would be helpful to the user experience if the protocol would be extended to allow the AP to provide information to non-AP STAs about which SSIDs are connected to the same LAN, so that the STA UI could guide the user in selecting the most appropriate SSID (e.g., pick the ESS where SAE is enabled in a case where a separate ESS is set with only PSK AKMs for legacy support).

Proposed Change:

Introduce a new information element that an AP can advertise to point to a preferred ESS which connects to the same LAN.

Proposed Resolution:

TBD

### Discussion

There has been significant amount of recent discussion related to deployment issues due to some deployed STAs being unable to connect with an AP that advertises a mix of allowed AKMP suites and cipher suites. This comment discussed one potential approach for addressing such interop issues with the use of multiple SSIDs. However, that is not the only potential approach, and we could discuss this issue with a wider scope to see if there is consensus on any mechanism that the standard should promote for this or if there were some more generic changes that could help with deployment challenges. Some of that discussion is covered here in the context of CID 6086 while additional consideration is available in CID 6087.

The multi-SSID case mentioned in the comment would apply to both multiple independent BSSs and to the cases where such BSSs were part of a Multiple BSSID set. The protocol extensions for these cases might be able to share the same design (e.g., a new information element that provides ESS selection guidance in general), but there might also be benefit of a more Multiple BSSID specific mechanism since the related BSSs being advertised in a single Beacon frame can avoid some potential attacks where the new mechanism could be used to try to direct STAs to select a BSS controlled by an attacker instead of a BSS from the actual operator of the network.

Multiple BSSID approach has also been proposed to be used with a single SSID, i.e., with the transmitted BSSID and one or more of the nontransmitted BSSIDs corresponding to a BSS that is part of the same ESS. This would have a benefit of the full set of BSSs being part of a single SSID (a.k.a. “network name” for the user). This type of a configuration might already be allowed (at least no explicit requirement disallowing this have been identified in REVme/D4.0, but there has been some speculation about P802.11be/D4.0 potentially implying this is not allowed). In any case, this combination can be unexpected by existing non-AP STA implementations that support Multiple BSSID procedure. Furthermore, there are no requirements defined for a non-AP STA to select the best BSS from a Multiple BSSID set, e.g., based on the available RSNE options to prefer SAE over PSK if SAE is used in a nontransmitted BSS with the same SSID as the transmitted BSS is used with PSK.

The current Multiple BSSID mechanism has been defined to require HT/VHT/HE/S1G/EHT Capabilities and Operation element to have the same contents for all BSSs in the set. This is somewhat problematic for the potential uses of Multiple BSSID procedure (both with the same or different SSIDs) for avoiding interoperability issues with deployed STAs. The likely configuration for those uses would be to configure the BSS corresponding to the transmitted BSSID to include older RSN parameters, e.g., PSK AKM, CCMP-128 pairwise cipher, and MFPC=0) while the BSS(s) corresponding to a nontransmitted BSSID would include newer RSN parameters, e.g., SAE AKM(s), GCMP-256 and CCMP-128 as pairwise ciphers, MFPC=1. While this is not yet of significant concern with the cases included in REVme/D4.0, this does get more challenging with addition of new amendments like P802.11be which when combined with the 6 GHz rules practically disallows use of PSK AKM for MLO associations and as such, would strictly speaking make the configuration of the BSS that corresponds to the transmitted BSSID be non-compliant in this type of use case: PSK-only and MFPC=0 on the transmitted BSS while still having to enable EHT/MLO in the Beacon frame and also having to enable beacon protection so that the BSS(s) corresponding to the nontransmitted BSSID would be able to use those capabilities with newer STAs.

Open items for TGme to figure out:

* Do we want to modify Multiple BSSID procedure to allow compliant implementation of legacy STA interoperable transmitted BSS (i.e., allow exception to things like HE/EHT/etc. Capabilities/Operations element to not apply for the transmitted BSS while they apply for nontransmitted BSSs, and to allow transmitted BSS to advertise MFPC=0 even when a nontransmitted BSS has MFPC=1 with beacon protection enabled?
* Do we want to provide recommendation or requirements on which BSS in a Multiple BSS set to select if there are multiple BSSs in use with the same SSID (e.g., prefer SAE over PSK)?
* Do we want to provide a Multi BSSID specific mechanism for an AP to guide STAs on different SSIDs/ESSs pointing to the same “network”/LAN and indicating how a STA should pick the best one for association, e.g., based on stronger security?
* Do we want to provide a generic mechanism for an AP to guide STAs on different SSIDs/ESSs pointing to the same “network”/LAN and indicating how a STA should pick the best one for association, e.g., based on stronger security? Keeping in mind that this is likely to have some challenges in avoiding new attack surface for making a STA connect to an attacker-controlled BSS?

### Proposed changes for CID 6086

**12.7.6.2 4-way handshake message 1**

*Modify 12.7.6.2 as shown (REVme-D4.1 P3036 L9):*

Abcd

# CID 6087 (MAC)

Clause Number: 9.4.2.23

Comment:

While RSNE was designed to be extensible, the last 20 years has shown that many vendors are unable to follow the extensibility rules described in the IEEE 802.11 standard and almost any extension to RSNE contents (e.g., adding another AKM Suite or using some of the previously unassigned RSN Capabilities) over the years has resulted in interop issues with deployed STAs. RSNXE was added to allow easier upgrade path in the future. Alas, some vendors have apparently managed to implement it incorrectly as well and that has resulted in yet additional future extensibility issues. There seems to be a need to allow less risky way of extending security of a BSS in a manner that does not result in deployed STAs becoming confused. This may need to be done by allowing RSNE and RSNXE to remain identical to a previously used snapshot while being able to use other elements to describe newly introduced mechanisms for newer STAs. While multiple ESSs/SSIDs with different RSN parameters could be used in some cases, it has significant issues from the viewpoint of airtime use and scalability. As such, a new mechanism that works with a single BSS would be beneficial.

Proposed Change:

Introduce new elements that can be used to override RSNE and RSNXE payload for newer STAs while maintaining RSNE and RSNXE with their old payload for older STAs.

Proposed Resolution:

TBD

### Discussion

While this issue is something that has been known for years, there has not been sufficient push to get the issue fixed or worked around. Since some of the cases are being forced on the surface with 6 GHz requirements and how those will end up impacting all bands when MLO is used, there has been significant recent interest on this topic. This comment is proposing one approach for working around the issues while still being able to use a single SSID and a single BSS to avoid extra complexity for the deployment and users; and also, to optimize airtime use (both Beacon frames and group-address Data frames). Some other potential approaches and proposed changes for them are discussed in CID 6086.

Open items for TGme to figure out:

* Do we want to do this with IEEE 802.11 defined elements within the standard? Or leave this for external consideration with vendor specific elements and externally defined rules to modify IEEE 802.11 behavior?
* If we want to do this, what would the payloads of the new elements look like? Identical to RSNE/RSNXE? A new design with limited set of possible combination?

### Proposed changes for CID 6087

**12.7.6.2 4-way handshake message 1**

*Modify 12.7.6.2 as shown (REVme-D4.1 P3036 L9):*

Abcd

# CID 6088 (SEC)

Clause Number: 12.2.2 Page: 2793 Line: 49

Comment:

There is now quite a few references to OWE and its behavior in the IEEE 802.11 standard. However, this is not sufficient to implement OWE without having to go through the combination of the IEEE 802.11 standard and IETF RFC 8110. This is unnecessarily complex and results in issues in maintaining OWE in the future. As an example of this, there is a known difference between RFC 8110 and deployed OWE implementations: https://www.rfc-editor.org/errata/eid6182. It would be good to get a single place that defines the actually implemented OWE and the best place for this would seem to be the IEEE 802.11 standard.

Proposed Change:

Merge in the key derivation and other protocol details from IETF RFC 8110 into IEEE 802.11 and updates these based on this errata: https://www.rfc-editor.org/errata/eid6182.

Proposed Resolution:

REVISED - Incorporate changes under the “Proposed changes for CID 6088” section of <this doc>.

### Discussion

IEEE P802.11-REVme/D4.0 actually already covers the main point of the IETF RFC 8110 referenced in the comment: Table 9-188 (AKM suite selectors) defines 00-0F-AC:18 to use the key derivation type defined in 12.7.1.6.2 instead of the one referenced in RFC 8110 (RFC5869). The main missing part in IEEE P802.11-REVme/D4.0 is the definition of the Diffie-Hellman Parameter element and description of how exactly it is used during association to derive a PMKSA. Inclusion of that element in (Re)Association Request/Response frames was already covered for the frame format.

It should also be noted that OWE was first proposed in IEEE 802.11 (TGmc) and as such, the changes were contributed to IEEE before the IETF document was published. See [https://mentor.ieee.org/802.11/dcn/15/11-15-1184-07-000m-owe.docx for the IEEE 802.11](https://mentor.ieee.org/802.11/dcn/15/11-15-1184-07-000m-owe.docx%20for%20the%20IEEE%20802.11) contribution. This document was used as a source for some of the changes proposed below.

### Proposed changes for CID 6088

**4.10 IEEE Std 802.11 and IEEE Std 802.1X-2020**

**4.10.3 Infrastructure functional model overview**

*Add a new subclause at the end of 4.10.3, i.e., just before the start of 4.10.4 (REVme-D4.1 P340 L39):*

**4.10.3.7 Opportunistic Wireless Encryption**

The following AKM operations are carried out when opportunistic encryption is accomplished using a Diffie-Hellman key exchange.

* The STA discovers the AP’s security policy indicating support for Opportunistic Wireless Encryption through passively monitoring Beacon frames or through actively probing. After discovery in a non-DMG infrastructure BSS, the STA performs Open System Authentication with the AP.
* The STA (re)associates with the AP and provides an ephemeral Diffie-Hellman public key in its (Re)Association Request frame. The AP responds with an ephemeral Diffie-Hellman public key in its (Re)Association Response frame.
* The STA and AP complete the Diffie-Hellman key exchange and generate a PMK.
* The 4-Way Handshake using EAPOL-Key frames is used, just as with IEEE Std 802.1X authentication, when an AS is present.

**9.3.3.6 Association Response frame format**

*Modify 9.3.3.6 / Table 9-65 as shown (REVme-D4.1 P723 L27):*

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| 10 | RSN | The RSNE is present if dot11FILSActivated is true or if performing OWE; otherwise not present. |

**9.3.3.8 Reassociation Response frame format**

*Modify 9.3.3.8 / Table 9-67 as shown (REVme-D4.1 P732 L48):*

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| 10 | RSN | An RSNE is present in a Reassociation Response frame if dot11FastBSSTransitionActivated is true, dot11RSNAActivated is true, and this frame is a response to a Reassociation Request frame that contained an FTE (i.e., part of a fast BSS transition in an RSN); or if dot11FILSActivated is true; or if performing OWE. Otherwise, not present. |

**9.4.2 Elements**

*Add a new subclause at the end of 9.4.2, i.e., just before the start of 9.4.3 (REVme-D4.1 P1532 L9):*

**9.4.2.175 Diffie-Hellman Parameter element**

The Diffie-Hellman Parameter element contains a Diffie-Hellman public value and an indicator of the finite cyclic group from which it was obtained. See Figure 9-XYZ (Diffie-Hellman Parameter element format).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element ID | Length | Element ID Extension | Group | Public Key |

Octets: 1 1 1 2 variable

**Figure 9-XYZ—Diffie-Hellman Parameter element format**

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

The Group field is a 16-bit unsigned integer that maps an identifying number from the “Group Description” registry maintained by IANA for IETF RFC 2409 (IKE) to a complete domain parameter set.

The Public Key is a Diffie-Hellman public key, an element in the group described by the domain parameter set indicated by the value in the Group field.

**12.6.8.3 Cached PMKSAs and RSNA key management**

*Modify 12.6.8.3 as shown (REVme-D4.1 P2995 L63):*

A STA might cache PMKSAs it establishes as a result of previous authentication. The PMKSA shall not be changed while cached. The PMK in the PMKSA is used with the 4-way handshake or FILS authentication to establish fresh PTKs.

If a STA in an infrastructure BSS has determined it has a valid PMKSA with an AP to which it is about to (re)associate, it performs Open System authentication to the AP, or performs FILS Authentication to the AP, or (only in the case of a DMG STA) does not perform IEEE 802.11 authentication. When Open System authentication is used or IEEE 802.11 authentication is not performed, it includes the PMKID for the PMKSA in the RSNE in the (re)association request. When FILS Authentication is performed, it includes the PMKID for the PMKSA in the FILS Authentication frame (see 12.11.2 (FILS authentication protocol)). When the PMKSA was not created using preauthentication, the AKM indicated in the RSNE by the STA in the (re)association request shall be identical to the AKM used to establish the cached PMKSA in the first place.

A cached PMKSA established using an AKMP for which the Authentication type column indicates FT authentication (see Table 9-190 (AKM suite selectors)) may be used in a subsequent FT Initial Mobility Domain Association using an identical AKMP. A cached PMKSA is not used in an FT authentication sequence.

If a cached PMKSA is used in FT Initial Mobility Domain Association, the cached MPMK is used to derive the PMK-R0 of a new FT key hierarchy (see 12.7.1.6 (FT key hierarchy)). The PMKID indicated by the STA in the (Re)Association Request frame and message 1 of the FT 4-way handshake (when FILS authentication is not used) or FILS Authentication frame (when FILS authentication is used) is the PMKID of the cached PMKSA as defined in 12.7.1.6.3 (PMK-R0) (i.e., not the PMKR0Name or PMKR1Name of the FT key hierarchy that was derived when the PMKSA was originally established). The PMKR1Name indicated in RSNE in messages 2 and 3 of the FT 4-way handshake (when FILS authentication is not used) or in (Re)Association Request and Response frames (when FILS authentication is used) is the PMKR1Name of the newly derived FT key hierarchy (see 13.4 (FT initial mobility domain association)).

Upon receipt of a (Re)Association Request frame with one or more PMKIDs following Open System authentication or (only in the case of a DMG AP) if IEEE 802.11 authentication was not performed, an AP checks whether its Authenticator has cached a PMKSA for the PMKIDs and whether the AKM in the cached PMKSA matches the AKM in the (Re)Association Request; and if so, it shall assert possession of that PMKSA by beginning the 4-way handshake after association has completed and shall include the PMKID in message 1 (see 12.7.6.2 (4-way handshake message 1)). If the Authenticator does not have a PMKSA for the PMKIDs in the (re)association request or the AKM does not match, its behavior depends on how the PMKSA was established. If SAE authentication was used to establish the PMKSA, then the AP shall reject (re)association by sending a (Re)Association Response frame with status code STATUS\_INVALID\_PMKID. Note that this allows the non-AP STA to fall back to full SAE authentication to establish another PMKSA. If IEEE 802.1X authentication was used to establish the PMKSA, the AP begins a full IEEE 802.1X authentication after association has completed.

Upon receipt of a FILS Authentication frame with one or more PMKIDs, an AP checks whether its Authenticator has cached a PMKSA for the PMKIDs, whether the AKM in the cached PMKSA matches the AKM in the FILS Authentication frame, and whether the PMK is still valid; and if so, it shall assert possession of that PMK by including the PMKID in the FILS Authentication frame sent in response (see 12.11.2.3.4 (AP construction of Authentication frame)). If the Authenticator does not have a PMK for the PMKIDs in the FILS Authentication frame or the AKM does not match, the AP may either reply with an EAP-Finish/Reauth packet to continue FILS Shared Key authentication option if the non-AP STA included sufficient information for that, or the AP rejects the authentication.

A non-AP STA may attempt PMKSA caching in conjunction with the OWE Authentication method by indicating AKMP 00-0F-AC:18 along with a PMKID in the RSNE and including a Diffie-Hellman parameter in the (Re)Association Request frame. If the AP has a PMKSA with a matching PMKID it shall not include a Diffie-Hellman parameter in its (Re)Association Response frame and not perform a Diffie-Hellman key exchange. If the AP does not have a PMKSA with a matching PMKID it shall perform OWE Authentication as if the PMKID was not present.

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

If the lifetime of a cached PMKSA expires, the STA shall delete the expired PMKSA.

If a STA roams to an AP with which it is preauthenticating and the STA does not have a PMKSA for that AP, the STA needs to initiate a full IEEE 802.1X EAP authentication.

**12. Security**

*Add a new subclause (and its subclauses) at the end of Clause 12, i.e., just before the start of Clause 13 (REVme-D4.1 P3099 L4):*

**12.14 Opportunistic Wireless Encryption**

**12.14.1 General**

The Opportunistic Wireless Encryption handshake is executed between a non-AP STA and an AP to establish a PMKSA using a simple Diffie-Hellman key exchange. The handshake does not provide true authentication of the non-AP STA or AP but does enable encryption. This handshake is designed for cases in which access control is either not necessary or can be handled outside of this standard.

**12.14.2 Opportunistic Wireless Encryption exchange**

The Opportunistic Wireless Encryption (OWE) exchange occurs during association of a non-AP STA and an AP that indicate the OWE AKM (9.4.2.23.3 AKM Suites) by placing a Diffie-Hellman Parameter element in the (Re)Association Request and Response frames, respectively.

First the non-AP STA chooses a group in which to perform the Diffie-Hellman exchange. It then generates a secret private key, *m*, by randomly choosing it such that 1 < *m* < *r*, where *r* is the (prime) order of the chosen group. It derives a public key, ***M***, from the private key and the generator from the chosen group, ***G***:

***M*** = scalar-op(*m*, ***G***)

Where scalar-op() is defined in 12.4.4 (Finite cyclic groups). The non-AP STA then constructs a Diffie-Hellman Parameter element by assigning the appropriate value from the IANA-maintained registry for the chosen group to the Group field and assigning the public key to the Public Key field using the conversion defined in 12.4.7.2.4 (Element to octet string conversion). The non-AP STA inserts the Diffie-Hellman Parameter element into its (Re)Association Request frame per 9.3.3.5 (Association Request frame format) or 9.3.3.7 (Reassociation Request frame format) and transmits the (Re)Association Request frame to the AP. If the non-AP STA chooses to do PMKSA caching it shall include a PMKID in the (Re)Association Request frame.

Upon receipt of the non-AP STA’s (Re)Association Request frame indicating OWE, the AP checks whether the STA is also attempting PMKSA caching. If so, and the AP has a PMKSA with a matching PMKID it shall proceed to the 4-way handshake as if it is not doing OWE, the STA’s Diffie-Hellman parameter is ignored. If it does not have a PMKSA with a matching PMKID it shall continue with OWE as if the PMKID was not present. The AP then checkes the Group field of the Diffie-Hellman Parameter element. If it is not an acceptable value, the AP shall reject association with the reason of UNSUPPORTED\_FINITE\_CYCLIC\_GROUP. Otherwise, it shall extract the non-AP STA’s public key from the Public Key field using the conversion defined in 12.4.7.2.5 (Octet string to element conversion). The extracted public key, ***M***, shall then be validated using the element validation technique from 12.4.5.4 (Processing of a peer’s SAE Commit message). If validation of the public key fails, the AP shall reject association with the reason of REQUEST\_DECLINED. If validation of the public key is successful, (Re)Association continues.

The AP shall then generate a private key, *n*, by randomly choosing it such that 1 < *n* < *r*, where *r* is the (prime) order of the group indicated in the Group field of the received Diffie-Hellman Parameter element. It derives a public key, ***N***, from the private key and the generator from the chosen group, ***G***:

***N*** = scalar-op(*n*, ***G***)

Where scalar-op() is defined in 12.4.4 (Finite cyclic groups). It shall then complete the Diffie-Hellman key exchange and generate a shared secret a PMK, and a PMKID:

***S*** = scalar-op(*n*, ***M***)

*s* = F(***S***)

*prk* = HKDF-Extract(F(***M***) || F(***N***) || *group*, *s*)

*PMK* = HKDF-Expand (*prk*, “OWE Key Generation”, *n*)

*PMKID* = Truncate-128(Hash(F(***M***) || F(***N***)))

Where HKDF-Extract() and HKDF-Expand() are the functions defined in IETF RFC 5869, instantiated with the hash algorithm from Table 12-1 (Hash algorithm based on length of prime), *n* is the bit length of the hash algorithm’s digest, F() is the element-to-scalar mapping function from 12.4.4 (Finite cyclic groups), and Hash() is a hash algorithm from Table 12-1 (Hash algorithm based on length of prime). Intermediate data, ***S*** and *s*, shall be irretrievably deleted upon generation of the PMK. Upon generation of the PMKID the public keys ***M*** and ***N*** can be deleted. A PMKSA shall be created containing the PMK and PMKID.

The AP shall then construct an (Re)Association Response frame per 9.3.3.6 (Association Response frame format) or 9.3.3.8 (Reassociation Response frame format) indicating OWE. The AP constructs a Diffie-Hellman Parameter element with the Group field being identical to the Group field of the received (Re)Association Request frame and assigning its public key to the Public Key field using the conversion defined in 12.4.7.2.4 (Element to octet string conversion). The AP inserts the Diffie-Hellman Parameter element into its (Re)Association Response frame per 9.3.3.6 (Association Response frame format) or 9.3.3.8 (Reassociation Response frame format) and transmits the (Re)Association Response frame to the STA.

Upon receipt of the AP’s (Re)Association Response frame indicating OWE, the STA checks that the Group field of the Diffie-Hellman Parameter element is identical to the value it inserted in its (Re)Association Request frame. If it is not the STA shall fail (Re)Association. Otherwise, it shall extract the AP’s public key from the Public Key field using the conversion defined in 12.4.7.2.5 (Octet string to element conversion). The extracted public key, ***N***, shall then be validated using the element validation technique from 12.4.5.4 (Processing of a peer’s SAE Commit message). If validation of the public key fails, the STA shall fail (Re)Association.

Otherwise, the STA shall then complete the Diffie-Hellman key exchange and generate a shared secret a PMK, and a PMKID:

***S*** = scalar-op(*m*, ***N***)

*s* = F(***S***)

*prk* = HKDF-Extract(F(***M***) || F(***N***) || *group*, *s*)

*PMK* = HKDF-Expand (*prk*, “OWE Key Generation”, *n*)

*PMKID* = Truncate-128(Hash(F(***M***) || F(***N***)))

Where HKDF-Extract() and HKDF-Expand() are the functions defined in IETF RFC 5869, instantiated with the hash algorithm from Table 12-1 (Hash algorithm based on length of prime), *n* is the bit length of the hash algorithm’s digest, F() is the element-to-scalar mapping function from 12.4.4 (Finite cyclic groups), and Hash() is a hash algorithm from Table 12-1 (Hash algorithm based on length of prime). Intermediate data, ***S*** and *s*, shall be irretrievably deleted upon generation of the PMK. Upon generation of the PMKID the public keys ***M*** and ***N*** can be deleted. A PMKSA shall be created using the PMK and PMKID.

# CID 6000

Clause Number:

Comment:

Proposed Change:

Proposed Resolution:

### Discussion

Abcd

### Proposed changes for CID 6000

**12.7.6.2 4-way handshake message 1**

*Modify 12.7.6.2 as shown (REVme-D4.1 P3036 L9):*

Abcd