802.11bi Draft Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proposed spec texts for PMKID requirement | | | | |
| Date: 2023-10-16 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | Email |
| Po-Kai Huang | Intel |  |  | po-kai.huang@intel.com |
| Ido Ouzieli |  |  |  |  |
| Johannes Berg |  |  |  |  |
| Ilan Peer |  |  |  |  |
| Robert Stacey |  |  |  |  |

Abstract

This submission proposes spec text based on the following passed requirement.

* ***11bi shall define a mechanism to prevent an eavesdropper distinguishing whether reassociation exchanges between CPE Clients and CPE APs use identical PMK or distinct PMK***

Revision History:

* Rev 0: Initial version of the document
* Rev 1: Revision based on the comments received for PMKID privacy during Nov 2023 F2F. Changes related to PMKID privacy are marked with green. Also, tag all proposals related to FT as **<tag FT>. <tag FT>** is not part of the texts to be proposed for the 11bi draft. It is mainly to differentiate the proposal for FT.

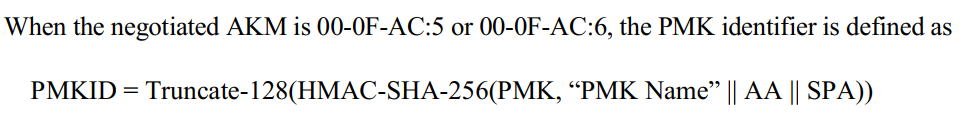
***Editing instructions formatted like this are intended to be copied into the TGbi Draft (i.e. they are instructions to the 802.11 editor on how to merge the text with the baseline documents).***

***TGbi Editor: Editing instructions preceded by “TGbi Editor” are instructions to the TGbi editor to modify or insert material in the TGbi draft. As a result of adopting the changes, the TGbi editor will execute the instructions rather than copy them to the TGbi Draft.***

**Discussion:**

It is mentioned during the teleconference call that usage of PMK caching privacy likely goes together with randomized MAC address while roaming. These two mechanisms likely will go hand in hand. Otherwise, tracking can be done through MAC address if MAC address is not randomized or PMKID if PMKID is not changed.

Once MAC address is randomized, then the formula for the PMKID computation can be reused to compute new PMKID. For example, for AKM 5 and 6, if SPA is randomized, then when PMKID is recomputed, a new PMKID identifier can be used next time. As a result, PMKID formula does not need to be changed at all.



Almost all of the PMKID formulas in 12.7.1.3 (Pairwise key hierarchy) and 12.7.1.6.3 (PMK-R0) do not need to be changed since SPA is in the input. However, for SAE and FILS, this is not the case based on the following formulas. Note that input variables for the following formulas are not available at all under PMK caching, where PMKID is directly used to identify PMK. Propose to simply use the following formula.

Truncate-128(HMAC-HASH(PMK, “PMK Name”||AA||SPA)).

Reference for current spec texts in revme D4.0.

*12.4.5.4 (Processing of a peer’s SAE Commit message).*

*PMKID = (#3506)ExtractBits(context, 0, 128)*

*12.11.2.5.2 PMKSA key derivation with FILS authentication*

*FILS shared key*

*PMKID = Truncate-128(Hash(EAP-Initiate/Reauth))*

|  |
| --- |
| *EAP-Initiate/Reauth is the EAP-RP packet sent by the STA during key establishment with FILS Shared Key authentication* |
|  |

*FILS Public Key authentication*

*PMKID = Truncate-128(Hash(gSTA || gAP))*

|  |
| --- |
| *gSTA is the STA’s Diffie-Hellman value*  *gAP is the AP’s Diffie-Hellman value* |
|  |

For FT, the identifier becomes PMKR0Name while roaming. What we need is only a new random salt. Propose to recompute PMK-R0Name-salt and have another truncation. Note that S0KH-ID is SPA. Likely R1KH will contact R0KH with the new S0KH-ID and S1KH-ID to retrieve PMK-R1 and recompute PMKR0Name. Note that PMKR1Name does not need to be recomputed since PMKR1Name is in reassociation request, which will be encrypted once OTA MAC address is randomized.

*PMK-R0Name-Salt = KDF-Hash-128(XXKey, “FT-R0”, SSIDlength || SSID || MDID || R0KHlength || R0KH-ID || S0KH-ID)*

*PMKR0Name = Truncate-128(Hash(“FT-R0N”|| PMK-R0Name-Salt))*

Reference for current spec texts in revme D4.0.

*12.7.1.6.3 PMK-R0*

*R0-Key-Data = KDF-Hash-Length(XXKey, “FT-R0”, SSIDlength || SSID || MDID || R0KHlength || R0KH-ID || S0KH-ID)*

*PMK-R0 = ExtractBits(R0-Key-Data, 0, Q)*

*PMK-R0Name-Salt = ExtractBits(R0-Key-Data, Q, 128)*

*PMKR0Name = Truncate-128(Hash(“FT-R0N” || PMK-R0Name-Salt))*

**Proposed Texts:**

**TGbi Editor: *Instruction: Insert new rows in Table 9-363 in 9.4.2.241 RSNXE as shown below***

9.4.2.241 RSNXE

|  |  |  |
| --- | --- | --- |
| * **Extended RSN Capabilities field** | | |
| **Bit** | **Information** | **Notes** |
| <ANA> | PMKSA Caching Privacy Support | A EDP STA sets the PMKSA Caching Privacy Support subfield to 1 if dot11EDPPMKSACachingRivacySupportActivated is true. Otherwise, this subfield is set to 0. See 12.13.x (PMKSA Caching Privacy Support). |

**TGbi Editor: *Instruction: Insert 12.13.x PMKSA caching privacy as shown below***

**12.13 Client Privacy Enhancement**

**12.13.x PMKSA caching privacy**

This subclause defines rules to have PMKSA caching privacy such that identifier related to PMKSA caching cannot be used for tracking.

A STA that sets the PMKSA Caching Privacy Support subfield in the RSNXE to 1 shall set the Encryption of the Frame Body Field of the (Re)Association Request/Response Frame Support subfield in the RSNXE to 1.

**12.13.x.1 PMKID privacy**

After the indicated PMKID identifies a cached PMKSA, and a PTKSA is established using the identified PMKSA,

* For non-MLO, if the EDP non-AP STA and the EDP AP set the PMKSA Caching Privacy Support subfield in the RSNXE to 1, both the EDP non-AP STA and the EDP AP shall recompute the PMKID for the identified PMKSA.
* For MLO, if any EDP non-AP STA affiliated with an EDP non-AP MLD and any EDP AP affiliated with an EDP AP MLD set the PMKSA Caching Privacy Support subfield in the RSNXE to 1, both the EDP non-AP MLD and the EDP AP MLD shall recompute the PMKID for the identified PMKSA.

NOTE - For MLO, all STAs affiliated with an MLD set the RSNXE to the same value.

The PMKID shall be recomputed as defined in 12.7.1.3 (Pairwise key hierarchy) and 12.7.1.6.3 (PMK-R0) except that

* when the negotiated AKM is 00-0F-AC:8 or 00-0F-AC:24 or 00-0F-AC:25, the PMKID is recomputed as

PMKID = Truncate-128(HMAC-Hash(PMK, “PMK Name”||AA||SPA))

where Hash is the hash algorithm specified in 12.4.2 (Assumptions on SAE).

* when FILS authentication is used, the PMKID is recomputed as

PMKID = Truncate-128(HMAC-Hash(PMK, “PMK Name”||AA||SPA))

where Hash is the hash algorithm specific to the negotiated AKM (see Table 9-188 (AKM suite selectors))

|  |
| --- |
|  |

NOTE – In order to recompute a different PMKID to ensure privacy, SPA address needs to be randomized in the frame indicating PMKID to identify cached PMKSA. As a result, the recomputed PMKID is different after each PTKSA is established, and the recomputed PMKID is used on the next connection to identify the cached PMKSA.

**12.13.x.2 PMKR0Name privacy <tag FT>**

APs in the same mobility domain shall set the PMKSA Caching Privacy Support subfield in the RSNXE to the same value.

If both a FTO and target FTR set the PMKSA Caching Privacy Support subfield in the RSNXE to 1, after the indicated PMKR0Name used by the target FTR to identify PMK-R1 (see 13.8.1 (Overview)), and a PTKSA is established using the identified PMK-R1,

* the R1KH of the target FTR shall contact R0KH to provide the latest S0KH-ID, and
* both the S0KH of the FTO and the R0KH contacted by the target FTR shall recompute the PMKR0Name.

NOTE - For MLO, all STAs affiliated with an MLD set the RSNXE to the same value.

The PMKR0Name shall be recomputed as follows:

PMK-R0Name-Salt = KDF-Hash-128 (XXKey, “FT-R0”, SSIDlength || SSID || MDID || R0KHlength || R0KH-ID || S0KH-ID)

where KDF-Hash-128, XXXKey, SSIDlength, SSID, MDID, R0KHlength, R0KH-ID, S0KH-ID are

defined in 12.7.1.6.3 (PMK-R0)

PMKR0Name = Truncate-128(Hash(“FT-R0N” || PMK-R0Name-Salt))

NOTE – In order to recompute a different PMKR0Name to ensure privacy, SPA address needs to be randomized in the frame indicating PMKR0Name to identify cached PMK-R0 security association. As a result, S0KH-ID and S1KH-ID are different, and the recomputed PMKR0Name is different after each PTKSA is established, and the recomputed PMKR0Name is used on the next connection to identify the cached PMK-R0 security association.

NOTE – PMKR1Name is still derived based on the indicated PMKR0Name with the same formula defined in 12.7.1.6.4 (PMK-R1) for the first time and PMKR1Name once derived is not recomputed due to encryption of Reassociation Request and Response frame.

The R0KH may then deliver the latest PMKR0Name to other R1KHs with corresponding PMK-R1 SA in the same mobility domain. The R1KH of the target FTR may also retrieve the latest PMKR0Name from the R0KH.

**TGbi Editor: *Instruction: Modify 9.4.2.23.5 as shown below***

* **PMKID**

The PMKID Count field indicates the number of PMKIDs that are contained in the PMKID List field. The PMKID List field contains a series (possibly empty) of PMKIDs.

When one or more PMKIDs are included in a (Re)Association Request frame or FILS Authentication frame to an AP, they identify PMKSAs that the STA believes to be valid for the destination AP. When a PMKID is included in a FILS Authentication frame to a STA, it identifies a PMKID that the AP has selected.

A PMKID in the PMKID List field can refer to

* The PMKID of a cached PMKSA that has been obtained through preauthentication with the target AP
* The latest derived PMKID of a cached PMKSA from an EAP, FILS, or SAE authentication
* The latest derived PMKID of a PMKSA derived from a PSK for the target AP
* The latest derived PMKR0Name of a PMK-R0 security association derived as part of an FT initial mobility domain association or recomputed as part of a fast BSS transition**<tag FT>**
* The PMKR1Name of a PMK-R1 security association derived as part of an FT initial mobility domain association or as part of a fast BSS transition.

See 12.7.1.3 (Pairwise key hierarchy), 12.7.1.6.3 (PMK-R0), and 12.13.x.1 (PMKID privacy) for the construction of the PMKID, 13.8 (FT authentication sequence) for the population of PMKID List for fast BSS transitions, 12.6.8.3 (Cached PMKSAs and RSNA key management) for the population of PMKID List when using PMKSA caching, 13.4 (FT initial mobility domain association) for the population of PMKID List for FT initial mobility domain association, 12.11.2 (FILS authentication protocol) for the population of PMKID List with FILS authentication, and 12.7.1.6 (FT key hierarchy) and 12.13.x.1 (PMKR0Name privacy) **<tag FT>**

for the construction of PMKR0Name and PMKR1Name.

NOTE—A STA need not insert a PMKID in the PMKID List field if the STA (M118)is not using that PMKSA.

**TGbi Editor: *Instruction: Modify 12.2.10 as shown below***

* **Requirements for support of MAC privacy enhancements**

MAC privacy enhancements are enabled on a non-AP STA when dot11MACPrivacyActivated is(M118) true. The STA shall periodically change its MAC address to a random value while not associated to a BSS. The STA shall construct the randomized MAC address from the locally administered address space as defined in IEEE Std 802-2014 and IEEE Std 802c™-2017. However, the non-AP STA shall not change its MAC address during a transactional exchange, for example, transmitting Public Action frames for preassociation discovery, or during the creation of state on an AP using preassociation capabilities, for example, RSN preauthentication or FT over-the-DS. The smaller the period of MAC address change, down to a single transmitted frame per MAC address, the greater the privacy these enhancements afford. The actual period used when changing a MAC address is implementation dependent and outside the scope of this standard.

If such a non-AP STA starts any transaction that establishes state bound to a MAC address and might elect to establish an association or establish transaction state with a discovered BSS, it shall check the value of dot11LocallyAdministeredMACConfig and shall configure its MAC address according to the rules of the local address space prior to the start of the transaction. State created with an AP using a prior MAC address, for instance, RSN preauthentication state or FT state established over-the-DS, is bound to the MAC address used when that state was created. Prior to establishing an association to the AP, the non-AP STA shall change its MAC address to the MAC address used when the state was created unless client privacy enhancement is used (see 12.13 (Client Privacy Enhancement)).

The SME of the non-AP STA may change the MAC address by generating an MLME-UPDATEMACADDRESS.request primitive containing the new MAC address. On receipt of an MLME‑UPDATEMACADDRESS.request primitive, the MLME shall attempt to update the MAC address that is to be used by the MAC entity and shall generate an MLME-UPDATEMACADDRESS.confirm primitive to notify the SME whether the MAC address has been changed to the new value.

Every time a MAC address is changed to a new random value, counters in (#270)all sequence number spaces used to identify each MSDU, A-MSDU or MMPDU shall be reset (see 10.3.2.14.2 (Transmitter requirements)) and the STA shall set the TXVECTOR parameter SCRAMBLER\_RESET to RESET\_SCRAMBLER on the next transmitted PPDU.

The non-AP STA connecting to an infrastructure BSS shall retain a single MAC address for the duration of its connection across an ESS unless client privacy enhancement is used (see 12.13 (Client Privacy Enhancement)). A PMKSA created as part of an RSNA (M118)contains the MAC address used to create the PMKSA (see 12.6.1.1.2 (PMKSA)). The non-AP STA that supports PMKSA caching shall, if necessary, change its MAC address back to that value when attempting a subsequent association to the ESS using PMKSA caching unless client privacy enhancement is used (see 12.13 (Client Privacy Enhancement)).

To construct a random MAC address, the STA shall select a randomized MAC address according to IEEE Std 802-2014 and IEEE Std 802c-2017.

To avoid leakage of possibly sensitive network identifying information, STAs should refrain from transmitting Probe Request frames containing preferred SSID values and, instead, use passive scanning or transmit Probe Request frames containing the wildcard SSID.

When dot11MACAddressPolicyActiviated is true, an AP shall set the MAC Address Policy field in the Extended Capabilities field to 1, indicating the existence of a MAC address policy. When dot11MACAddressPolicyActivated is false, an AP STA shall set the MAC Address Policy field in the Extended Capabilities field to 0, indicating that local MAC addresses are not restricted.

A non-AP STA that receives from an AP an Extended Capabilities field with the Local MAC Address Policy subfield set to 1 should, unless it has previously stored the MAC address policy for the ESS, discover that policy, using the MAC Address Policy ANQP-element, before sending any Association Request frame to that AP using a local MAC address as the TA.

MAC privacy enhancements are enabled on a non-AP MLD when dot11MACPrivacyActivated is set to true. When enabled, the non-AP MLD shall adhere to the above requirements for a non-AP STA (that is not affiliated with an AP MLD) in selecting a MLD MAC address, including sequence number space and scrambler requirements. The above requirements defined for a non-AP STA in managing its MAC address during association or establishing transaction state with an AP shall apply to the non-AP MLD in managing its MLD MAC address during association or establishing transaction state with an AP MLD.

When a non-AP MLD with MAC privacy enhancements enabled becomes a non-AP STA for the purpose of BSS transition to an AP, the non-AP STA shall adhere to the requirements above and 35.3.1 (General) for managing and selecting the MLD MAC address. Similarly, when a non-AP STA becomes a non-AP MLD for the purpose of BSS transition to an AP MLD, the non-AP MLD shall adhere to the requirements above and 35.3.1 (General) for managing the MLD MAC address.

NOTE—The non-AP STA MAC address is the MLD MAC address when a non-AP MLD transitions to an AP. See

35.3.1 (General).

When MAC privacy enhancements are enabled on a non-AP MLD, the SME of the non-AP MLD manages the MAC addresses for each of the affiliated non-AP STAs. The randomized MAC address for an affiliated non-AP STA shall be selected according to IEEE Std 802-2014 and IEEE Std 802c-2017. Every time an affiliated STA MAC address is changed to a new random value, counters in all sequence number spaces used to identify each MMPDU shall be reset (see 10.3.2.14.2 (Transmitter requirements)) and the STA shall set

the TXVECTOR parameter SCRAMBLER\_RESET to RESET\_SCRAMBLER on the next transmitted PPDU.

A non-AP MLD connecting to an AP MLD shall not change the affiliated non-AP STA MAC address(es) for the duration of its association to the AP MLD. An affiliated non-AP STA MAC address may be changed subject to the requirements above when performing BSS transition or ESS transition.

**TGbi Editor: *Instruction: Modify 12.6.1.1.2 PMKSA as shown below***

* **PMKSA**

The PMKSA is created by the Authenticator’s SME and Supplicant’s SME when EAP authentication, SAE authentication, (#1084)FILS authentication, or an OWE exchange completes successfully, or when the PSK is configured.

When the negotiated AKM uses PMKID derivation with (#3744)PTK-KCK as a parameter as defined in 12.7.1.3 (Pairwise key hierarchy), the PMKID derived from the PTK-KCK during the initial 4-way handshake is not changed during the lifetime of this PMKSA except when PMKSA caching privacy is used, see 12.13.x.1 (PMKID privacy).

A PMKSA association is bidirectional. In other words, both parties use the information in the security association for both sending and receiving. The PMKSA is used to create the PTKSA. PMKSAs have a certain lifetime. The PMKSA consists of the following:

* Latest derived PMKID, as defined in 12.7.1.3 (Pairwise key hierarchy) or 12.7.1.6.3 (PMK-R0) or 12.13.x.1 (PMKID privacy). The PMKID identifies the security association.
* Authenticator’s or peer’s MAC address. For multi-band RSNA, the MAC address is associated with the operating band in use when the PMKSA is established.
* PMK; or if the PMKSA was established with an (#3266)AKMP for which the Authentication type column includes FT authentication (see Table 9-188 (AKM suite selectors)), MPMK (see 12.7.1.6.3 (PMK-R0)).
* Lifetime, as defined in 12.7.1.3 (Pairwise key hierarchy) or 12.7.1.6 (FT key hierarchy).
* AKMP.
* All authorization parameters specified by the AS or local configuration. This might include parameters such as the STA’s authorized SSID.
* Cache Identifier, if advertised by the AP in FILS Indication element.

**TGbi Editor: *Instruction: Modify 12.6.1.1.3 as shown below***

* **PMK-R0 security association**

The PMK-R0 security association is the result of a successful completion of the IEEE 802.1X authentication, SAE authentication, or use of PSK during the FT initial mobility domain association. This security association is bidirectional. It has a certain lifetime. It consists of the following:

* SSID
* (#1776)MDID
* PMK-R0
* R0KH-ID
* Latest derived PMKR0Name**<tag FT>**
* Latest S0KH-ID**<tag FT>**
* PMK-R0 lifetime
* Pairwise cipher suite selector
* All authorization parameters specified by the AS or local configuration

**TGbi Editor: *Instruction: Modify 12.6.1.1.4 as shown below***

* **PMK-R1 security association**

The PMK-R1 security association is the result of

* A successful completion of the IEEE 802.1X authentication, SAE authentication, or use of PSK during the FT initial mobility domain association or
* A successful completion of the authentication phase in the fast BSS transition to the target AP

This security association is bidirectional. It has a certain lifetime. It consists of the following:

* SSID
* MDID
* PMK-R1
* PMK-R1 lifetime
* PMKR1Name
* R1KH-ID
* R0KH-ID
* Latest derived PMKR0Name**<tag FT>**
* Latest S0KH-ID**<tag FT>**
* Latest S1KH-ID**<tag FT>**
* Pairwise cipher suite selector
* All authorization parameters specified by the AS or local configuration

**TGbi Editor: *Instruction: Modify 13.5.2 as shown below***

* **Over-the-air FT protocol authentication in an RSN<tag FT>**

(…existing texts…)

If PMKSA caching privacy is not used, the R1KH of the target FTR uses the value of PMKR0Name and other information in the frame to calculate PMKR1Name and check if a PMK-R1 can be identified with the PMKR1Name. If PMKSA caching privacy is used, then the R1KH of the target FTR uses the value of PMKR0Name to check if a PMK-R1 and corresponding PMKR1Name can be identified (see 12.6.1.1.4 (PMK-R1 security association)).

If the target FTR does not identify a PMK-R1, it may retrieve that key from the R0KH identified by the FTO. See 13.2 (Key holders). Upon receiving a new PMK-R1 for a STA, the target AP shall delete the prior PMK-R1 security association and PTKSAs derived from the prior PMK-R1.

(…existing texts…)

**TGbi Editor: *Instruction: Modify 13.8.1 as shown below***

* **Overview<tag FT>**

(…existing texts…)

The first message is used by the FTO to initiate a fast BSS transition. When RSNA is enabled, the FTO shall include the R0KH-ID and the SNonce in the FTE and the PMKR0Name in the RSNE. If PMKSA caching privacy is not used, the target FTR can use the PMKR0Name to derive the PMKR1Name and check if a PMK-R1 can be identified. If PMKSA caching privacy is used, then the R1KH of the target FTR uses the value of PMKR0Name to check if a PMK-R1 and corresponding PMKR1Name can be identified (see 12.6.1.1.4 (PMK-R1 security association)). If the target FTR does not identify a PMK-R1, it may attempt to retrieve that key from the R0KH identified by R0KH-ID. See 13.2 (Key holders). The FTO includes a fresh SNonce as its contribution to the association instance identifier and to provide key separation of the derived PTK; it is selected randomly to serve as a challenge that demonstrates the liveness of the peer in the fourth message.

(…existing texts…)