IEEE P802.11
Wireless LANs

|  |
| --- |
| 802.11bi – Support for rotating MAC addresses while associated |
| Date: September 13, 2022 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Antonio de la Oliva | Interdigital Ltd, UC3M |  |  | aoliva@it.uc3m.es |
| Joseph Levy | Interdigital Ltd, |  |  |  |

 Abstract

This submission tackles the required modifications to the standard to support modification of the A1, A2 MAC Addresses of frames while the STA is associated.

This contribution only tackles at the moment PV0 frames and non-MLD. Baseline for this document is REVme\_D2.0.

Proposed contribution

**3 Definitions**

otaMAC: over the air medium access control address

*Note: A temporal MAC address used in frames transmitted over the air. For individually addressed frames where the To DS bit is set to 1 and the From DS bit is set to 0, the otaMAC is transmitted as the Address 2. For individually addressed frames where the To DS bit is set to 0 and the From DS bit is set to 1, the otaMAC is transmitted as the Address 1. The purpose of the otaMAC is to keep private the aaMAC of the STA and may potentially be changed on a packet basis.*

aaMAC: association and authentication medium access control address

*Note: The aaMAC corresponds to the MAC address used for the association and authentication process between the AP and the STA and is the one indexed in the RSNA. This MAC address is used to set up the RSNA, for routing traffic to the STA on the DS network segment and it is also the one used for mobility related operations such as Fast Transition mechanisms.*

**5.1.5 MAC data service architecture**

**5.1.5.1 General**

When transparent FST is not being used, the MAC data plane architecture (i.e., processes that involve transport of all or part of an MSDU) is shown in Figure 5-1 (MAC data plane architecture(#1001)(11ay)). When transparent FST is being used, the MAC data plane architecture is shown in Figure 5-2 (MAC data plane architecture (transparent FST)(11ay)(#1938)).(#110)

The dotted line box labeled “Role-specific behaviors” is replaced by one of several options, depending on the role of the STA. See the following subclauses

During transmission, an MSDU goes through the processes shown in the left-hand side of Figure 5-1 (MAC data plane architecture(#1001)(11ay)). When transparent FST is used, an MSDU first goes, as shown in Figure 5-2 (MAC data plane architecture (transparent FST)(11ay)(#1938)), through an additional transparent FST entity that contains a demultiplexing process that forwards the MSDU down to the selected TX MSDU Rate Limiting process and from there to MAC data plane processing as described in the previous sentence. IEEE Std 802.1X-2010 may block the MSDU at the Controlled Port before the preceding processing occurs. Otherwise, at some point, the Data frames that contain all or part of the MSDU are queued per AC/TS. In the case of an EP (Enhanced Privacy) MPDU, the block “EP anonymisation” modifies the Address 1 (e.g., from AP to non-AP STA transmission) and/or the Address 2 (e.g., from non-AP STA to AP transmission) of the MAC header, replacing the aaMAC by the otaMAC currently in used by the intended receiver of the EP MPDU (Address 1) and/or the current transmitter of the EP MPDU (Address 2).

During reception, a received Data frame goes through the processes shown in the right-hand side of of Figure 5-1 (MAC data plane architecture(#1001)(11ay)). Then, one or more MSDUs are delivered to the MAC SAP or, via the DSAF, to either the DS or an IEEE 802.1Q bridge port. When transparent FST is used, MSDUs originating from different PHY SAPs go, as shown in Figure 5-2 (MAC data plane architecture (transparent FST)(11ay)(#1938)), through a final step of a transparent FST entity that contains a multiplexing process before delivering the MSDU. The IEEE 802.1X -Controlled/Uncontrolled Ports discard any received MSDU if the Controlled Port is not enabled and if the MSDU does not represent an IEEE 802.1X frame. In the case of an EP MPDU, the block “EP de-anonymisation” modifies the Address 1 (e.g., from AP to non-AP STA transmission) and/or Address 2 (e.g., from non-AP STA to AP transmission) of the MAC header by replacing the otaMAC with the aaMAC.

NOTE—Many of the processes shown in Figure 5-1 (MAC data plane architecture(#1001)(11ay)) also apply to MMPDU flows for the MAC control plane architecture, and the processes shown at the bottom also apply to Control and Extension frames.

(#1938)When transparent FST is used, the same security keys and PN counters are used by the MAC data plane to encrypt the MPDU prior to and following an FST, and the same security keys and replay counters are used to check the integrity and perform the protection of MPDUs. When nontransparent FST is used, independent RSNAs, security keys, replay counters, and PN counters have to be established for each MAC data plane to be used prior to and following an FST. When transparent FST is used, a single MAC SAP at each peer is presented to the higher layers of that peer for all of the frequency bands/channels that are identified by the same MAC address at that peer. When nontransparent FST is used, different MAC SAPs are presented to higher layers since different MAC addresses are used prior to and following an FST.



Figure 5-1 – MAC data plane architecture

**12.5.2.3 CCMP cryptographic encapsulation**

**12.5.2.3.1 General**

The CCMP cryptographic encapsulation process is depicted in modified Figure 12-18 (CCMP encapsulation block diagram(#1589)) and described as follows:

1. For secure PV0 MPDUs, CCMP encrypts the Frame Body field of a plaintext MPDU and encapsulates the resulting cipher text using the following steps:
2. Increment the PN, to obtain a fresh nonzero(#1505) PN for each MPDU, so that the PN never repeats for the same temporal key.

NOTE 1—Retransmitted MPDUs are not modified on retransmission.

1. Use the fields in the MPDU header to construct the additional authentication data (AAD) for CCM. The CCM algorithm provides integrity protection for the fields included in the AAD. (#1951)MPDU header fields that might change when retransmitted are muted by being (#1951)masked out when calculating the AAD.
2. Construct the CCM nonce block as defined in 12.5.2.3.4 (Construct CCM nonce) from the PN, A2, and the priority value of the MPDU where A2 is MPDU Address 2. If the Type field of the Frame Control field is 10 (Data frame) and there is a QoS Control field present in the MPDU header, the priority value of the MPDU is equal to the value of the TID subfield of the QoS Control field (bits 0 to 3 of the QoS Control field). If the Type field of the Frame Control field is 00 (Management frame) and the frame is a QMF, the priority value of the MPDU is equal to the value in the ACI subfield of the Sequence Number field. Otherwise, the priority value of the MPDU is equal to the fixed value 0.
3. Construct the CCMP header as defined in 12.5.2.3.5 (Construct CCMP header for PV0 MPDUs).
4. In case the MPDU is addressed towards an EP STA with an aaMAC and otaMAC binding in place, use the temporal key associated to the aaMAC, AAD, nonce, and MPDU data to form the cipher text and the encrypted MIC. Otherwise, use the temporal key, AAD, nonce, and MPDU data to form the cipher text and the encrypted MIC. This step is known as CCM originator processing.
5. Form the encrypted MPDU by combining the original MPDU header, the CCMP header, the encrypted data and the encrypted MIC, as described in 12.5.2.2 (CCMP MPDU format).

The CCM reference describes the processing of the key, nonce, AAD, and data to produce the encrypted output. See 12.5.2.3.2 (PN processing) to 12.5.2.3.7 (CCM originator processing) for details of the creation of the AAD and nonce from the MPDU and the associated MPDU-specific processing.



**Figure 12- 18 – CCMP encapsulation block diagram**

**12.5.2.4 CCMP decapsulation**

**12.5.2.4.1 General**

Figure 12-23 (CCMP decapsulation block diagram) depicts the CCMP decapsulation process and is described as follows.

1. For secure PV0 MPDUs, CCMP decrypts the Frame Body field of a cipher text MPDU and decapsulates a plaintext MPDU using the following steps:
2. The encrypted MPDU is parsed to construct the AAD (see 12.5.2.3.3 (Construct AAD)) and nonce (see 12.5.2.3.4 (Construct CCM nonce)) values.
3. The MIC is extracted for use in CCM integrity checking.
4. In case the MPDU is received at an EP STA with an aaMAC and otaMAC binding in place, CCM recipient processing uses the temporal key associated to the aaMAC, AAD, nonce, encrypted MIC, and MPDU cipher text data to recover the MPDU plaintext data as well as to check the integrity of the AAD and MPDU plaintext data. Otherwise, CCM recipient processing uses the temporal key, AAD, nonce, encrypted MIC, and MPDU cipher text data to recover the MPDU plaintext data as well as to check the integrity of the AAD and MPDU plaintext data.
5. The received MPDU header and the MPDU plaintext data from CCM recipient processing are concatenated to form a plaintext MPDU.
6. The decryption processing prevents replay of MPDUs by validating that the PN in the MPDU is greater than the replay counter maintained for the session(#193), and TID (for Data frames) or ACI (for QMFs).

See 12.5.2.4.2 (CCM recipient processing) to 12.5.2.4.4 (PN and replay detection) for details of this processing.

When the received frame is a CCMP protected individually addressed robust Management frame or PV1 Management frame, contents of the MMPDU body after protection is removed shall be delivered to the SME via the MLME primitive designated for that MMPDU or PV1 Management frame rather than through the MA‑UNITDATA.indication primitive.



**Figure 12- 23 – CCMP decapsulation block diagram**