IEEE P802.11
Wireless LANs

|  |
| --- |
| Proposed Draft Text for MLO Multi-Link Security: Operation |
| Date: 2021-01-01 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Gaurav Patwardhan | HPE |  |  | gaurav.patwardhan@hpe.com |
| Dan Harkins | HPE |  |  |  |
| Eldad Perahia | HPE |  |  |  |

Abstract

This document contains draft text for MLO Multi-Link Security: Operation, for inclusion into TGbe draft D0.4 based on the motions and SPs listed in the following pages.

Revisions:

* Rev 0: Initial version of the document.
* Rev 1: Incorporated some comments.

This text is being prepared for the following SPs/motions:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MAC | MLO-Multi-link security: Operation | Gaurav Patwardhan | Rojan Chitrakar, Subir Das, Mike Montemurro, Po-Kai Huang, Joseph Levy, Jay Yang, Duncan Ho, Mark Hamilton, Xiandong Dong, Yong Liu | R1 | Uploaded:Presented:Straw Polled: | Motion 144, #SP329Motion 144, #SP330 |

In R1, the followings are supportedfor the cases <To DS =0, From DS = 1> and <To DS = 1, From DS = 0> for individually addressed Data frames:

1. replacing Addresses A1 and A2 with MLD MAC Addresses for AAD computation,
2. replacing Address A3 (only in case when A3 is set to BSSID) with MLD MAC Address for AAD computation,
3. using MLD MAC address in A2 for constructing Nonce.

[Motion 144, #SP329, [35] and [205]]

In R1, the followings are supported for the case <To DS =1, From DS = 1> for individually addressed Data frames:

1. replacing Addresses A1 and A2 with MLD MAC Addresses for AAD computation,
2. replacing Addresses A3 and A4 (only in case when A3 and A4 both are set to BSSID) with MLD MAC Addresses for AAD computation,
3. Using MLD MAC address in A2 for constructing Nonce.

[Motion 144, #SP330, [35] and [205]]

**Discussion:**

Logic for AAD construction changes:

Sublcause 12.5.3.3.3 bullet 3): Addresses the red box highlighted in Table 9-30 figure shown below.

Subclause 12.5.3.3.3 bullet 4): Initial IF condition addresses the pink boxed section

* Subclause 12.5.3.3.3 bullet 4) sub-bullet 1: Addresses the blue boxed section
* Subclause 12.5.3.3.3 bullet 4) sub-bullet 2: Addresses the green boxed section
* Subclause 12.5.3.3.3 bullet 4) sub-bullet 3: Addresses the orange boxed section



***TGbe editor: Modify subclause 12.5.3.3.1 (General) as follows:***

* CCMP cryptographic encapsulation(#2720)
* General(#2720)

The CCMP cryptographic encapsulation process is depicted in Figure 12-18 (CCMP encapsulation block diagram(#4087)).



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

* (11ah)For secure PV0 MPDUs, CCMP encrypts the Frame Body field of a plaintext MPDU and encapsulates the resulting cipher text using the following steps:
* Increment the PN, to obtain a fresh PN for each MPDU, so that the PN never repeats for the same temporal key.

(#4613)NOTE—Retransmitted MPDUs are not modified on retransmission.

* 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. MPDU header fields that may change when retransmitted are muted by being set to a known valuewhen calculating the AAD.
* In case of a secure PV0 MPDU that is an individually addressed Data frame to be encrypted by an MLD, construct the CCM nonce block as defined in 12.5.3.3.4 (Construct CCM nonce) from the PN, transmitting MLD MAC Address, and the priority value of the MPDU. Otherwise, construct the CCM nonce block (#4613)as defined in 12.5.3.3.4 (Construct CCM nonce(#2720)) 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 (#4385)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)(#4613) 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.
* (#4613)Construct the CCMP header as defined in 12.5.3.3.5 (Construct CCMP header for PV0 MPDUs(#2720)).
* Use the temporal key, AAD, nonce, and MPDU data to form the cipher text and (#4088)the encrypted MIC. This step is known as CCM originator processing.
* Form the encrypted MPDU by combining the original MPDU header, the CCMP header, the encrypted data and (#4088)the encrypted MIC, as described in 12.5.3.2 (CCMP MPDU format).

***TGbe editor: Modify subclause 12.5.3.3.3 (Construct AAD) as follows:***

* Construct AAD(#2720)
* (11ah)For PV0 MPDUs, the format of the AAD is shown in Figure 12-19 (AAD construction for PV0 MPDUs(11ah)). The length of the AAD for PV0 varies depending on the presence or absence of the QC and A4 fields and is shown in Table 12-3 (AAD length for PV0 MPDUs(11ah)).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | FC | A1 | A2 | A3 | SC | A4 | QC |
| Octets: | 2 | 6 | 6 | 6 | 2 | 6 | 2 |
| * AAD construction for PV0 MPDUs(11ah)
 |

|  |
| --- |
| * AAD length for PV0 MPDUs(11ah)
 |
| QC field | A4 field | AAD length(octets) |
| Absent | Absent | 22 |
| Present | Absent | 24 |
| Absent | Present | 28 |
| Present | Present | 30 |

The AAD is constructed from the MPDU header. (#2352)The AAD includes neither the Duration/ID field nor the HT Control field because the contents of these fields might change during normal operation (e.g., due to a rate change preceding retransmission). The HT Control field might also be inserted or removed during normal operation (e.g., retransmission of an (#4130)A-MPDU where the original (#4130)A-MPDU included an MRQ that has already generated a response). For similar reasons, several subfields in the Frame Control field are masked to 0. (11ah)For PV0 MPDUs, (#4261)AAD construction is performed as follows:

* (11ah)FC – MPDU Frame Control field, with
* Subtype subfield (bits 4 5 6) in a Data frame masked to 0
* Retry subfield (bit 11) masked to 0
* Power Management subfield (bit 12) masked to 0
* More Data subfield (bit 13) masked to 0
* Protected Frame subfield (bit 14) always set to 1
* +HTC(#66) subfield (bit 15) as follows:
* Masked to 0 in all Data frames containing a QoS Control field
* Unmasked otherwise
* Other subfields are not modified(#4417)
* (11ah) If dot11MultiLinkActivated is true, either of To DS or From DS subfields in the MAC header of the MPDU is set to 1, and the MPDU is an individually addressed Data frame then A1 is set to:
* the MLD MAC Address of the intended receiver MLD of the MPDU.

– otherwise, A1 is set to MPDU Address 1 field.

* (11ah) If dot11MultiLinkActivated is true, if either of To DS or From DS subfields in the MAC header of the MPDU is set to 1, and the MPDU is an individually addressed Data frame then A2 is set to:
* the MLD MAC Address of the transmitting MLD of the MPDU.

– otherwise, A2 is set to MPDU Address 2 field.

* (11ah)If dot11MultiLinkActivated is true, MPDU Address field 3 is BSSID, and the MPDUis an individually addressed Data frame, then:
* if To DS subfield is set to 0 and From DS subfield is set to 1 in the MAC header of the MPDU, set A3 to the MLD MAC Address of the transmitting MLD,
* else if To DS subfield is set to 1 in the MAC header of the MPDU, set A3 to the MLD MAC Address of the receiving MLD, ,
* else set A3 to MPDU Address 3 field.

–

* (11ah)SC – MPDU Sequence Control field, with the Sequence Number subfield (bits 4–15 of the Sequence Control field) masked to 0. The Fragment Number subfield is not modified.
* (11ah)A4, if present, is set as follows:
* If dot11MultiLinkActivated is true, MPDU Address 4 field is a BSSID, and the MPDU is an individually addressed Data frame, then A4 is set to the MLD MAC Address of the transmitting MLD.

– otherwise A4, if present is set to MPDU Address 4 field.

* (11ah)QC – QoS Control field contains(M158) the MSDU priority, if present. The QC TID is used in the construction of the AAD. When in a non-DMG BSS and both the STA and its peer have their SPP A-MSDU Capable fields equal to 1, bit 7 (the A-MSDU Present field) is used in the construction of the AAD. The remaining QC fields are masked to 0 for the AAD calculation (bits 4 to 6, bits 8 to 15, and bit 7 when either the STA or its peer has the SPP A-MSDU Capable field equal to 0). When in a DMG BSS, the A-MSDU Present bit 7 and A-MSDU Type bit 8 are used in the construction of the AAD, and the remaining QC fields are masked to 0 for the AAD calculation (bits 4 to 6, bits 9 to 15).

***TGbe editor: Modify subclause 12.5.3.3.4 (Construct CCM nonce) as follows:***

* Construct CCM nonce(#2720)

The Nonce field occupies 13 octets, and its structure is shown in Figure 12-21 (Nonce field(#1406)(11ah)). The structure of the Nonce Flags subfield of the Nonce field is shown in Figure 12-22 (Nonce Flags subfield(11ah)).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Nonce Flags | STA or MLD MAC Address Identified By A2 | PN |
| Octets: | 1 | 6 | 6 |
| * Nonce field(#1406)(11ah)
 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | B0         B3 | B4 | B5 | B6         B7 |
|  | Priority | Management | PV1 | Zeros |
| Bits: | 4 | 1 | 1 | 2 |
| * Nonce Flags subfield(11ah)
 |

(#4614)The Priority subfield shall be set to the priority value of the MPDU.

(#4614)The Management subfield shall be set to 1 if the MPDU is a Management frame and management frame protection is negotiated; otherwise, it shall be set to 0.

(#4614)The PV1 subfield shall be set to 1 for a PV1 frame; otherwise, it shall be set to 0.

(#4614)The Zeros subfield shall be set to 0.

(#4614) If dot11MultiLinkActivated is true, either To DS or From DS subfields in the MAC header of the MPDU are set to 1, and the MPDU is an individually addressed Data frame, then the STA or MLD MAC Address Identified by A2 subfield shall contain the MLD MAC Address of the transmitting MLD. Otherwise, the STA or MLD MAC Address Identified By A2 subfield shall contain the Address 2 field from the MAC header for PV0 MPDUs and the MAC address identified by the A2 field in the MAC header for PV1 MPDUs (see 9.8.3.2 (Address fields)).

(#4614)The PN subfield shall contain the packet number, with PN0 in the last octet of the subfield.

***TGbe editor: Modify subclause 12.5.3.4.1 (General) as follows:***

* CCMP decapsulation(#2720)
* General(#2720)

Figure 12-23 (CCMP decapsulation block diagram(#4087)) depicts the CCMP decapsulation process.



**Figure 12-23—CCMP decapsulation block diagram (#4087)**

* (11ah)For secure PV0 MPDUs, CCMP decrypts the Frame Body field of a cipher text MPDU and decapsulates a plaintext MPDU using the following steps:
* (#4614)(11ah)The encrypted MPDU is parsed to construct the AAD (see 12.5.3.3.3 (Construct AAD(#2720))) and nonce (see 12.5.3.3.4 (Construct CCM nonce(#2720))) values. In addition, if the intended receiver of the MPDU is an MLD and the MPDU is an individually addressed Data frame, then the intended receiver MLD MAC Address is passed to construct the AAD (see 12.5.3.3.3 (Construct AAD)) and nonce (see 12.5.5.3.4 (Construct CCM nonce)) values.
* (11ah)The(#4088) MIC is extracted for use in (#4386)CCM integrity checking.

***TGbe editor: Modify subclause 12.5.5.3.1 (General) as follows:***

* GCMP cryptographic encapsulation
* General

The GCMP cryptographic encapsulation process is depicted in Figure 12-27 (GCMP encapsulation block diagram(#4087)).



**Figure 12-27—GCMP encapsulation block diagram**

GCMP encrypts the Frame Body field of a plaintext MPDU and encapsulates the resulting cipher text using the following steps:

* Increment the PN, to obtain a fresh PN for each MPDU, so that the PN never repeats for the same temporal key.

NOTE—Retransmitted MPDUs are not modified on retransmission.

* Use the fields in the MPDU header to construct the additional authentication data (AAD) for GCM. The GCM algorithm provides integrity protection for the fields included in the AAD. MPDU header fields that may change when retransmitted are muted by being(#4613) set to a known value when calculating the AAD.
* Construct the GCM nonce(#1406) block (#4613)as defined in 12.5.3.3.4 (Construct CCM nonce(#2720)).
* Construct the GCMP header as defined in 12.5.5.3.5 (Construct GCMP header).(#4613)
* Use the temporal key, AAD, nonce, and MPDU data to form the cipher text and (#4088)the MIC. This step is known as GCM originator processing.
* Form the encrypted MPDU by combining the original MPDU header, the GCMP header, the encrypted data and (#4088)the MIC, as described in 12.5.5.2 (GCMP MPDU format).

***TGbe editor: Modify subclause 12.5.5.3.4 (Construct GCM nonce) as follows:***

* Construct GCM nonce

The Nonce field occupies 12 octets, and its structure is shown in Figure 12-28 (Nonce field(#1406)).

|  |  |  |
| --- | --- | --- |
|  | A2 | PN |
| Octets: | 6 | 6 |
| * Nonce field(#1406)
 |

(#4614) If dot11MultiLinkActivated is true, either To DS or From DS subfields in the MAC header of the MPDU are set to 1, and the MPDU is an individually addressed Data frame, then the A2 subfield shall contain the MLD MAC Address of the transmitting MLD. Otherwise, the A2 subfield shall contain the Address 2 field from the MAC header.

(#4614)The PN subfield shall contain the packet number, with PN0 in the last octet of the subfield.

***TGbe editor: Modify subclause 12.5.5.5.1 (General) as follows:***

* GCMP decapsulation
* General

Figure 12-29 (GCMP decapsulation block diagram(#4087)) shows the GCMP decapsulation process.



**Figure 12-29—GCMP decapsulation block diagram (#4087)**

GCMP decrypts the Frame Body field of a cipher text MPDU and decapsulates a plaintext MPDU using the following steps:

* (#4614)The encrypted MPDU is parsed to construct the AAD (see 12.5.5.3.3 (Construct AAD)) and nonce (see 12.5.5.3.4 (Construct GCM nonce)) values. In addition, if the intended receiver of the MPDU is an MLD and if the MPDU is an individually addressed Data frame, then the intended receiver MLD MAC Address is passed to construct the AAD (see 12.5.5.3.3 (Construct AAD)) and nonce (see 12.5.5.3.4 (Construct GCM nonce)) values.
* The MIC is extracted for use in (#4386)GCM integrity checking.