IEEE P802.11  
Wireless LANs

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | CR PV1 Security | | | | | | Date: 2020-06-03 | | | | | | Author(s): | | | | | | Name | Affiliation | Address | Phone | email | | Matthew Fischer | Broadcom |  |  | [Matthew.fischer@broadcom.com](mailto:Matthew.fischer@broadcom.com) | |  |  |  |  |  | |  |  |  |  |  | |

Abstract

Proposed language to address TGmd D3.0 SA1 CIDs 4416, 4613, 4614 on PV1

Changes are referenced to TGmd D3.2.

**REVISION NOTES:**

**R0**:

Initial

**END OF REVISION NOTES**

Interpretation of a Motion to Adopt

A motion to approve this submission means that the editing instructions and any changed or added material are actioned in the TGmd Draft. This introduction is not part of the adopted material.

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

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

**CIDs**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CID** | **Commenter** | **Clause** | **Page** | **Comment** | **Proposed Change** | **Resolution (Proposed)** |
| 4466 | Mark Rison | 12.5.3.3.1 | 2603.00 | "Otherwise, the priority value of the MPDU is equal to the fixed value 0." -- it should be derived from the TID (in the PTID/Subtype subfield field | As it says in the comment | Reject – the cited text is from a subbullet which lies below an introductory sentence that begins with the qualifier “For secure PV0 MPDUs” and therefore, the PTID case for PV1 MPDUs does not exist. |
| 4613 | Mark Rison | 12.5.3.3.1 | 2602.00 | Significant differences remain between the encryption descriptions for PV0 under a) and PV1 under b).  The asterisks below highlight them: 1)  Increment the PN\*\*\*, to obtain a fresh PN for each MPDU,\*\*\* so that the PN never repeats for the same temporal key. Note that retransmitted MPDUs are not modified on retransmission. v. 1)  When the (Ed)sequence number of the MPDU is less than the previous (Ed)sequence number and satisfies the BPN update conditions in 12.5.3.3.6 (Construct CCMP header for PV1 MPDUs(11ah)) for that TID/ACI, increment the \*\*\*base\*\*\* PN so that the PN never repeats for the same temporal key \*\*\*and TID/ACI\*\*\*. Note that retransmitted MPDUs are not modified on retransmission.  2)  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 masked to 0 when calculating the AAD. v. 2)  Use the fields in the MPDU header to construct the AAD for CCM. The CCM algorithm provides integrity protection for the fields included in the AAD. MPDU header fields that \*\*\*might\*\*\* change when retransmitted are muted by being masked to 0 when calculating the AAD.  3)  Construct the CCM nonce block 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 QC field TID (bits 0 to 3 of the QC 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. 4)  Place the new PN and the key identifier into the 8-octet CCMP header. v. 3)  Construct the CCMP header as defined in 12.5.3.3.6 (Construct CCMP header for PV1 MPDUs(11ah)). If the Type field of the Frame Control field is 001 (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. 4)  Construct the CCM nonce block from the PN, the A2, and the \*\*\*Priority field of the MPDU\*\*\* where A2 is \*\*\*the STA MAC address identified by\*\*\* MPDU Address 2. | Align the wording for a) and b), and also align with the wording for GCMP | Revise - TGmd editor to make changes as shown in 11-20/0877r0 that are marked with CID 4613 which generally agree with the commenter’s suggestion but which provide exact language to be used to make the language of the two procedures as parallel as is possible, and noting that GCMP is not supported by PV1. Note that no explicit prohibition for the use of GCMP with S1G is included in the standard because of legality issues that prevent the explicit exlusion of a specific mechanism. |
| 4614 | Mark Rison | 12.5.3.3.4 | 2606.00 | "(11ah)STA MAC Address Identified By A2 field occupies octets 1-6. This shall be encoded with the octets ordered  with (11ah)STA  MAC Address Identified By A2 octet 0 at octet index 1  and (11ah)STA MAC Address Identified By A2 octet 5 at octet index 6." says nothing of value, but omits the key point of what the field actually contains | Change to "The STA MAC Address Identified By A2 field contains the Address 2 field from the PV0 MPDU or the address derived from the A2 field from the PV1 MPDU (see 9.8.3.2)." | Revise - TGmd editor to make changes as shown in 11-20/0877r0 that are marked with CID 4614 which generally agree with the commenter’s suggestion but retain some of the wording that was discarded by the commenter’s proposed change. |
|  |  |  |  |  |  |  |

**Discussion:**

**CID 466:**

The commenter is asking for the language for the CCMP and GCMP procedure to be made similar for the PV0 and PV1 cases.

The original PV1 text was written based on the template provided by the PV0 language, and so, is, per the TGah committee, as close to the PV0 text as it was possible to make it, assuming that the goal of the TGah committee was to make the PV1 procedure as close to the PV0 procedure as was possible, given the other changes that were made for PV1. I believe that was the goal of the group.

So, a reader who was not part of the TGah committee might ask the question of whether the TGah committee actually did make language for the PV1 case that is as limited in changes as is possible. Let’s examine why the PV1 case is different at all:

1. The PV1 MPDUs were created by TGah to allow a reduction of the MAC header overhead of MPDUs because TGah defined behaviour for operation within a very limited amount of spectrum using rather narrow BW values. Under those conditions, the achievable data rates are much lower than those that most people are accustomed to when referring to 802.11 operation, e.g. 300 kbps, i.e. 0.3 Mbps, and many of the higher rates of operation are still below 10 Mbps. Given these rather low data rates, the goal of reducing MAC header overhead was a laudable one.
2. Some of the obvious places to reduce MAC header overhead are: addresses and the security header
3. The baseline security header (i.e. pre-TGah, pre-PV1, pre S1G) is 8 octets, of which six octets are the PN
4. The PV1 frames reduce the size of the security header by providing only 2 octets of the PN in the MPDU and labelling the other 4 octets as the BASE PN, i.e. BPN which are to be maintained, in synch, at each end of the STA pair
5. The 2 octets of PN that are transmitted with a PV1 MPDU are placed in the SEQ field of the MAC header, so that the size of the security header is reduced to 0 octets
6. For a pair of STAs which are an AP and an associated non-AP STA, one of the six-octet addresses can be substituted with a less-than two octet AID value of the non-AP STA, thereby reducing the MAC header by another 4 octets

Because of these changes for PV1, the CCMP/GCMP procedures are different. However, some improvements can be made:

Step 1), PN increment:

For PV1, there is not a straightforward increment of PN operation, as the SEQ number and PN are now coupled. The SEQ value is obtained outside of the encryption process and based on the SEQ value of the MPDU, the transmitter must decide whether to increment the BPN (not the PN) for each MPDU. The BPN is incremented whenever the SEQ value wraps its space. These steps must be different between PV0 and PV1.

Step 2), AAD

The difference being highlighted here is the use of the fully expanded additiona authentication data (AAD) in the first, PV0 description, and that expansion is not used in the PV1 description.

Note that this reference is the first use of the abbreviation AAD, and as is common through clause 12, the first occurrence of an abbreviation is expanded.

Step 3), NONCE

The PV0 and PV1 steps diverge here as the CCMP header and nonce steps are reversed for the PV1 case.

This is not necessary.

The order was probably reversed because there was mass delirium in TGah such that it was assumed that the CCMP header could not be constructed before the nonce, because nonce construction was needed to determine the PN value which is needed for the CCMP header. However, step 1) already determined the PN value, so the order of the PV1 steps can be reversed to match the order of the PV0 steps.

An extraneous comma in the PV0 case is noted and removed.

Step 4) CCMP header

The steps are different but could be the same.

Wording adjusted.

The remaining steps are the same.

**Proposed Changes to TGmd D3.2:**

**CID 4613**

***TGmd editor: within TGmd D3.2, in 12.5.3.3.1 General, change the text as shown:***

**12.5.3.3 CCMP cryptographic encapsulation**

**12.5.3.3.1 General**

a) For secure PV0 MPDUs, CCMP encrypts the Frame Body field of a plaintext MPDU and encapsulates the resulting cipher text using the following steps:

1) Increment the PN, to obtain a fresh PN for each MPDU, so that the PN never repeats for the same temporal key. Note that retransmitted MPDUs are not modified on retransmission.

2) 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 masked to 0 when calculating the AAD.

3) Construct the CCM nonce block as defined in 12.5.3.3.4 (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. **(#4613)**

4) Construct the CCMP header as defined in 12.5.3.3.5 (Construct CCMP header for PV0 MPDUs).

**(#4613)**

5) Use the temporal key, AAD, nonce, and MPDU data to form the cipher text and MIC. This step is known as CCM originator processing.

6) Form the encrypted MPDU by combining the original MPDU header, the CCMP header, the encrypted data and MIC, as described in 12.5.3.2 (CCMP MPDU format).

b) For secure PV1 MPDUs, CCMP encrypts the Frame Body field of a plaintext MPDU and encapsulates the resulting cipher text using the following steps:

1) When the sequence number of the MPDU is less than the previous sequence number and satisfies the BPN update conditions in 12.5.3.3.6 (Construct CCMP header for PV1 MPDUs(#2720)(11ah)) for that TID/ACI, increment the base PN so that the PN never repeats for the same temporal key and TID/ACI. Note that retransmitted MPDUs are not modified on retransmission.

2) Use the fields in the MPDU header to construct the AAD for CCM. The CCM algorithm provides integrity protection for the fields included in the AAD. MPDU header fields that might change when retransmitted are muted by being masked to 0 when calculating the AAD.

3) Construct the CCM nonce block as defined in 12.5.3.3.4 (Construct CCM nonce), from the PN, the A2, and the priority value of the MPDU where A2 is the STA MAC address identified by MPDU Address 2. If the MPDU is a QoS Data MPDU, the priority value is equal to the PTID subfield of the MPDU. If the Type field of the Frame Control field is 001 (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. **(#4613)**

4) Construct the CCMP header as defined in 12.5.3.3.6 (Construct CCMP header for PV1 MPDUs). **(#4613)**

**(#4613)**

5) Use the temporal key, AAD, nonce, and MPDU data to form the cipher text and MIC. This step is known as CCM originator processing.

6) Form the encrypted MPDU by combining the original MPDU header, the encrypted data, and the MIC, as described in 12.5.3.2 (CCMP MPDU format).

The CCM reference describes the processing of the key, nonce, AAD, and data to produce the encrypted output. See 12.5.3.3.2 (PN processing) to 12.5.3.3.7 (CCM originator processing) for details of the creation of the AAD and nonce from the MPDU and the associated MPDU-specific processing.

**CID 4614**

***TGmd editor: within TGmd D3.2, modify text within 12.5.3.3.4 Construct CCM nonce, as shown:***

**12.5.3.3.4 Construct CCM nonce**

The STA MAC Address Identified By A2 field contains the STA address from the Address 2 field from the PV0 MPDU or the STA address derived from the A2 field from the PV1 MPDU (see 9.8.3.2 (Address fields)) and occupies octets 1–6. The field shall be encoded with the octets ordered with octet 0 of that STA address at octet index 1 and octet 5 of that STA address at octet index 6.

**End of proposed changes.**