IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CR for CID 6053 PV1 Frame Security | | | | |
| Date: 2024-01-09 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| David Goodall | Morse Micro |  |  | dave@morsemicro.com |
| Henry Ptasinski | Morse Micro |  |  | henry@e78com.com |

##### This submission present proposed resolution for the following CID:

6053

##### The proposed changes are based on REVme/D4.2.

##### Revision history:

R0 – Initial version

# CID 6053

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Clause | Page | Line | Comment | Proposed Change |
| 6053 | 12.5.2.3.6 | 2849 | 1 | For PV1 frame security the PN is not sent with the frame and has to be constructed locally at both ends, i.e. at the originator and recipient. Section 12.5.2.3.6 (Construct CCMP header for PV1 MPDUs) describes the PN as being constructed from a BPN stored locally, concatenated with the sequence number field from the MPDU header (fragment field masked out). It also states in one option for the BPN update that: "The locally stored BPN shall be incremented by 1 when the sequence number of the MPDU is less than the previous sequence number for that TID (for Data frames) or ACI (for QMFs) if any of the following two conditions is satisfied:  — Block Ack is not used  — Block Ack is used but decryption occurs after Block Ack reordering"  That should be PTID (partial TID) and PV1 frames don't support QMFs, but those are separate issues. The main issue is that if a PV1 capable STA is supporting multiple traffic streams of different priorities, and also protected management frames, then to maintain the requirement for a single PN the BPN may be incremented by any of the sequence counters being maintained by the originator (for a list see Table 10-5—Transmitter sequence number spaces). This means that the BPN can go out of sync if a frame from sequence number space that rolled over and incremented the BPN at the transmitter is dropped and another frame is sent from a different sequence number space that also rolled over and incremented the BPN. | Modify the requirement in 12.5.2.3.2 (PN processing) that for PV1 MPDUs, the PN shall never repeat for a series of encrypted MPDUs using the same temporal key and (for Data frames) PTID, so that it requires the nonce to never repeat. That will allow a separate BPN to be maintained per PV1 sequence number space. |

Discussion

Section 12.5.2.3.7 (CCM originator processing) states:

“The PN values sequentially number each MPDU. Each transmitter shall maintain a single PN (48-bit counter) for each PTKSA and GTKSA. The PN shall be implemented as a 48-bit strictly increasing integer, initialized to 0 when the corresponding temporal key is initialized or refreshed (by rekeying).”

Therefore, the specification is clear that there is only one packet number (PN) and that applies to both PV0 and PV1 frames. When PV1 frame encryption is in use then STAs exchange only PV1 frames until a Deauthentication or Disassociation frame is transmitted.

Section 12.5.2.3.6 (Construct CCMP header for PV1 MPDUs) states:

“The locally stored BPN shall be incremented by 1 when the sequence number of the MPDU is less than the previous sequence number for that TID (for Data frames) or ACI (for QMFs) if any of the following two conditions is satisfied:

— Block Ack is not used

— Block Ack is used but decryption occurs after Block Ack reordering”

Note that the base packet number (BPN) update is only described for the recipient and not for the originator, so the originator behavior is open to interpretation. Ignoring the issues in the text around use of TID rather than PTID (partial TID) and that PV1 frame encryption does not support QMF, which will be addressed as part of the proposed resolution text, the implication for the originator is that the BPN is updated whenever any sequence number space rolls over, or alternatively there is a single sequence number space covering all PTIDs (for PV1 data frames) and PV1 management frames. If the single BPN can be updated by the rollover of any sequence number space then the recipient BPN may lose synchronization with the originator BPN, for example if the originator updates the BPN and then drops a frame as described in the comment above. The method of resynchronizing via encrypted PV1 Header Compression frames won’t work if the recipient BPN is out of synchonization with the originator BPN. Alternatively, enforcing a single sequence number space would lead to problems in duplicate detection and conflicts directly with Table 10-5 (Transmitter sequence number spaces), which for PV1 capable STAs requires a sequence number space per PTID (8 in total for SNS6), one sequence number space for PV1 management frames (SNS7) and one sequence number space for PV0 management frames (TR1). A PV1 capable STA still uses protected PV0 deauthentication and disassociation frames since there are no PV1 equivalents for these frames.

To address BPN synchronization for encrypted PV1 frames and retain duplicate detection functionality what we need to ensure is that the same nonce, rather than PN, isn't reused. This is consistent with IETF RFC 3610 (Counter with CBC-MAC (CCM)) which states in Section 2.1 that: “Within the scope of any encryption key K, the nonce value MUST be unique.”

The nonce construction for PV1 frames is described in Section 12.5.2.3.4 Construct CCM nonce) for PV1. As shown in Figure 12-21 (see below) the CCM Nonce Flags field has Priority, Management, and PV1 flags, which indicate which of the sequence number spaces is being used (with Priority == PTID for PV1 data frames) and ensure that the nonce is different for each of those spaces. One BPN per PV1 sequence number space (8 for PV1 data frames plus 1 for PV1 management frames) and incrementing the BPN when that sequence number rolls over, would produce unique nonces for all PV1 frames. There would be no change to encrypted PV0 frame behavior.



### Proposed Resolution:

ACCEPTED. Request the TGme editor to apply the changes below:

9.4.2.212 Header Compression element

***Change the text as shown:***

The Header Compression element is used by a STA to inform its intended receiver regarding frame header fields that are compressed and that it needs to store. The format of the Header Compression element is shown in Figure 9-817 (Header Compression element format).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Element  ID | Length | Header Compression Control | A3  (optional) | A4  (optional) | CCMP Update  (optional) |
| Octets: | 1 | 1 | 1 | 0 or 6 | 0 or 6 | 0 or 5 |
| Figure 9-817—Header Compression element format | | | | | | |

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

The Header Compression Control field is shown in Figure 9-818 (Header Compression Control field format).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | B0 | B1 | B2 | B3 | B4 | B5 B7 |
|  | Request/  Response | Store A3 | Store A4 | CCMP Update  Present | PV1 Data Type 3 Supported | Reserved |
| Bits: | 1 | 1 | 1 | 1 | 1 | 3 |
| Figure 9-818—Header Compression Control field format | | | | | | |

The Request/Response subfield is set to 0 to indicate a header compression request and set to 1 to indicate a header compression response.

The Store A3 subfield is set to 1 in the header compression request to request the intended receiver of the frame to store the A3 field and is set to 1 in the header compression response to confirm storage of the A3 field. Otherwise, it is set to 0 in the header compression request to indicate no storage request for the A3 field and is set to 0 in the header compression response to indicate unsuccessful storage or release of the stored A3 field.

The Store A4 subfield is set to 1 to request the intended receiver of the header compression request to store the A4 field and is set to 1 in the header compression response to confirm storing of the A4 field. Otherwise, it is set to 0 in the header compression request to indicate no storage request for the A4 field and is set to 0 in the header compression response to indicate unsuccessful storage or release of the stored A4 field.

The CCMP Update Present subfield is set to 1 to indicate the intended receiver of the header compression request to update the base packet number (BPN) and Key ID fields for the specified sequence number (SN) space, as identified by the PTID~~/ACI~~ subfield and Management subfield in the CCMP Update field, and is set to 1 in the header compression response to confirm the update of the fields or to indicate decryption error for the specified ~~TID/ACI~~SN space. Otherwise, it is set to 0 in the header compression request to indicate no CCMP update request and is set to 0 in the header compression response to indicate no CCMP update confirmation.

NOTE 1—See Table 10-5 (Transmitter sequence number spaces) for SN space selection.

The PV1 Data Type 3 Supported subfield is set to 1 to indicate that reception of PV1 frames with Type field equal to 3 is enabled. Otherwise, it is set to 0.

The A3 field in the Header Compression element is present if the Request/Response subfield is 0 and the Store A3 subfield is 1. Otherwise, it is not present.

The A4 field in the Header Compression element is present if the Request/Response subfield is 0 and the Store A4 subfield is 1. Otherwise, it is not present.

The CCMP Update field in the Header Compression element is present if the CCMP Update Present subfield is 1. Otherwise, it is not present.

The CCMP Update field contains the BPN and key ID for the specified SN space~~a given TID/ACI~~, as shown in Figure 9-819 (CCMP Update field format).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | B0 B31 | B32 B33 | B34 B36~~7~~ | ‍B37 | B38 B39 |
|  | BPN | Key ID | PTID~~/ACI~~ | Management‍ | Reserved |
| Bits: | 32 | 2 | ~~4~~3 | ‍1 | 2 |
| Figure 9-819—CCMP Update field format | | | | | |

The BPN subfield contains the BPN for the ~~TID/ACI~~specified SN space ~~in the CCMP Update field~~. The BPN subfield consists of the PN2, PN3, PN4, and PN5 octets, as shown in Figure 9-820 (BPN subfield format).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | B0 B7 | B8 B15 | B16 B23 | B24 B31 |
|  | PN2 | PN3 | PN4 | PN5 |
| Bits: | 8 | 8 | 8 | 8 |
| Figure 9-820—BPN subfield format | | | | |

The Key ID subfield contains the key ID for the ~~TID/ACI~~specified SN space ~~included in the CCMP Update field~~.

The PTID~~/ACI~~ subfield contains the PTID~~/ACI~~ for ~~which the BPN and the Key ID subfields apply~~the specified SN space when the Management subfield value is 0, and is set to 0 otherwise.

The Management subfield is set to 1 to identify the specified SN space as SNS7, which is for individually addressed PV1 Management frames, and is set to 0 otherwise.

10.3.2.14.2 Transmitter requirements

***Update the entry for SNS6 in Table 10-5 as shown:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SNS6 | Individually  addressed  PV1 Data  frame | A STA operating as an S1G  STA transmitting a PV1 Data  frame | Mandatory | Indexed by  <STA MAC  Address  identified by  Address 1,  PTID> | TR4 |

***Add TR4 to the end of Table 10-5 as shown:***

|  |
| --- |
| TR3: Sequence numbers for transmitted QoS (+)Null frames may be set to any value.  TR4: The same modulo 1024 counter per <STA MAC Address identified by Address 1, PTID> tuple shall be used for both PV1 Type 0 (QoS Data) and Type 3 (QoS Data) frames. |

10.57 Generation of PV1 MPDUs and header compression procedure

***For the paragraph starting at 2386.10 make the changes shown:***

An S1G STA indicates a request to store address fields by sending a header compression request with the Store A3 and/or Store A4 subfields equal to 1. Upon receipt of such a request, the receiving STA shall respond with a header compression response indicating which of the optional fields it stores, by setting the Store A3 and/or Store A4 subfields in the transmitted header compression response to 1. Stored address fields can subsequently be omitted from the MAC header of PV1 frames transmitted by the STA that sent the header compression request. Address A3 and/or A4 fields for which the header compression response indicated 0 are not stored at the receiver and cannot be omitted by the transmitter when the A3 and/or A4 fields contain values that are different from the A1 and/or A2 fields of the same. The receiver stores one A3 and one A4 per transmitter. The stored A3 and A4 addresses shall be used at the receiver for all frames from the transmitter that omit the stored addresses.

***For the paragraphs starting at 2386.44 make the changes shown:***

A STA that receives a PV1 frame with one or more compressed addresses that it has not stored or which causes a decryption error should transmit an unsolicited header compression response to the transmitter of the PV1 frames, in which the Store A3, and Store A4 fields are all equal to 0. The unsolicited header compression response shall ~~include the TID/ACI~~specify the SN space of the received PV1 frame using~~in~~ the PTID~~/ACI~~ subfield and Management subfield of the CCMP Update field if the received frame caused a decryption error, where the CCMP Update field shall indicate the stored values for the BPN and key ID that correspond to the received PV1 frame.

A STA that has previously transmitted PV1 frames of a given ~~TID/ACI~~SN space to a peer STA and that receives an unsolicited header compression response from the peer STA relative to that ~~TID/ACI~~SN space shall transmit a header compression request to the transmitter of the header compression response. The header compression request shall include all the addresses that the transmitting STA requests to be stored at the receiver and/or the security information that corresponds to the indicated ~~TID/ACI~~SN space.

12.5.2.3.2 PN processing

***Change the first paragraph as shown:***

The PN is incremented by a positive number for each MPDU. The PN shall be incremented in steps of 1 forconstituent MPDUs of fragmented MSDUs, A-MSDUs, and MMPDUs. For PV0 MPDUs, the PN shall never repeat for a series of encrypted MPDUs using the same temporal key. For PV1 MPDUs, the PN shall never repeat for a series of encrypted MPDUs using the same temporal key and SN space~~(for Data frames) PTID~~.

12.5.2.3.6 Construct CCMP header for PV1 MPDUs

***Change the text as shown:***

The CCMP header is not present in secure PV1 MPDUs, but constructed locally at the STA as follows:

* The PN is composed of the Sequence Control field and the base PN (BPN) for the SN space of the MPDU, where
* The Sequence Control field is present in the MPDU header
* PN0||PN1 = Sequence Control field ~~with the Fragment Number subfield masked out when the PV1 MPDU is carried in an A‑MPDU that is not an S-MPDU~~
* The base PN is retrieved from the local storage at the receiver
* PN2||PN3||PN4||PN5 = BPN
* PN = PN0||PN1|| PN2||PN3||PN4||PN5 (= Sequence Control ||BPN)
* The key ID is retrieved from the local storage at the receiver

The locally stored BPNs and key ID are initialized at 0 when the PTKSA is established.

The locally stored BPN for the SN space of the MPDU shall be incremented by 1 when the sequence number of the MPDU is less than the previous sequence number for that ~~TID (for Data frames) or ACI (for QMFs)~~SN space if any of the following two conditions is satisfied:

* Block Ack is not used
* Block Ack is used but decryption occurs after Block Ack reordering

When Block Ack is used and decryption occurs before Block Ack reordering, the BPN for the SN space of the MPDU may be updated as follows. The receiver maintains a sequence number window of size *w*, which is equal to twice the the block ack agreement reordering buffer. The sequence number window has a lower edge *a* and an upper edge *b*. For a received sequence number SN (as part of the received Sequence Control field), the associated packet number (PN) is determined as follows (where *b* is initialized as *b* = 0):

* if (*b* ≥ w) then
* *a* = *b* − w
* if (SN < *a*) then BPN = BPN + 1
* PN = Sequence Control ||BPN
* if not (*a* < SN < *b*) then *b* = SN
* else (i.e., *b* < w) then
* *a* = *b* − w + 212
* if (SN < *a*) then PN = Sequence Control ||BPN
* if (SN ≥ *a*) then PN = Sequence Control ||(BPN − 1)
* if (b < SN < *a*) then *b* = SN

The BPN can also be updated explicitly through a header compression request/response frame exchange, as defined in 10.57 (Generation of PV1 MPDUs and header compression procedure).

12.5.2.3.7 CCM originator processing

***Change the fifth paragraph as shown:***

The PN values sequentially number each MPDU. For each PTKSA and GTKSA, e~~E~~ach transmitter shall maintain a single PN (48-bit counter) for PV0 frames~~each PTKSA and GTKSA~~. If dot11PV1MACHeaderOptionImplemented is true, the transmitter shall additionally maintain a PN for each PV1 SN space. The PNs shall be implemented as ~~a~~ 48-bit strictly increasing integers, initialized to 0 when the corresponding temporal key is initialized or refreshed (by rekeying).

12.6.20.3 Transparent multi-band RSNA

***Change the 3rd paragraph as shown:***

A STA shall use a single PN counter (12.5.2.3 (CCMP cryptographic encapsulation) and 12.5.4.3 (GCMP cryptographic encapsulation)) for transmission in both the current operating band and the other supported band(s) when transparent multi-band RSNA is used. A STA shall not use PV1 frames when using transparent multi-band RSNA.

**References:**