IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REVmd security comments | | | | |
| Date: 2018-07-10 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Jouni Malinen | Qualcomm, Inc. |  |  | jouni@qca.qualcomm.com |
|  |  |  |  |  |

Abstract

This document discusses some REVmd comments and proposes resolutions for them.

## CID 1440

**Clause**: 9.3.3.15 (Action No Ack frame format)

D1.0 page 824 line 8

**Comment**: There is no reason Action No Acks can't have MICEs. While at the moment it may be the case that such frames carry "information [...] that is of time critical but transient value" (resolution of CID 6343 in mc), this is not a property of Action No Acks per se. The resolution of CID 320 was to reject this comment because "There is no need to cryptographically authenticate such data, until such uses are added, and a MIC element can be added at that time." but adding this later might lead to backward-compatibility issues

**Proposed Change**: Replace the last row of Table 9-43 with the last four rows (including the NOTE, i.e. from the "Last - 2" row) from Table 9-42

**Context in D1.0**:





**Discussion**: It should be noted that the “MICE” and “MIC element” in the comment seem to be referring to the element call the Management MIC element (MME) in the current standard, i.e., this is related to use of BIP to protect robust group addressed Management frames. While the topic of this comment has been previously discussed and the conclusion may be accurate in the sense that no existing use case of Action No Ack frame describe in the standard would use BIP, it would sound reasonable to describe same BIP behavior for Action frames and Action No Ack frames. It should also be noted that the Vendor-specific Protected Category value 126 in Table 9-53 (Category values) indicate that Action frames and Action No Ack frames using the Category value would use BIP. While Vendor-specific Protected (or Vendor-specific for that matter) is not currently described explicitly as allowing to use the Action No Ack frame subtype, there might be a need to be able to use a vendor specific Action No Ack frame that would need BIP, but Table 9-43 discussed in this comment would claim that BIP cannot be used there.

**Proposed resolution**: Accepted.

## CID 1404

**Clause**: 11.3.4.3 (Authentication—destination STA)

D1.0 page 2018 line 1

**Comment**: It is not sufficiently clear what what happens if the STA is in an IBSS and management frame protection was negotiated, specifically when the PTKSA etc. do get deleted

**Proposed Change**: Add after e) a "NOTE---If management frame protection has been negotiated the SAs are never deleted."

**Context in D1.0**:

Upon receipt of an Authentication frame with authentication transaction sequence number equal to 1, the

destination STA shall authenticate with the originating STA using the following procedure:

a) If Open System or Shared Key authentication algorithm is being used, the STA shall execute the procedure described in 12.3.3.2 (Open System authentication) or 12.3.3.3 (Shared Key authentication), respectively. These result in the generation of an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request.

b) If FT authentication is being used, the MLME shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request, including the FT Authentication Elements, and the SME shall execute the procedure as described in 13.5 (FT protocol) or 13.6 (FT resource request protocol).

c) If SAE authentication is being used in an infrastructure BSS, IBSS, or MBSS, the MLME shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request, including the SAE authentication elements, and the SME shall execute the procedure as described in 12.4 (Authentication using a password)(Ed) .

d) If FILS authentication is being used, the MLME shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request, and the SME shall execute the procedure described in 12.12 (Authentication for FILS(11ai))(11ai) .

e) If the STA is in an IBSS and management frame protection was not negotiated when the PTKSA(s) were created, the SME shall delete any PTKSA, GTKSA, IGTKSA and temporal keys held for communication with the originating STA by using the MLME-DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)).

NOTE—If management frame protection has been negotiated the SAs are never deleted.

f) Upon receipt of an MLME-AUTHENTICATE.response primitive, if the ResultCode is not SUCCESS, the MLME shall transmit an Authentication frame with the corresponding status code, as defined in 9.4.1.9 (Status Code field), and the state for the originating STA shall be left unchanged. The Authentication frame is constructed using the appropriate procedure in 12.3.3.2 (Open System authentication), 12.3.3.3 (Shared Key authentication), 13.5 (FT protocol) or 13.6 (FT resource request protocol).

g) Upon receipt of an MLME-AUTHENTICATE.response primitive, if the ResultCode is SUCCESS, the MLME shall transmit an Authentication frame that is constructed using the appropriate procedure in 12.3.3.2 (Open System authentication), 12.3.3.3 (Shared Key authentication), 13.5 (FT protocol) or 13.6 (FT resource request protocol), with a status code of SUCCESS, and the state for the originating STA shall be set to State 2 if it was in State 1.

If the STA is in an IBSS, if the SME decides to initiate an RSNA, and if the SME does not know the security policy of the peer, it may issue an individually addressed Probe Request frame to the peer by invoking an MLME-SCAN.request primitive to discover the peer’s security policy. When a non-AP STA receives an Authentication frame that includes an Association Delay Info element, the non-AP STA sets the dot11AssociationResponseTimeOutequal to or larger than the value of the Association Delay Info field(11ai) .

**Discussion**: SAs need to be deleted at some point; at latest when a STA leaves the IBSS. As such, the proposed language “the SAs are never deleted” does not look correct. Furthermore, this subclause is not really the place to describe when SAs are deleted in general; this subclause provides rules on the destination of an Authentication frame exchange for updating local state, e.g., by setting State and deleting a specific SA as a result of the authentication exchange.

While the current standard does not explicitly describe a use case for the SA Query procedure in IBSS, that exchange might be used by a STA to detect that there is a mismatch in SAs between two STAs in the IBSS, e.g., triggered by reception of an unprotected EAPOL frame (that would be sent following this Authentication exchange that did not delete SAs). If such SA Query procedure does not result in receiving a valid response, the STA that initiated the SA Query could delete the SA. This would then allow EAPOL frame exchange to be used to derive a new SA and allow the STAs to recover from broken connectivity even when management frame protection is used in IBSS.

**Proposed resolution (alternative 1)**: Rejected. It is not correct to say that SAs are never deleted since they do get deleted, e.g., when leaving the IBSS or when the SA expires. The procedure described in the cited subclause is unambiguous on stating that the particular authentication exchange does not delete the SAs, but that does not rule out other triggers for deleting SAs.

**Proposed resolution (alternative 2)**: Revised. Add following at the end of the item e: “NOTE—If management frame protection was negotiated when the PTKSA(s) were created, the SME does not delete any of the previously created SAs or temporal keys.”

## CID 1402

**Clause**: 11.3.4.3 (Authentication—destination STA)

D1.0 page 2018 line 1

**Comment**: It is not sufficiently clear what happens if a STA receives an Auth frame on a PMF link. It needs to ignore it (no key changes and no auth/assoc state changes) and wait until the association step to determine whether it's a legitimate request. This includes FT and SAE auth

**Proposed Change**: Add a new step before a): "If management frame protection has been negotiated the STA shall ignore the Authentication frame and make no changes."

**Context in D1.0**:

Upon receipt of an Authentication frame with authentication transaction sequence number equal to 1, the

destination STA shall authenticate with the originating STA using the following procedure:

a) If management frame protection has been negotiated the STA shall ignore the Authentication frame and make no changes.

b) If Open System or Shared Key authentication algorithm is being used, the STA shall execute the procedure described in 12.3.3.2 (Open System authentication) or 12.3.3.3 (Shared Key authentication), respectively. These result in the generation of an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request.

c) If FT authentication is being used, the MLME shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request, including the FT Authentication Elements, and the SME shall execute the procedure as described in 13.5 (FT protocol) or 13.6 (FT resource request protocol).

d) If SAE authentication is being used in an infrastructure BSS, IBSS, or MBSS, the MLME shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request, including the SAE authentication elements, and the SME shall execute the procedure as described in 12.4 (Authentication using a password)(Ed) .

e) If FILS authentication is being used, the MLME shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request, and the SME shall execute the procedure described in 12.12 (Authentication for FILS(11ai))(11ai) .

f) If the STA is in an IBSS and management frame protection was not negotiated when the PTKSA(s) were created, the SME shall delete any PTKSA, GTKSA, IGTKSA and temporal keys held for communication with the originating STA by using the MLME-DELETEKEYS.request primitive (see 12.6.18 (RSNA security association termination)).

g) Upon receipt of an MLME-AUTHENTICATE.response primitive, if the ResultCode is not SUCCESS, the MLME shall transmit an Authentication frame with the corresponding status code, as defined in 9.4.1.9 (Status Code field), and the state for the originating STA shall be left unchanged. The Authentication frame is constructed using the appropriate procedure in 12.3.3.2 (Open System authentication), 12.3.3.3 (Shared Key authentication), 13.5 (FT protocol) or 13.6 (FT resource request protocol).

h) Upon receipt of an MLME-AUTHENTICATE.response primitive, if the ResultCode is SUCCESS, the MLME shall transmit an Authentication frame that is constructed using the appropriate procedure in 12.3.3.2 (Open System authentication), 12.3.3.3 (Shared Key authentication), 13.5 (FT protocol) or 13.6 (FT resource request protocol), with a status code of SUCCESS, and the state for the originating STA shall be set to State 2 if it was in State 1.

If the STA is in an IBSS, if the SME decides to initiate an RSNA, and if the SME does not know the security policy of the peer, it may issue an individually addressed Probe Request frame to the peer by invoking an MLME-SCAN.request primitive to discover the peer’s security policy. When a non-AP STA receives an Authentication frame that includes an Association Delay Info element, the non-AP STA sets the dot11AssociationResponseTimeOutequal to or larger than the value of the Association Delay Info field(11ai) .

**Discussion**: It is not correct to say that the Authentication frame is ignored. The frame needs to be processed and that processing may actually include significant operations, e.g., when using SAE authentication, FILS authentication, or FT protocol. When there exists an association that has negotiated management frame protection, State and SAs are not allowed to be dropped to avoid DoS attacks.

**Proposed resolution (alternative 1)**: Rejected. Authentication frame needs to be processed even when there exists an association that has negotiated management frame protection, e.g., to perform SAE authentication. As such, the Authentication frames cannot be ignored. SAs are not delete and State is not dropped before confirming that the peer STA is indeed able to complete the authentication and key management steps. The current standard describes this and this subclause does not require additional procedure steps to address this.

**Proposed resolution (alternative 2)**: Revised. Add the following before the final paragraph of 11.3.4.3 (D1.0 page 2018 line 44): “NOTE—If management frame protection was negotiated, the SME does not change the state for the originating STA and does not delete any of the previously created SAs or temporal keys as a part of this authentication procedure.”

## CID 1274

**Clause**: 9.4.2.24.2 (Cipher suites)

D1.0 page 1020 line 27

**Comment**: RSNE description of invalid data cipher suites was not updated when the new BIP ciphers were added. All four BIP ciphers are unsuitable as data ciphers. Note that an incorrect name for the BIP-CMAC-128 cipher was used here, so that is also fixed in the proposed change. I have a separate comment addressing that misnaming with an identical change in this location.

**Proposed Change**: Replace "Use of AES-128-CMAC is not valid as a data cipher suite" with "Use of BIP-CMAC-128, BIP-GMAC-128, BIP-GMAC-256, and BIP-CMAC-256 is not valid as a data cipher suite"

**Context in D1.0**:

The Group Data Cipher Suite field contains the cipher suite selector used by the BSS to protect group

addressed frames.

The Pairwise Cipher Suite Count field indicates the number of pairwise cipher suite selectors that are

contained in the Pairwise Cipher Suite List field. The value 0 is reserved.

The Pairwise Cipher Suite List field contains a series of cipher suite selectors that indicate the pairwise

cipher suites.

The Group Management Cipher Suite field contains the cipher suite selector used by the BSS to protect

group addressed robust Management frames.

When management frame protection is negotiated, the negotiated pairwise cipher suite is used to protect

individually addressed robust Management frames, and the group management cipher suite is used to protect

group addressed robust Management frames. Use of BIP-CMAC-128, BIP-GMAC-128, BIP-GMAC-256, and BIP-CMAC-256 is not valid as a data cipher suite.

A suite selector has the format shown in Figure 9-280 (Suite selector format).

**Discussion**: The comment identifies a missing detail and identifies the exact change in the text (shown in redline above).

**Proposed Resolution**: Accepted.

## CID 1259

**Clause**: 10.54 (Generation of PV1 MPDUs and header compression procedure)

D1.0 page 1931 line 9

**Comment**: Header compression from P802.11ah uses BPN to construct the PN for CCMP/GCMP. As such, it is critical for BPN never to be decremented for the same key since such decrementation could result in nonce reuse with CCMP/GCMP. While 802.11ah added a statement saying that a PN shall not be used multiple times, with the same key, it did not add any rules on processing the header compression update frames as is related to BPN updates.

**Proposed Change**: Replace "An S1G STA indicates a request to update security parameters by sending a header compression request with the CCMP Update subfield equal to 1. The receiver STA shall respond with a header compression response acknowledging receipt of the updated security parameters." with "An S1G STA indicates a request to update security parameters by sending a header compression request with the CCMP Update subfield equal to 1. The receiver STA shall respond with a header compression response acknowledging receipt of the updated security parameters. If the received BPN value would decrement the currently stored BPN value, the receiver STA shall ignore the security parameter update."

**Context in D1.0**:

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.

An S1G STA indicates a request to update security parameters by sending a header compression request with the CCMP Update subfield equal to 1. The receiver STA shall respond with a header compression response acknowledging receipt of the updated security parameters. If the received BPN value would decrement the currently stored BPN value, the receiver STA shall ignore the security parameter update.

**Discussion**:

BPN = base PN

PN is constructed using the following definition when header compression is used: PN = PN0||PN1|| PN2||PN3||PN4||PN5 (= SC||BPN) where PN0||PN1 = SC and PN2||PN3||PN4||PN5 = BPN

12.5.3.3 describes how BPN can be incremented (by one) on the transmitter and on the receiver. 10.54 describes explicit mechanism for updating BPN through a header compression request/response exchange.

If a header compression request would be allowed to decrement the value of the locally stored BPN on the receiver, the receiver would effectively allow replays of previously used PN values (PN=SC||BPN) which is obviously not desired due to the implications of CCM/GCM nonce replay. As the commenter indicates, P802.11ah does include language talking about the same PN value not being reused, but that seems to be clear mainly on the transmitter side and even if this were to implicitly place in some rules on the receiver to verify that the same PN has not been used with the same key, such validation cannot really be expected to be implemented by remembering all used PN values due to the resource requirements such an implementation would have. The receiver side protection should be done more efficiently and for header compression request/response exchange, adding a constraint on the new BPN sounds like an appropriate way of doing this. It might also make sense to state when BPN is updated more explicitly and also to make it clear that the response frame is sent out regardless of whether the security parameter update is “ignored”.

**Proposed resolution**: Revised. Add to the end of the “An S1G STA indicates a request to update security parameters by sending a header compression request with the CCMP Update subfield equal to 1. The receiver STA shall respond with a header compression response acknowledging receipt of the updated security parameters.” paragraph (D1.0 page 1931 line 10): “The receiver STA shall compare the received BPN value to the current locally stored BPN value. If the received BPN value is greater than the locally stored BPN value, the receiver STA shall update the locally stored BPN value. Otherwise, the receiver STA shall not update the locally stored BPN value.”

## CID 1258

**Clause**: 10.54 (Generation of PV1 MPDUs and header compression procedure)

D1.0 page 1930 line 16

**Comment**: Security of PV1 MPDUs and header compression (from P802.11ah) with RSN enabled depends on the Header Compression frames being protected. If they are not protected, an attacker could inject arbitrary Header Compression frames to update the header compression information (address mapping) and CCMP packet number (through BPN modification). Either of those could invalidate security properties of CCMP (or GCMP for that matter). Since the Header Compression frame is a robust Action frame, this frame can be protected by mandating use of management frame protection whenever an S1G STA supports header compression and uses RSN.

**Proposed Change**: Add following to the end of the "The header compression procedure enables S1G STAs to store addresses and/or update security parameters at the receiver." paragraph: "An S1G STA with dot11PV1MACHeaderOptionImplemented equal to true shall negotiate use of management frame protection for RSNA."

**Context in D1.0**:

The header compression procedure enables S1G STAs to store addresses and/or update security parameters at the receiver. An S1G STA with dot11PV1MACHeaderOptionImplemented equal to true shall negotiate use of management frame protection for RSNA.

**Discussion**: The current standard text does not seem to have explicit requirement for using management frame protection whenever using RSNA for S1G. If RSNA is used without it, header compression request/response exchange would open a replay vulnerability unless the receiver is somehow providing undefined extra protection to prevent PN reuse. The proposed resolution for CID 1259 addresses the replay vulnerability, but it would still leave possibility for a DoS issue where header compression request frame could be used to increment BPN to significantly larger value which would practically make the receiving STA drop all following header compressed Data frames as replays (received PN < PN generated from locally stored BPN). It looks like the standard should either disallow use of header compression in RSNA without management frame protection or mandate use of management frame protection when header compression is used. The comment proposes the latter.

**Proposed resolution**: Accepted.

## CID 1029

**Clause**: 10.54 (Generation of PV1 MPDUs and header compression procedure)

D1.0 page 1931 line 7

**Comment**: Could the Header Compression request/response be used by an attacker to compromise security (particularly if it could reset the CCMP counters)

**Proposed Change**: Add some normative requirement in the header compression request/response to indicate that the transmitter shall set counters to a greater value in the response.

**Context in D1.0**:

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.

An S1G STA indicates a request to update security parameters by sending a header compression request with the CCMP Update subfield equal to 1. The receiver STA shall respond with a header compression response acknowledging receipt of the updated security parameters.

**Discussion**: CID 1259 proposed text to add normative requirement on the receiver to not allow BPN to be decremented. As far as this CID 1029 is concerned, it should be noted that an unsolicited header compression response is used to indicate locally stored BPN when processing of a received Data frame using header compression failed due to a decryption error. It is not clear what the “a greater value in the response” refers to in the comment.. The behavior for the transmitted of both the header compression request and response frames seems to be already defined in normative language (BPN value is the locally stored BPN). As such, it does not look like there would be a need to add any more requirements on the transmitter (and CID 1259 addressed receiver where new normative language seems useful to add).

**Proposed resolution**: Rejected. The transmitter behavior for setting the BPN value in the Header Compression Request/Response frame is already defined in normative language to use the locally stored BPN value.