IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| On FrAttacks and related matters | | | | |
| Date: 2021-08-09 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Mark RISON | Samsung Cambridge Solution Centre | SJH, CB4 0DS, U.K. | +44 1223 434600 | at samsung (a global commercial entity) I'm the letter emme then dot rison |
| Mathy Vanhoef | KU Leuven / NYUAD | Leuven, Belgium |  | mathy.vanhoef at nyu.edu or at kuleuven.be |
| Mark Hamilton | Ruckus/CommScope | 350 W Java Drive  Sunnyvale, CA 94089 | +1-303-818-8472 | mark.hamilton@commscope.com |
| Jouni Malinen | Qualcomm, Inc. |  |  | jouni@qca.qualcomm.com |

Abstract

This submission discusses various considerations regarding vulnerabilities related to fragmentation, prompted by <https://papers.mathyvanhoef.com/usenix2021.pdf>, and related matters.

Green highlight indicates material agreed to in the group, yellow material to be discussed, red material rejected by the group and cyan material not to be overlooked. The “Final” view should be selected in Word.

Discussion:

<https://papers.mathyvanhoef.com/usenix2021.pdf> has identified a number of design and implementation issues related to fragmentation/defragmentation in RSNs. Thinking around them, further issues can be identified. All these ought to be stated or emphasised in the standard:

* When the standard says you increment by exactly 1 for the PNs of fragments this is not just a serving suggestion!
* MSDUs (and MMPDUs if MFP is in use) must not be reassembled from fragments with different keys, or from unprotected fragments. This includes:
  + Not using fragments from a previous association/PTK
  + Not combining unprotected initial fragments of an MSDU/MMPDU before the 4WH with protected final fragments of that MSDU/MMPDU after the 4WH (possible if different EDCA queues are used)
  + Not using multiple key( ID)s for a given MSDU/MMPDU under “Extended Key ID for Individually Addressed Frames”
* A-MSDUs must not be fragmented
  + Note A-MSDUs may be (“dynamic”ally) fragmented under 802.11ax-2021 (which is not in the scope of 802.11me/D0.0). When this is rolled in, it should be made clear that the A-MSDU Present bit must not change for a set of QoS Data frames with the same SN. Something like this might do, w.r.t. 802.11ax/D8.0/26.3.3.1:

Defragmentation of dynamic fragments shall follow the rules defined in 10.5 (MSDU, A-MSDU, and MMPDU defragmentation) with the following exceptions:

[…]

- The A-MSDU Present subfield of all QoS Data frames containing fragments with the same sequence number and TID shall be the same.

and a corresponding NOTE in 10.5 after the second para:

NOTE—All the fragments of an MSDU or A-MSDU are required to have the same indication of whether they are fragments of an MSDU or an A-MSDU (see 26.3.3.1).

The following should also be stated or emphasised:

* EAPOL PDUs must not be forwarded
* EAPOL PDUs must not be groupcast
* The controlled port must stay blocked/closed until the 4WH completes
* Unprotected Data frames (whether groupcast or not, and whether fragments or not) must be discarded, except EAPOL PDUs during the initial 4WH (but not a rekeying 4WH)
  + Similarly with unprotected Management frames if MFP is used
* Retransmissions of M3 are tricky
  + Need to allow for an M3 retx during the initial 4WH that will be unencrypted (and potentially fragmented too), even though by then the key will have been installed at the Supplicant
  + Need to allow for an M3 retx during a rekeying 4WH that will be encrypted with the old key (and potentially fragmented too), even though by then the new key will have been installed at the Supplicant

Proposed changes:

In 10.4 MSDU and MMPDU fragmentation add after the third para (“A fragment is an MPDU […]”):

NOTE—Packet numbers for fragments in an RSNA are required to be consecutive (see 12.5.3.3.2 (PN processing) and 12.5.5.3.2 (PN processing)).

In 10.5 MSDU and MMPDU defragmentation add at the end of the second para (“The destination STA shall reconstruct the MSDU or MMPDU […]”):

An MSDU or MMPDU shall not be reassembled from fragments that were encrypted with different keys, or from a mixture of encrypted and unencrypted fragments.

NOTE—Packet numbers for fragments in an RSNA are required to be consecutive (see 12.5.3.4.4 (PN and replay detection) and 12.5.5.4.4 (PN and replay detection) under d).

Change 12.6.14 RSNA key management in an infrastructure BSS as follows:

The Supplicant and Authenticator signal the completion of key management by utilizing the MLME-SETKEYS.request primitive to configure the agreed-upon temporal pairwise key into the IEEE 802.11 MAC and by calling the MLME-SETPROTECTION.request primitive to enable its use. Any MSDU fragments previously received under the corresponding Key ID, and if management frame protection is in use, any MMPDU fragments previously received under the corresponding Key ID, shall be discarded at this point.

Change 12.6.21 RSNA rekeying as follows:

When both ends of the link support extended Key IDs for individually addressed frames, it is possible to install the new PTKSA without data loss, provided the new PTKSA uses a different Key ID from the old PTKSA. Data loss might occur if the same Key ID is used because it is not possible to precisely coordinate (due to software processing delays) when the new key is used for transmit at one end and when it is applied to receive at the other end. If a different Key ID is used for the new PTKSA, then provided the new key is installed at the receive side prior to its first use at the transmit side there is no need for precise coordination. During the transition, received packets are unambiguously identified using the Key ID as belonging to either the old or new PTKSA. The same Key ID shall be used for all fragments of a given MSDU or, if management frame protection is in use, MMPDU.

Change 12.6.9 RSN management of the IEEE 802.1X Controlled Port as follows:

~~When the policy selection process chooses IEEE 802.1X authentication~~In an RSN, this standard assumes that IEEE 802.1X Supplicants and Authenticators exchange protocol information via the IEEE 802.1X Uncontrolled port. The IEEE 802.1X Controlled Port is blocked from passing general data traffic between the STAs until an IEEE 802.1X authentication procedure completes successfully over the IEEE 802.1X Uncontrolled Port. The security of an RSNA depends on this assumption being true.

This standard assumes each Controlled Port remains blocked until the IEEE 802.1X state variables portValid and keyDone both become true. This assumption means that the IEEE 802.1X Controlled Port discards MSDUs sent across the IEEE 802.11 channel prior to the installation of cryptographic keys into the MAC. This protects the STA’s host from forged MSDUs written to the channel while it is still being initialized.

NOTE—This means that Data frames other than those containing EAPOL PDUs are discarded when received before the initial 4-way handshake (see 12.7.6) completes, and that unprotected Data frames (other than those containing retransmissions of the third message of the initial 4-way handshake (see 12.7.6.6 (4-way handshake implementation considerations)) are discarded when received after the initial 4-way handshake completes.

The MAC does not distinguish between MSDUs for the Controlled Port, and MSDUs for the Uncontrolled Port. In other words, EAPOL-Start frames and EAPOL-Key frames are encrypted and decrypted ~~only~~ after invocation of the MLME-SETPROTECTION.request primitive.

NOTE—An Authenticator might retransmit the third message of the 4-way handshake (see 12.7.6.6 (4-way handshake implementation considerations)). In the initial 4-way handshake this third message (and the fourth message sent in response) will be unprotected and in a rekeying 4-way handshake the third (and fourth) message will be protected with the old key. If the ends of the link do not both support extended Key IDs for individually addressed frames, the mechanism by which a Supplicant might accept and respond to this retransmission of the third message if the MLME-SETKEYS.request and MLME-SETPROTECTION.request primitives have already been invoked is outside the scope of this standard.

This standard assumes that IEEE Std 802.1X-2010 does not block the Controlled Port when authentication is triggered through reauthentication. During IEEE 802.1X reauthentication, an existing RSNA can protect all MSDUs exchanged between the STAs. Blocking MSDUs is not required during reauthentication over an RSNA.

EAPOL PDUs shall not be delivered to the Controlled Port.

NOTE—This means that an AP does not forward EAPOL PDUs received from a STA to any other STA in the BSS or ESS, to the portal, to the attached bridge port, or to a local higher layer other than the PAE.

EAPOL PDUs shall be carried in individually addressed MSDUs.

Change 11.3.5.2 Non-AP and non-PCP STA association initiation procedures as follows:

i) Upon receipt of the MLME-SETPROTECTION.request(Rx\_Tx) primitive, the MLME shall set the state of the STA to State 4.

NOTE—Any MSDU fragments from the AP, and if management frame protection is in use, any MMPDU fragments from the AP, are discarded at this point (see 12.6.14), since fragments are required to be encrypted with the same key (see 10.5).

Change 11.3.5.3 AP or PCP association receipt procedures as follows:

p) If the ResultCode in the MLME-ASSOCIATE.response primitive is SUCCESS and RSNA establishment is required, and FILS authentication was not used, the SME shall attempt a 4-way handshake. Upon a successful completion of the 4-way handshake, the SME shall enable protection by issuing an MLME-SETPROTECTION.request(Rx\_Tx) primitive. If FILS authentication was used, the SME shall enable protection by generating an MLME-SETPROTECTION.request(Rx\_Tx) primitive. In either case, upon receipt of the MLME-SETPROTECTION.request(Rx\_Tx) primitive, the MLME shall set the state for the STA to State 4.

NOTE—Any MSDU fragments from the STA, and if management frame protection is in use, any MMPDU fragments from the STA, are discarded at this point (see 12.6.14), since fragments are required to be encrypted with the same key (see 10.5).

TBD: ditto for reassoc

Addendum: 12.6.19 Protection of robust Management frames

Discussion

12.6.19 says the following (numbering added for later referencing):

1. [MFPC=0, unicast] A STA with dot11RSNAProtectedManagementFramesActivated equal to false shall transmit and receive unprotected individually addressed robust Management frames to and from any associated STA and shall discard protected individually addressed robust Management frames received from any associated STA.

2. [MFPC=1 local only, MFPR=0, unicast] A STA with dot11RSNAProtectedManagementFramesActivated equal to true and dot11RSNAUnprotectedManagementFramesAllowed equal to true shall transmit and receive unprotected individually addressed robust Management frames to and from any associated STA that advertised MFPC = 0 and shall discard protected individually addressed robust Management frames received from any associated STA that advertised MFPC = 0.

3. [MFPC=1 both, MFPR=0, unicast] A STA with dot11RSNAProtectedManagementFramesActivated equal to true and dot11RSNAUnprotectedManagementFramesAllowed equal to true shall transmit and receive protected individually addressed robust Management frames to and from any associated STA that advertised MFPC = 1, shall discard unprotected individually addressed robust Action frames received from any STA that advertised MFPC = 1, and shall discard unprotected individually addressed Disassociation and Deauthentication frames received from a STA that advertised MFPC = 1 after the PTK and IGTK have been installed. The receiver shall process unprotected individually addressed Disassociation and Deauthentication frames before the PTK and IGTK are installed.

4. [MFPC=1, MFPR=1] A STA with dot11RSNAProtectedManagementFramesActivated equal to true and dot11RSNAUnprotectedManagementFramesAllowed equal to false shall transmit and receive protected individually addressed robust Action frames to and from any STA, shall not transmit unprotected individually addressed robust Action frames to any STA, and shall discard unprotected individually addressed robust Action frames received from a STA after the PTK and IGTK have been installed. The receiver shall process unprotected individually addressed Disassociation and Deauthentication frames before the PTK and IGTK are installed.

5. [MFPC=1 both, group] A STA with dot11RSNAProtectedManagementFramesActivated equal to true shall discard group addressed robust Management frames received from any associated STA that advertised MFPC = 1 if the frames are unprotected or if a matching IGTK is not available.

6. [MFPC=1, MFPR=1, group] A STA with dot11RSNAProtectedManagementFramesActivated equal to true and dot11RSNAUnprotectedManagementFramesAllowed equal to false shall discard received group addressed robust Management frames that are unprotected or for which a matching IGTK is not available.

7. [MFPC=0, group] A STA with dot11RSNAProtectedManagementFramesActivated equal to false shall transmit group addressed robust Management frames unprotected and shall ignore the protection on received group addressed robust Management frames.

8. The STA *[sic]* shall discard any robust Action frames received before the PTK and IGTK are installed.

This corresponds to the following requirements, where red and yellow indicates a need for discussion (see below table), orange indicates a need for clarification/rewording and blue is unspecified but “obvious” behaviour:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | MFPR=1 | MFPR=0 MFPC=1 both | MFPC=1 local only | MFPC=0 local |
| unprot unicast robust action before PTKSA established | discard (8) [scope of 8 not clear] | discard (3) -- also covered by (8) [scope of 8 not clear] | tx/rx (2) assuming can ignore (8) [scope of 8 not clear] | tx/rx (1) assuming can ignore (8) [scope of 8 not clear] |
| unprot unicast deauth/disassoc before PTKSA established | rx (4) so DoS risk? | rx (3) so DoS risk? | tx/rx (2) | tx/rx (1) |
| unprot unicast robust action after PTKSA established | discard (4) | discard (3) | tx/rx (2) | tx/rx (1) |
| unprot unicast deauth/disassoc after PTKSA established | unspecified but must be like MFPR=0 MFPC=1 but MFPR=0 MFPC=1 is broken | discard (3) so will prevent SAQ | tx/rx (2) | tx/rx (1) |
| prot unicast robust action before PTKSA established | tx/rx (3) but really should be impossible so discard (8)? [scope of 8 not clear] | tx/rx (3) but really should be impossible so discard (8)? [scope of 8 not clear] | discard (2) but should be impossible -- also covered by (8) [scope of 8 not clear] | discard (1) but should be impossible -- also covered by (8) [scope of 8 not clear] |
| prot unicast deauth/disassoc before PTKSA established | unspecified but should be impossible | tx/rx (3) but really should be impossible | discard (2) but should be impossible | discard (1) but should be impossible |
| prot unicast robust action after PTKSA established | tx/rx (4) | tx/rx (3) | discard (2) but should not happen (no MFP) | discard (1) but should not happen (no MFP) |
| prot unicast deauth/disassoc after PTKSA established | unspecified but must be tx/rx like MFPR=0 MFPC=1 | tx/rx (3) | discard (2) but should not happen (no MFP) | discard (1) but should not happen (no MFP) |
| unprot group robust action before IGTKSA established | discard (5)(6); also if no IGTK -- also covered by (8) [scope of 8 not clear] | discard (5); also if no IGTK -- also covered by (8) [scope of 8 not clear] | unspecified but must be tx/rx -- assuming can ignore (8) [scope of 8 not clear] | tx/rx (7) assuming can ignore (8) [scope of 8 not clear] |
| unprot group deauth/disassoc before IGTKSA established | discard (5)(6); also if no IGTK | discard (5); also if no IGTK | unspecified but must be tx/rx | tx/rx (7) |
| unprot group robust action after IGTKSA established | discard (5)(6); also if no IGTK | discard (5); also if no IGTK | unspecified but must be tx/rx | tx/rx (7) |
| unprot group deauth/disassoc after IGTKSA established | discard (5)(6) so will prevent SAQ; also if no IGTK | discard (5) so will prevent SAQ; also if no IGTK | unspecified but must be tx/rx | tx/rx (7) |
| prot group robust action before IGTKSA established | discard (8) [scope of 8 not clear] | discard (8) [scope of 8 not clear] | unspecified but must be rx -- assuming can ignore (8) [scope of 8 not clear] | rx (7) assuming can ignore (8) [scope of 8 not clear] |
| prot group deauth/disassoc before IGTKSA established | unspecified | unspecified | unspecified but must be rx | rx (7) |
| prot group robust action after IGTKSA established | unspecified but must be tx/rx by implication | unspecified but must be tx/rx by implication | unspecified but must be rx | rx (7) |
| prot group deauth/disassoc after IGTKSA established | unspecified but must be tx/rx by implication | unspecified but must be tx/rx by implication | unspecified but must be rx | rx (7) |

In particular:

* Accepting a unicast deauth/disassoc before the PTKSA is established is desirable in case the association needs to be aborted, but presents obvious DoS implications
* Accepting an unprotected unicast deauth/disassoc after the PTKSA is established is necessary to allow for a peer reset, but needs to be checked using the SA Query procedures
* Accepting a group-addressed deauth/disassoc, except when they protected and sent after the IGTKSA has been received, is not necessary as this is not needed to recover from a peer reset and if unprotected presents obvious DoS implications

On the basis of this, the following updated requirements are proposed, where

* yellow indicates a need for discussion (see above)
* grey indicates the condition is N/A for conformant STAs (and hence need not be normatively specified, though a NOTE might help to avoid doubt)
* MFP is used on a given link if both sides have MFPC=1, otherwise (except for group tx as shown in cyan) MFP is not used (and if one side has MFPR=1 and the other has MFPC=0 then there is no link at all)
* tx/rx is subject to Management frame Class (1/2/3) requirements:

|  |  |  |
| --- | --- | --- |
|  | MFP on link | No MFP on link |
| unprot unicast robust action before PTKSA established | no tx / discard on rx | tx/rx |
| unprot unicast deauth/disassoc before PTKSA established | tx/rx | tx/rx |
| unprot unicast robust action after PTKSA established | no tx / discard on rx | tx/rx |
| unprot unicast deauth/disassoc after PTKSA established | no tx / discard on rx and optionally do SAQ if reason code is invalid Class 2/3 | tx/rx |
| prot unicast robust action before PTKSA established | no tx / discard on rx (N/A because impossible) | no tx / discard on rx (N/A because impossible) |
| prot unicast deauth/disassoc before PTKSA established | no tx / discard on rx (N/A because impossible) | no tx / discard on rx (N/A because impossible) |
| prot unicast robust action after PTKSA established | tx/rx | no tx / discard on rx (N/A because no MFP) |
| prot unicast deauth/disassoc after PTKSA established | tx/rx | no tx / discard on rx (N/A because no MFP) |
| unprot group robust action before IGTKSA established | no tx / discard on rx (also if prot but no IGTK) | tx (AP) / rx (non-AP) |
| unprot group deauth/disassoc before IGTKSA established | no tx / discard on rx (also if prot but no IGTK) | tx (AP) / rx (non-AP) |
| unprot group robust action after IGTKSA established | no tx / discard on rx (also if prot but no IGTK) | tx (AP) / rx (non-AP) |
| unprot group deauth/disassoc after IGTKSA established | no tx / discard on rx (also if prot but no IGTK), no SAQ | tx (AP) / rx (non-AP) |
| prot group robust action before IGTKSA established | tx (AP with other MFP links) / discard on rx (from AP with other MFP links) | tx (AP with MFP links) / rx (non-AP, from AP with MFP links; ignore protection) |
| prot group deauth/disassoc before IGTKSA established | tx (AP with other MFP links) / discard on rx (from AP with other MFP links), no SAQ | tx (AP with MFP links) / rx (non-AP, from AP with MFP links; ignore protection) |
| prot group robust action after IGTKSA established | tx (AP) / rx (non-AP) | tx (AP with MFP links) / rx (non-AP, from AP with MFP links; ignore protection) |
| prot group deauth/disassoc after IGTKSA established | tx (AP) / rx (non-AP) | tx (AP with MFP links) / rx (non-AP, from AP with MFP links; ignore protection) |

Proposed changes:

Change 12.6.19 Protection of robust Management frames as follows:

This subclause defines rules that shall be followed by STAs that implement Management Frame protection and have dot11RSNAActivated equal to true.

NOTE—Management frame protection is negotiated for link if both STAs set the MFPC bit to 1 (see 12.6.3 (RSNA policy selection in an infrastructure BSS) and 12.6.5 (RSNA policy selection in an IBSS)).

***<delete the next 8 paras and the NOTE (which is moved to near the end, below)>***

Protection of group addressed robust Management frames shall be provided by a service in the MLME as described in 11.12 (Group addressed management frame protection procedures).

Management frame protection cannot be applied until the PTK and (except for a TDLS direct link) IGTK ha~~s~~ve been set.~~established with the STA. A STA shall not transmit robust Action frames until it has installed the PTK for the peer STA, or in the case of group addressed frames, has installed the IGTK. The STA shall discard any robust Action frames received before the PTK and IGTK are installed.~~

If management frame protection was not negotiated for the link, a STA shall ignore any protection on a received group addressed robust Management frame.

NOTE—The STA will not be sent any protected individually addressed robust Management frames. The STA might be sent a protected group addressed robust Management frame by a peer STA that has negotiated management frame protection for links with other STAs.

If management frame protection was negotiated for the link, a STA may discard an unprotected individually addressed Deauthentication or Disassociation frame received before the PTKSA has been established

NOTE—The STA might start a timer upon receiving the frame, and if the PTKSA establishment progresses or completes before the timer expires, the STA might discard the frame.

If management frame protection was negotiated for the link, a STA shall not transmit any of the following, and shall discard all of the following:

* An unprotected individually addressed Deauthentication or Disassociation frame after the PTKSA has been established

NOTE—The STA might invoke the SA Query procedures (see 11.13 (SA Query procedures)) if received with a reason code of INVALID\_CLASS2\_FRAME or INVALID\_CLASS3\_FRAME.

* An unprotected individually addressed robust Action frame

NOTE—The STA will not be sent any protected individually addressed robust Management frames before the PTKSA has been established.

* An unprotected group addressed robust Management frame

If management frame protection was negotiated for the link, a STA shall use the group management cipher suite for protected group addressed robust Management frames.

NOTE—BIP does not provide protection against forgery by associated (if in an infrastructure BSS, and optionally in a PBSS) and authenticated STAs. A STA that has left the BSS can successfully forge group addressed robust Management frames until the IGTK is updated.

If management frame protection was negotiated for the link, a STA shall discard all of the following:

* A protected group addressed robust Management frame when the STA has no IGTKSA for the link
* A protected group addressed robust Management frame that fails the checks described in 12.5.4.6 (BIP reception)

NOTE—The STA might be sent a protected group addressed robust Management frame when the STA has no IGTKSA for the link by a peer STA that has negotiated management frame protection for links with other STAs.

In 12.5.4.6 BIP reception delete “If management frame protection is negotiated, group addressed robust Management frames that are received without BIP protection shall be discarded.”

**References:**

802.11me/D0.0

802.11ax-2021

<https://papers.mathyvanhoef.com/usenix2021.pdf>