IEEE P802.11
Wireless LANs

|  |
| --- |
| Reassociating STA recognition |
| Date: 2023-06-02 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Jouni Malinen | Qualcomm Technologies, Inc. |  |  | jouni@qca.qualcomm.com |
|  |  |  |  |  |

Abstract

This document discusses issues related to secure recognition of a reassociating STA by an AP and proposed new mechanism to allow this to be done. This is related to the association comeback in management frame protection and how the use of SA Query can result in undesired latency in being able to negotiate new parameters for an association in the reassociate-to-same-BSS case. Furthermore, the proposed design can provide some help in addressing recently reported security vulnerabilities in MAC address “ownership” and potential insider attacks.

This is the topic of the REVme/D3.0 LB273 CID 4069.

This was marked ready-for-motion during the 2023-06-02 telco (using the REVISED resolution and the language in 537r6).

**REVme/D3.0 LB273 CID 4069**

Clause: 11.3.5.3

Comment:

Management frame protection includes a mechanism for an AP to delay an association if a (Re)Association Request frame is received for a STA that has a current security association with the AP. This is done to provide protection against disconnection attacks. However, this can also result in undesired latency for cases where a STA is trying to reassociate back to the same BSS to renegotiate some parameters or associate to a BSS when a prior association was not explicitly removed with deauthentication or disassociation.

While this potential extra delay has been known to exist since the IEEE 802.11w design, there is now increased interest in trying to optimize this to allow the reassociation back to the same BSS, or AP MLD in this case, to renegotiate link parameters. This on its own could justify changes to the management frame protection design in IEEE 802.11be, but the issue itself is more generic and of more use which justifies addressing this in IEEE 802.11-REVme.

In addition to the extra delay due to association comeback mechanism, more justification for recognizing a specific returning STA securely has come up from a recently published draft paper Framing Frames: Bypassing Wi-Fi Encryption by Manipulating Transmit Queues discussing various security vulnerabilities related to the overriding a security context, "ownership" of a MAC address, and insider attacks that might be enabled when any STA with valid credentials is allowed to pick any arbitrary MAC address for an association.

This comment proposes a new mechanism that allows the AP and STA to remember PTK-KCK for a longer period than the PTKSA itself, if needed for some sequences where the PTKSA would be otherwise required to be deleted. The STA can then use this still shared PTK-KCK to demonstrate the possession of that key and by doing that, show that it is the same STA and not another STA that happens to have credentials to access the BSS and that is attempting to use a MAC address that the original STA used in a recent association. This provides sufficient information to the AP to allow it to skip the SA Query procedure and association comeback. This might also help the AP in limiting reuse of a specific MAC address between STAs in attempt to mitigate potential attacks that try to recover pending frames that are being received by the AP through the DS for a STA with a specific MAC address.

Proposed Change:

Incorporated changes under the "Proposed changes" section of https://mentor.ieee.org/802.11/dcn/23/11-23-0537-00-000m-reassociating-sta-recognition.docx.

Proposed Resolution:

REVISED – Incorporate changes under the “Proposed changes for CID 4069” section of <this doc>. This is an updated version of the changes proposed in the comment with additional changes to address items that came up while discussing the comment. This covers only the optimization for (re)association-back-to-same-BSS case. Recognization of a returning STA for protecting a MAC address is being discussed in P802.11bh.

## Discussion

Management frame protection includes a mechanism for an AP to delay an association if a (Re)Association Request frame is received for a STA that has a current security association with the AP. This is done to provide protection against disconnection attacks. However, this can also result in undesired latency for cases where a STA is trying to reassociate back to the same BSS to renegotiate some parameters or associate to a BSS when a prior association was not explicitly removed with deauthentication or disassociation.

While this potential extra delay has been known to exist since the IEEE 802.11w design, there is now increased interest in trying to optimize this to allow the reassociation back to the same BSS, or AP MLD in this case, to renegotiate link parameters. This on its own could justify changes to the management frame protection design in IEEE 802.11be, but the issue itself is more generic and of more use which justifies addressing this in IEEE 802.11-REVme. It should be noted that 802.11be has added a separate mechanism for removing links.

In addition to the extra delay due to association comeback mechanism, more justification for recognizing a specific returning STA securely has come up from a recently published draft paper [Framing Frames: Bypassing Wi-Fi Encryption by Manipulating Transmit Queues](https://papers.mathyvanhoef.com/usenix2023-wifi.pdf) Section 5 (Overriding the Victim’s Security Context) discussing various security vulnerabilities related to the overriding a security context, “ownership” of a MAC address, and insider attacks that might be enabled when any STA with valid credentials is allowed to pick any arbitrary MAC address for an association.

This contribution proposes a new mechanism that allows the AP and STA to remember PTK-KCK for a longer period than the PTKSA itself, if needed for some sequences where the PTKSA would be otherwise required to be deleted. The STA can then use this still shared PTK-KCK to demonstrate the possession of that key and by doing that, show that it is the same STA and not another STA that happens to have credentials to access the BSS and that is attempting to use a MAC address that the original STA used in a recent association. This provides sufficient information to the AP to allow it to skip the SA Query procedure and association comeback. This might also help the AP in limiting reuse of a specific MAC address between STAs in attempt to mitigate potential attacks that try to recover pending frames that are being received by the AP through the DS for a STA with a specific MAC address. For example, the APs in an ESS could reject association that uses a MAC address that is currently used in the ESS or that has been used in the recent past, if the (Re)Association Request frame does not connect the request securely to the STA that uses/used that MAC address. Such rejection for a relatively short duration would mitigate attacks that try to capture frames that are still buffered somewhere in the network for the same MAC address.

It should also be noted that TGbh may have finally managed to move ahead in the process by approving IEEE P802.11bh/D1.0 to be created and sent for WG LB. This happened after the REVme/D3.0 comment was filed. P802.11bh includes a similar capability to allow a returning STA to be recognized. While the design used there does not seem suitable for skipping the SA Query procedure in the reassociation-back-to-the-same-BSS case, it might be suitable for the MAC address ownership case. As such, the document does not propose mechanisms that are better discussed in TGbh to avoid definition of duplicated functionality.

## Proposed changes for CID 4069

*Note: The referenced subclauses, pages, and lines are based on IEEE P802.11-REVme/D3.0.*

**9.3.3.5 Association Request frame format**

*Insert the following row to the end of Table 9-62 (Association Request frame body) in 9.3.3.5 (P702 L13) immediately before the “Last/Vendor Specific” row. The header row is included here for context and only the following row is to be inserted.*

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| <next available number> | Known STA Identification | The Known STA Identification element is optionally present if dot11RSNAActivated is true and the recipient of the frame has indicated support for this element in its Extended Capabilities field; otherwise not present. |

**9.3.3.7 Reassociation Request frame format**

*Insert the following row to the end of Table 9-64 (Reassociation Request frame body) in 9.3.3.7 (P710 L8) immediately before the “Last/Vendor Specific” row. The header row is included here for context and only the following row is to be inserted.*

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| <next available number> | Known STA Identification | The Known STA Identification element is optionally present if dot11RSNAActivated is true and the recipient of the frame has indicated support for this element in its Extended Capabilities field; otherwise not present. |

**9.4.2 Elements
9.4.2.1 General**

*Insert the following row to Table 9-128 (Element IDs) in 9.4.2.1 (P854 L62) immediately before the last “Reserved” row and update the Reserved Element ID Extension range appropriately. The header row is included here for context and only the following row is to be inserted.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Element ID** | **Element ID Extension**  | **Extensible** | **Fragmentable** |
| Known STA Identification (see 9.4.2.x) | 255 | <ANA> | Yes | No |

**9.4.2.25 Extended Capabilities element**

*Insert the following row to Table 9-190 (Extended Capabilities field) in 9.4.2.25 (P987 L46) immediately before the “Reserved” row and update the Reserved Information range appropriately. The header row is included here for context and only the following row is to be inserted.*

|  |  |  |
| --- | --- | --- |
| **Bit** | **Information** | **Notes** |
| <ANA> | Known STA Identification Enabled | Set to 1 to indicate that the AP has enabled use of Known STA Identification element. Set to 0 otherwise. This field is reserved for a non-AP STA. |

*Insert the following subclause at the end of 9.4.2, i.e., immediately before 9.4.3 (P1481 L43):*

**9.4.2.x Known STA Identification element**

The Known STA Identification element is used to demonstrate possession of a recently used PTK-KCK to allow a (re)associating STA to be identified in an RSN.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Element ID | Length | Element ID Extension | Timestamp | MIC Length | MIC |
| Octets: | 1 | 1 | 1 | 8 | 1 | variable |

 **Figure 9-y—Known STA Identification element format**

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

The Timestamp field contains the value of the most recently received Timestamp field from the AP to which the Known STA Identification element is being sent.

The MIC Length field contains the length of the MIC field in octets.

The MIC field contains a MIC calculated as specified in 11.13 (SA Query procedures).

**11.3.5.3 AP or PCP association receipt procedures**

*Modify 11.3.5.3 list item (e) as shown (P2438 L15):*

e) Otherwise, if the state for the STA is State 4, the STA has a valid security association, the STA has negotiated management frame protection, the STA has not performed a successful SAE authentication after the current association was established, the STA did not include a valid Known STA Identification element (see 11.13), and there has been no earlier, timed out SA Query procedure with the STA (which would have allowed a new association process to be started, without an additional SA Query procedure):

1) The SME shall refuse the association request by issuing an MLME-ASSOCIATE.response primitive with ResultCode REFUSED\_TEMPORARILY and TimeoutInterval containing a TIE with the Timeout Interval Type field set to 3 (association comeback time).

If the SME is in an ongoing SA Query with the STA, the Timeout Interval Value field shall be set to the remaining SA Query period, otherwise it shall be set to dot11AssociationSAQueryMaximumTimeout.

2) The state for the STA shall be left unchanged.

3) Following this, if the SME is not in an ongoing SA Query with the STA, the SME shall issue one MLME-SA-QUERY.request primitive addressed to the STA every dot11AssociationSAQueryRetryTimeout Tus until an MLME-SA-QUERY.confirm primitive for the STA is received or dot11AssociationSAQueryMaximumTimeout Tus from the beginning of the SA Query procedure have passed. The SME shall increment the TransactionIdentifier by 1 for each MLME-SA-QUERY.request primitive, rolling it over to 0 after the maximum allowed value is reached.

4) If no MLME-SA-QUERY.confirm primitive for the STA is received within the dot11AssociationSAQueryMaximumTimeout period, the SME shall allow a subsequent association process with the STA to be started without starting an additional SA Query procedure, except that the SME may deny a subsequent association process with the STA if an MSDU was received from the STA within this period.

NOTE 1—Reception of an MSDU implies reception of a valid protected frame, which obviates the need for the SA Query procedure.

**11.3.5.5 AP or PCP reassociation receipt procedures**

*Modify 11.3.5.5 list item € as shown (P2442 L9):*

e) Otherwise, if the state for the STA is State 4, the STA has a valid security association, the STA has negotiated management frame protection, the reassociation is not a part of a fast BSS transition, the STA has not performed a successful SAE authentication after the current association was established, the STA did not include a valid Known STA Identification element (see 11.13), and there has been no earlier, timed out SA Query procedure with the STA (which would have allowed a new reassociation process to be started, without an additional SA Query procedure):

1) The SME shall refuse the reassociation request by issuing an MLME-REASSOCIATE.response primitive with ResultCode REFUSED\_TEMPORARILY and TimeoutInterval containing a TIE with the Timeout Interval Type field set to 3 (association comeback time). If the SME is in an ongoing SA Query with the STA, the Timeout Interval Value field shall be set to the remaining SA Query period, otherwise it shall be set to dot11AssociationSAQueryMaximumTimeout.

2) The state for the STA shall be left unchanged.

3) Following this, if the SME is not in an ongoing SA Query with the STA, the SME shall issue one MLME-SA-QUERY.request primitive addressed to the STA every dot11AssociationSAQueryRetryTimeout TUs until an MLME-SA-QUERY.confirm primitive

**11.13 SA Query procedures**

*Insert the following paragraphs at the end of 11.13 (P2539 L13):*

A non-AP STA can identify itself securely when (re)associating with the same AP to avoid the need to use the SA Query procedure and association comeback delay when that non-AP STA and the AP share a common PTK-KCK. The non-AP STA includes a Known STA Identification element in its (Re)Association Request frame to do this, and the AP allows the (re)association to proceed without an SA Query procedure if it receives a valid Known STA Identification element.

The MIC for the Known STA Identification element shall be calculated using the most recently used PTK-KCK and the integrity algorithm (see Table 12-11) for the AKM that was negotiated to be used with that key: MIC(PTK-KCK, Timestamp). The Timestamp value is the most recently received Timestamp field the non-AP STA has received from the AP, e.g., from a Beacon frame. The AP may use this to discard Known STA Identification elements that are based on old information to limit the possibility of replay attacks. The AP should allow timestamp values that it has sent during the last 30 seconds, or the BSS max idle period, if it is advertised and larger, to be recognized as valid, but the AP may accept a larger window to accommodate cases where the non-AP STA might use cached information for an association. The AP shall discard the Known STA Identification element if the same timestamp value is used more than once. The Known STA Identification element is valid if it was not discarded, and the received MIC matches the locally calculated value. Otherwise, the Known STA Identification element is invalid and shall not be used to skip an SA Query procedure.

**12.6.1.1.6 PTKSA**

*Insert the following paragraph at the end of 12.6.1.1.6 (P2857 L38):*

A non-AP STA may retain the PTK-KCK and the negotiated AKM suite selector when deleting a PTKSA to allow Known STA Identification element to be used even when no PTKSA exists, see 11.13 (SA Query procedures).