IEEE P802.11
Wireless LANs

|  |
| --- |
| Some REVmc D4.0 SB RSN comments |
| Date: 2015-12-02 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Jouni Malinen | Qualcomm |  |  | jouni@qca.qualcomm.com |
|  |  |  |  |  |

Abstract

This document discusses couple of REVmc D4.0 SB comments and issues that came up from earlier discussions in this area. This includes proposed resolutions to comments CID 6332 and CID 6403 and proposed changes to address issues that do not have an explicit comment, but that are related to a discussion or research started by one of the comments.

The text with yellow highlighting identifies the proposed subjects of motions to approve comment resolutions and draft changes.

# CID 6332 – What does “default” suite mean in RSNE?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Page** | **Line** | **Comment** | **Proposed Change** |
| 6332 | 8.4.2.24.2 | 819 | 44 | It says "default pairwise cipher suite and default group cipher suite for Data frames in an RSNA" but what does default mean here? | Is it referring to the case where the RSNE is truncated before these fields? If so, say so; if not, say what it means |

Relevant text in the current draft (both D4.0 and D4.3)

**P802.11-REVmv/D4.0 - P819 L44**







Note that CID 6333 and CID 6334 resolutions modified the text here for 00-0F-AC:4 and 00-0F-AC:8.

CID 6333: REVISED (MAC: 2015-09-14 09:19:39Z): For Suite type 4, in the Meaning column, insert "for a non-DMG STA" at the end.

CID 6334: REVISED (MAC: 2015-09-14 09:14:45Z): For Suite type 8, in the Meaning column, replace "default" with "default pairwise cipher suite and default group cipher suite for Data frames in an RSNA"

The current REVmc/D4.3 (note: This is now Table 8-129 P836 L40 and L51) Meaning column text for these is as follows:

00-0F-AC:4: CCMP-128 – default pairwise cipher suite and default group cipher suite for Data frames in an RSNA for a non-DMG STA

00-0F-AC:8: GCMP-128 – default pairwise cipher suite and default group cipher suite for Data frames in an RSNA for a DMG STA

Following the Table 8-129 in REVmc/D4.3 P837 L13 there is the following relevant, but not anymore accurate, statement:

The cipher suite selector 00-0F-AC:4 (CCMP-128) is the default cipher suite value.

Also note that there is similar “default” case for AKM suite selectors:



This table (Table 8-131 in REVmc/D4.3) is followed by following paragraph in REVmc/D4.3 P840 L27:

The AKM suite selector value 00-0F-AC:1 (i.e., Authentication negotiated over IEEE Std 802.1X with RSNA key management as defined in 11.7 (Keys and key distribution) or using PMKSA caching as defined in 11.6.10.3 (Cached PMKSAs and RSNA key management)) is the assumed default when the AKM suite selector field is not supplied.

**8.4.2.24.4 RSN capabilities**

The RSN Capabilities field indicates requested or advertised capabilities. If the RSN Capabilities field is not present, the default value of 0 is used for all of the capability subfields.

Discussion:

The comment is asking what the word “default” mean in Table 8-128 and the proposed change from the commenter asks whether this is referring to the case where RSNE is truncated before these fields. That is indeed the case. The default suites indicated in Table 8-128 for cipher suites and Table 8-130 for AKM suites are the values that are to be used if the relevant optional field (Group Data Cipher Suite, Pairwise Cipher Suite List, AKM Suite List, Group Management Cipher Suite) is not included in the RSNE. Since all the fields after the Version field are optional, this default value is needed to be able to determine how the receiver is to interpret the element. The paragraph following the AKM suite selectors table has clearer language indicating that the default row is used when the “field is not supplied”. That same wording could be used in the paragraph following the Cipher suite selectors table. In addition, the DMG change (802.11ad) on the default cipher suite (GCMP-128 instead of CCMP-128) needs to be covered in that paragraph. This paragraph should also note that BIP-CMAC-128 is the default group management cipher suite.

It looks like instructions on how to interpret all other optional fields when they are not present in the RSNE are already described in sufficient detail apart from the fields where the cipher suite selectors are used. The text in the Meaning column of the Table 8-129 (REVmc/D4.3) seems to be sufficient (and consistent with the style used for AKM suite selectors) and the comment can be addressed by updating the paragraph following Table 8-129 to describe the meaning of “default” and the relevant fields not being included in an RSNE.

Proposed draft changes for CID 6332 (baseline: REVmc/D4.3):

*Change REVmc/D4.3 P837 L14 (8.4.2.24.2 Cipher suites) as shown:*

In non-DMG RSNA, the cipher suite selector 00-0F-AC:4 (CCMP-128) is the default pairwise cipher suite and the default group cipher suite for Data frames value when the Group Data Cipher Suite or Pairwise Cipher Suite List fields are not included. In DMG RSNA, the cipher suite selector 00-0F-AC:8 (GCMP-128) is the default pairwise cipher suite and the default group cipher suite for Data frames value when the Group Data Cipher Suite or Pairwise Cipher Suite List fields are not included. In an RSNA with management frame protection enabled, 00-0F-AC:6 (BIP-CMAC-128) is the default group management cipher suite value when the Group Management Cipher Suite field is not included.

Proposed Resolution for CID 6332:

REVISED: In REVmc/D4.3 P837 L14 (8.4.2.24.2 Cipher suites), replace

“The cipher suite selector 00-0F-AC:4 (CCMP-128) is the default cipher suite value.”

with

“In non-DMG RSNA, the cipher suite selector 00-0F-AC:4 (CCMP-128) is the default pairwise cipher suite and the default group cipher suite for Data frames value when the Group Data Cipher Suite or Pairwise Cipher Suite List fields are not included. In DMG RSNA, the cipher suite selector 00-0F-AC:8 (GCMP-128) is the default pairwise cipher suite and the default group cipher suite for Data frames value when the Group Data Cipher Suite or Pairwise Cipher Suite List fields are not included. In an RSNA with management frame protection enabled, 00-0F-AC:6 (BIP-CMAC-128) is the default group management cipher suite value when the Group Management Cipher Suite field is not included.”.

# CID 6403 – Truncate-n() and “irretrievably deleted”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Page** | **Line** | **Comment** | **Proposed Change** |
| 6403 |  | 1941 | 63 | Do the msbs need to be irretrievably deleted at 1941.63 (EAPOL-Key MIC), 1896.4 (BIP-CMAC) and 959.5 (Emergency Alert Identifier Hash) in D3.0? | Add a Truncate-128() for the first one, and generalise Truncate-128() to Truncate-n(), so the latter two can use Truncate-64() |

Note: Page/line references to D4.0 are 1963.44 (EAPOL-Key MIC), 1918.4 (BIP-CMAC), and 968.4 (Emergency Alert Identifier Hash).

Relevant text in the current draft (both D4.0 and D4.3)

**P802.11-REVmv/D4.0 – P3 L54 (no changes in D4.3)**

**1.5 Terminology for mathematical, logical and bit operations**

Truncate-128 (*S*) is bits 0 to 127 of the bit string *S* starting from the left, using the IEEE Std 802.11 bit conventions from 8.2.2 (Conventions). Other bits are irretrievably deleted.

**P802.11-REVmv/D4.0 – P1963 L44**

**11.6.2 EAPOL-Key frames**



**P802.11-REVmv/D4.0 – P1918 L4 (note: see below for D4.3 changes)**

**11.4.4.1 BIP overview**

BIP-CMAC-128 provides data integrity and replay protection, using AES-128 in CMAC Mode with a 128-bit integrity key and a CMAC TLen value of 128 (16 octets). BIP-CMAC-256 provides data integrity and replay protection, using AES-256 in CMAC Mode with a 256-bit integrity key and a CMAC TLen value of 128 (16 octets). NIST SP 800-38B defines the CMAC algorithm, and NIST SP 800-38D defines the GMAC algorithm. BIP processing uses AES with a 128-bit or 256-bit integrity key and a CMAC TLen value of 128 (16 octets). The CMAC output for both BIP-CMAC-128 and BIP-CMAC-256 is truncated to 64 bits:

MIC = L(CMAC Output, 0, 64).

**P802.11-REVmv/D4.3 – P1984 L12**

**11.5.4.1 BIP overview**

BIP processing uses AES with a 128-bit or 256-bit integrity key and a CMAC TLen value of 128 (16 octets). The CMAC output for BIP-CMAC-256 is not truncated and shall be 128 bits (16 octets).(#6564) The CMAC output for BIP-CMAC-128(#6564) is truncated to 64 bits:

MIC = L(CMAC Output, 0, 64)(#239)(#3431).

BIP-GCMP-128 uses AES with a 128-bit integrity key, and BIP-GCMP-256 uses AES with a 256-bit integrity key. The authentication tag for both BIP-GCMP-128 and BIP-GCMP-256 is not truncated and shall be 128 bits (16 octets).

**P802.11-REVmv/D4.0 – P968 L4**

**8.4.2.96 Emergency Alert Identifier element**

The Alert Identifier Hash (AIH) is an 8-octet field. It is a unique value used to indicate an instance of an EAS message. The value of this field is the hash produced by the HMAC-SHA-1-64 hash algorithm operating on the EAS message.

AIH =HMAC-SHA-1-64(“ES\_ALERT”, Emergency\_Alert\_Message)

Where

 “ES\_ALERT” is treated as a sequence of ASCII-encoded octets without a terminating null Emergency\_Alert\_Message is the EAS message itself

Discussion:

The comment asks whether the “other bits are irretrievably deleted” condition used in Truncate-128(S) should be applicable in these three occurrences of HMAC/CMAC hash truncation. “Irretrievably deleted” here describes an operation that is internal to an implementation and its effects are not observable over air or externally to the device. The reason for deleting some internal data irretrievably is to reduce risk of exposing this information due to an implementation bug or some type of attack trying to extract such internal information and then use it to do something (in the cases here, likely to try to figure out one of the private keys).

In the current draft, Truncate-128(S) is used in the following cases:

* PMKID derivation from HMAC-SHA-256/384(PMK/KCK, …)
* SMKID derivation from HMAC-SHA-256(SMK, …)
* PMKR0Name derivation from SHA-256(… || PMK-R0Name-Salt)
* PMKR1Name derivation from SHA-256(… || PMKR0Name || …)
* PTKName derivation from SHA-256(PMKR1Name || …)
* PMKID derivation from SHA-256(Q1 || Q2 || …)

All these cases derive key names that could be sent out in clear. Most of these use private information in the input parameters (with the exception PMKR1Name and PTKName which are derived based on publicly known PMKR0Name and identifiers).

In addition to use of Truncate-128(S), the current draft “irretrievably deletes” *rand* and *mask* values in 11.4.5.2 (SAE/PWE), SAE protocol instance in 11.4.8.2.2 and 11.4.8.6.1, and *keyseed* in PMK derivation in 11.10.2 (AP PeerKey protocol).

EAPOL-Key MIC field is 16 octets in cases where HMAC-SHA-1-128 is used. The truncation of the 160 bit HMAC-SHA-1 value to 128 bits is used to fit this into the 16 octet field. The EAPOL-Key Key MIC field is sent in clear when going through the initial 4-way handshake. KCK is used has the HMAC key in this operation. As such, this is quite similar to the existing PMKID derivation case. It would seem reasonable to use the same “irretrievable deleted” behaviour here. However, the cited location is not a formula for deriving Key MIC, so use of Truncate-128(S) here would not look ideal. The “shall be discarded” language can be replaced with “shall be irretrievably deleted” to get the same meaning.

BIP MIC is derived from BIP-CMAC-128 output with MIC = L(CMAC Output, 0, 64). It should be noted that in D4.0, this was also the case for BIP-CMAC-256 in the cited location, but that truncation was fixed to apply only for the BIP-CMAC-128 case. BIP-CMAC-128 uses IGTK as the key and it looks reasonable to irretrievably delete the bits of CMAC Output that are not used for MIC. In other words, L(CMAC Output, 0, 64) could be replaced with Truncate-64(CMAC Output) if we were to define Truncate-64(S) similarly to the existing Truncate-128(S).

Alert Identifier Hash (AIH) is used to identify an instance of an active EAS message that is currently available from the network. Such a message is public information and the mechanism described for fetching it in 10.25.7 uses unencrypted GAS messages that are available to any station. As such, there is no private information that would be protected by discarding truncated bits from HMAC-SHA-1 output. Use of Truncate-64(S) here would be excessive and no changes are needed to the draft.

Proposed draft changes for CID 6403 (baseline: REVmc/D4.3):

*Change REVmc/D4.3 P4 L16 (1.5 Terminology for mathematical, logical and bit operations) as shown:*

Truncate-N (*S*) is bits 0 to N–1 of the bit string *S* starting from the left, using the IEEE Std 802.11 bit conventions from 8.2.2 (Conventions). Other bits are irretrievably deleted.

*Change REVmc/D4.3 P1984 L12 (11.5.4.1 BIP overview) as shown:*

BIP processing uses AES with a 128-bit or 256-bit integrity key and a CMAC TLen value of 128 (16 octets). The CMAC output for BIP-CMAC-256 is not truncated and shall be 128 bits (16 octets). The CMAC output for BIP-CMAC-128 is truncated to 64 bits:

MIC = Truncate-64(CMAC Output).

BIP-GCMP-128 uses AES with a 128-bit integrity key, and BIP-GCMP-256 uses AES with a 256-bit integrity key. The authentication tag for both BIP-GCMP-128 and BIP-GCMP-256 is not truncated and shall be 128 bits (16 octets).

*Change REVmc/D4.3 P2031 L40 (11.7.2 EAPOL-Key frames) as shown:*

HMAC-SHA-1-128 is the EAPOL-Key MIC. HMAC is defined in IETF RFC 2104; and SHA-1, by FIPS PUB 180-3-2008. The output of the HMAC-SHA-1 shall be truncated to its 128 MSBs (octets 0–15 of the digest output by HMAC-SHA-1), i.e., the last four octets generated shall be irretrievably deleted.

Proposed Resolution for CID 6403:

REVISED: Irretrievably delete discarded EAPOL-Key MIC and BIP MIC bits. No changes for AIH since the hashed data is public information.

In REVmc/D4.3 P4 L16 (1.5 Terminology for mathematical, logical and bit operations), replace

“Truncate-128 (S) is bits 0 to 127 of the bit string S starting from the left, using the IEEE Std 802.11 bit conventions from 8.2.2”

with

“Truncate-N (S) is bits 0 to N-1 of the bit string S starting from the left, using the IEEE Std 802.11 bit conventions from 8.2.2”.

In REVmc/D4.3 P1984 L12 (11.5.4.1 BIP overview), replace

“MIC = L(CMAC Output, 0, 64)”

with

“MIC = Truncate-64(CMAC Output)”

In REVmc/D4.3 P2031 L40 (11.7.2 EAPOL-Key frames), replace

“the last four octets generated shall be discarded”

with

“the last four octets generated shall be irretrievably deleted”.

# CCMP/GCMP transmitter rules for PN during fragmentation

Discussion:

During the discussion on CID 6024 and CID 6239 regarding GCMP receiver rules on replay protection of fragmentated frames (“The receiver shall discard MSDUs and MMPDUs whose constituent MPDU PN values are not sequential”; see 15/1132r2 for more detailed history and description of this functionality) it was identified that the current standard does not have clear rules for the transmitter side to match this receiver rule. The resolutions approved for CID 6024 and CID 6239 (the changes in 15/1132r2) addressed the receiver side. The transmitter side was strictly speaking out-of-scope for the comments and addressing it was left as a separate item. While there is no clear comment for fixing this, the issue is valid and should be addressed as part of REVmc process.

This submission proposed text changes to introduce the needed transmitter requirements for this replay protection mechanism to work correctly. In practice, all implementations of CCMP and GCMP already need to comply with this rule to be able to interoperate with a compliant CCMP/GCMP receiver. As such, these changes should be considered as cleanup for making the description complete rather than technical changes to existing behavior.

In addition to adding the explicit transmitter rule for fragmented frames, the changes below synchronizes CCMP and GCMP processing for PN on the transmitter side. 16/1132r2 handled many similar cases mainly from the recipient side, but the PN processing subclause and its transmitter role rules were not covered. However, these have the same issue with some changes after 802.11ad merge having been made only in the CCMP section even though the changes are applicable to both CCMP and GCMP. There are also unnecessary differences in the description between CCMP and GCMP which are supposed to use PN in the identical way.

Proposed changes to CCMP and GCMP PN processing

*Change REVmc/D4.3 P1979 L6-14 (CCMP) as shown:*

**11.5.3.3.2 PN processing**

The PN is incremented by a positive number for each MPDU. The PN shall be incremented in steps of 1 for constituent MPDUs of fragmented MSDUs and MMPDUs. The PN shall never repeat for a series of encrypted MPDUs using the same temporal key.

NOTE—When a group addressed MSDU is retransmitted using GCR, it is concealed from non-GCR capable STAs using the procedures described in 10.24.16.3.5 (Concealment of GCR transmissions). The MPDU containing this concealed A-MSDU will have a different PN than the MPDU that contained the original transmission of the group addressed MSDU.

*Change REVmc/D4.3 P1979 L6-14 (GCMP) as shown:*

**11.5.5.3.2 PN processing**

The PN is incremented by a positive number for each MPDU. The PN shall be incremented in steps of 1 for constituent MPDUs of fragmented MSDUs and MMPDUs. The PN shall never repeat for a series of encrypted MPDUs using the same temporal key.

If the PN is larger than the value of dot11PNExhaustionThreshold, an MLME-PN- EXHAUSTION.indication primitive shall be generated.

NOTE—When a group addressed MSDU is retransmitted using GCR, it is concealed from non-GCR capable STAs using the procedures described in 10.24.16.3.5 (Concealment of GCR transmissions). The MPDU containing this concealed A-MSDU will have a different PN than the MPDU that contained the original transmission of the group addressed MSDU.

# Missed PMK length changes

The 802.11ac changes to introduce a variable length PMK missed some changes. PMK is 256 bits in most cases, but with AKM 00-0F-AC:12 and 00-0F-AC:13 a longer 384-bit PMK is used. 11.7.1.3 notes the 00-0F-AC:12 (non-FT) exception and 00-0F-AC:13. 11.7.1.7.3 mentions the 00-0F-AC:13 (FT) exception. The Supplicant and Authenticator state machines were not updated to use PMK\_bits as the length of the PMK instead of the fixed 256.

Proposed changes to PMK length

*Change REVmc/D4.3 P2020 L57 as shown:*

**11.7.1.3 Pairwise key hierarchy**

When not using a PSK, the PMK is derived from the MSK. The PMK shall be computed as the first PMK\_bits bits (bits 0 to PMK\_bits–1) of the MSK: PMK  L(MSK, 0, PMK\_bits).

*Change REVmc/D4.3 P2070 L33 as shown:*

**11.7.9.5 Supplicant state machine procedures**

…

**StaProcessEAPOL-Key**

…

*PMK 🡨* L(MSK, 0, PMK\_bits)

*Change REVmc/D4.3 P2078 L20 Figure 11-50 INITPMK state contents as shown:*

PMK = L( MSK, 0, PMK\_bits )