IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11bi Comment resolution for CID 2071,2072 | | | | |
| Date: 2025-09-15 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | Email |
| Domenico Ficara | Cisco |  |  | dficara@cisco.com |
| Ugo Campiglio | Cisco |  |  | ucampigl@cisco.com |
| Javier Contreras | Cisco |  |  | jacontre@cisco.com |
| Jerome Henry | Cisco |  |  | jhenry@cisco.com |
| Federico Lovison | Cisco |  |  | flovison@cisco.com |

Abstract

This submission resolves the following CIDs:

2071, 2072

Revisions:

* Rev 0: Initial version of the document.
* Rev 1: Address comments from the presentation of Sep 18th 2025

***Editing instructions formatted like this are intended to be copied into the TGbi D2.0 Draft. (i.e. they are instructions to the 802.11 editor on how to merge the text with the baseline documents). TGbi Editor: Editing instructions preceded by “TGbi Editor” are instructions to the TGbi editor to modify existing material in the TGbi draft. As a result of adopting the changes, the TGbi editor will execute the instructions rather than copy them to the TGbi Draft.***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **Clause** | **P.L** | **Comment** | **Proposed Change** | **Resolution** |
| 2071 | 10.71.3 | 102.06 | Since Seed, n and p are just numbers that we add/multiply, there is quite some chance that (n+p) x EpochInterval + Seed can generate common subsequences if jumping from a EPP Group to another EPP Group. | commenter will provide a contribution to solve this. | Revised –  As described in 25/1549r0, proposal 1, Seed is appended to the rest of the computation in the context. This is also reflected in the other equations using Seed in 10.71.2.4 and 10.71.4.  Instructions to the editor:  Please make the changes as shown under CID 2071 in this document |
| 2072 | 10.71.3 | 102.06 | In 802.11bn PTK sharing is part of the current draft. I'm not sure if we need to address this now, but since it cause no harm, I suggest that this computation should be linked to the AP MLD MAC. | commenter will provide a contribution to solve this. | Revised –  As described in 25/1549r0, proposal 1, AP\_MLD\_MAC is appended to the rest of the computation in the context. This is also reflected in the other equations using Seed in 10.71.2.4 and 10.71.4  Instructions to the editor:  Please make the changes as shown under CID 2072 in this document |

***Proposal:***

**TGbi Editor: *Instruction: Please modify the equation for*** ΔIT(*n*) ***and its description in 10.71.2.4 as shown below***

* EPP Epoch Start Time Computation

[…]

ΔIT(*n*) = int (KDF-*Hash*-*Length*(PGTK, “ERCM”, Seed ||AP\_MLD\_MAC|| (*n* × EpochInterval))) mod TimeRangeTU(#2071, #2072)

and where

*n* is the current number of the EPP epoch in the EPP epoch sequence.

PlannedEpochTSFStartTime(*n*) is the TSF timer value of the link corresponding to the nominal start

time of the EPP epoch number n in the EPP epoch sequence.

This planned start time occurs at a regular time interval equal to the

epoch interval.

EpochNumberOffset is the value indicated in the Epoch Number Offset field of the

EPP Epoch Settings field.

EpochInterval is the value in TU corresponding to the Epoch Interval

field of the EPP Epoch Settings field .

KDF-*Hash*-*Length* is the key derivation function as defined in

12.7.1.6.2 (Key derivation function (KDF)) using the

hash algorithm identified by the AKM suite selector

(see -- Editor Note: (ANA assignment is done)).

*Length* is the number of bits to derive. 16 bits are derived for ΔIT.

FirstPlannedEpochTSFStartTime is the value of the first epoch TSF start time,

initialized, upon reception of an EPP element by the STA with

the First Epoch TSF Start Time value of the EPP element of

the received EPP Epoch Settings field.

TimeRangeTU is the value in TU corresponding to the value of the Epoch Start Time

Variation Range field multiplied by the number of TU in the Epoch

Interval Unit field of the EPP Epoch Settings field.

PGTK is the cryptographic key assigned by an EPP AP MLD that is

used to manage the group EPP epoch, distributed to the EPP

non-AP MLDs associated with the EPP AP MLD.

Seed is the Group Epoch Seed field value of the received EPP Epoch Settings

field.

AP\_MLD\_MAC is the MLD MAC address of the CPE AP.(#2072)

[…]

**TGbi Editor: *Instruction: Please modify the equation for CPE\_MHA\_block and its description in 10.71.3 as shown below***

* Establishing CPE MAC header anonymization parameter sets

[…]

CPE\_MHA\_block =*KDF*-*Hash*-*Length*( KDK, “CPE\_MHA\_block”,

Seed *||*AP\_MLD\_MAC*||* ((*n* + *p*) × EpochInterval) (#2071, #2072)

where

CPE\_MHA\_block is the block of bits that is partitioned into the sets of all possible

values for each CPE MHA parameter

KDF-*Hash*-*Length* is the key derivation function as defined in 12.7.1.6.2 (Key derivation

function (KDF)) using the hash algorithm identified by the AKM suite

selector (see Table 9-190 (AKM suite selectors))

KDK is the Key Derivation Key

*n* is the current number of the EPP epoch in the EPP epoch sequence as

defined in 10.71.2.4 (EPP Epoch Start Time Computation)

*Length* is the total number of bits to derive. A total of 1728 bits are derived for a

CPE\_MHA\_block.

*p* is the value of the latest exchanged non-AP MLD Specific Collision Epoch

Offset field if received and if n is greater or equal to colliding epoch number *c*

(see 10.71.2.5 (OTA MAC address collision avoidance); otherwise, *p* equals 0.

Seed is the value of the Group Epoch Seed field of the received EPP Epoch Settings

field.

AP\_MLD\_MAC is the MLD MAC address of the CPE AP.(#2072)

[…]

**TGbi Editor: *Instruction: Please modify the equation for BPE\_MHA\_block in 10.71.4 as shown below***

* Establishing BPE MAC header anonymization parameter sets

[…]

BPE\_MHA\_block = KDF-*Hash*-*Length* (PGTK, “BPE\_MHA\_block”, Seed ||AP\_MLD\_MAC|| (*n* × EpochInterval))(#2071, #2072),

where

KDF-*Hash*-*Length* is the key derivation function as defined in 12.7.1.6.2 (Key derivation

function (KDF)) using the hash algorithm identified by the AKM suite

selector (see Table 9-190 (AKM suite selectors))

PGTK is the Privacy Group Transient Key

*n* is the current number of the EPP epoch in the EPP epoch sequence as

defined in 10.71.2.4 (EPP Epoch Start Time Computation)

*Length* is the total number of bits to derive. A total of 960 bits are derived for a

BPE\_MHA\_block.

Seed is the value of the Group Epoch Seed field of the received EPP Epoch Settings

field.

AP\_MLD\_MAC is the MLD MAC address of the CPE AP.(#2072)

[…]