**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |
| --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) |
| Title | **Resolution of selected comments from SA ballot recirculation 1** |
| Date Submitted | 3 March 2020 |
| Source | Billy Verso (Qorvo), Frank Leong (NXP),Ben Rolfe (BCA) | Billy.Verso at qorvo.com |
| Re: | Contribution to TG4z for IEEE 802.15.4z to address selected comments  |
| Abstract | Contribution to TG4z amendment of IEEE Std 802.15.4-2015 |
| Purpose | This submission is the proposed resolution to address a number of comments on the SA Ballot recirculation #1 on draft 6 of the 802.15.4z amendment.  |
| Notice | This document does not represent the agreed views of the IEEE 802.15 Working Group or IEEE 802.15.8 Task Group. It represents only the views of the participants listed in the “Source(s)” field above. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. |
| Release | The contributor acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15. |
| Patent Policy | The contributor is familiar with the IEEE-SA Patent Policy and Procedures:<http://standards.ieee.org/guides/bylaws/sect6-7.html#6> and<http://standards.ieee.org/guides/opman/sect6.html#6.3>.Further information is located at <http://standards.ieee.org/board/pat/pat-material.html> and<http://standards.ieee.org/board/pat>. |

|  |
| --- |
| NOTE(s):  |

# Validity Code field

|  |  |
| --- | --- |
| Comment ID | 262292 |
| Comment # | R1-34 |

Re: subclause 7.4.4.39 on page 88 line # 17

**Proposed resolution:** REVISED

**Resolution detail:**

Replace “Validity Code” with “Application Code” and remove all mention of its use for validity or authentication purposes, which is achieved by the following set of changes:

1. in Figure 60—RSKD IE Content field format, change rightmost bottom cell from “Validity Code” to “Application Code” and rename “SVCP” field (bits 5-6) to “ACP”.
2. Replace the paragraph (p89 line3) describing the SVCP field with the following: “The ACP field indicates the presence of the Application Code field as per Table 24.”
3. Replace the paragraph (p89 line14) describing Validity Code Field with the following: “The Application Code field if present, provides a mechanism for the next higher layer to transfer additional application specific information relating to the use of the IE content. The Application Code field content is defined by the higher layers. The presence and length of the Application Code field is determined by the ACP field as per Table 24
4. Change the “SVCP field” appearing in the Table 24 caption and heading of its first column to “ACP field”.
5. In each row of Table 24 change “Validity Code field” to “Application Code field”
6. In Annex G, p182, change “Validity Code field” (on line 19) to “Application Code field”, and “SVCP field” (on line 20) to “ACP field”.

# Header IE to send STS seed information

|  |  |
| --- | --- |
| Comment ID | 262291 |
| Comment # | R1-33 |

Re: subclause 7.4.2.19 on page 77 line # 14

**Discussion:**

The functionality here is to allow for a header IE to convey STS seed information to devices which do not have the ability or encryption keys to decode the encrypted part of the frame, but nonetheless can benefit by using the STS to improve ranging accuracy of their ranging exchanges. As the commenter says a payload IE could be used for this in a non-secured frame, however that would require two separate frames – one to convey the STS seed info in a non-secure frame and the second to send whatever secured information is needed. The additional frame, especially when used as part of repeated ranging cycles, will lead to a significant increase in channel occupancy. Whether the Header IE is useful then depends on whether it can be delivered to the next higher layer when a secured receive frame fails to pass the receive security processing.

Upon analysis it seems that a MLME-COMMS-STATUS.indication can serve this purpose since in the 2020 revision it has a HeaderIeList parameter via which it can deliver the RSKD IE Header IE and content. The higher layers will know that the CRC was good, or the frame would have been discarded, and can infer that any STS seed info from the header IE is ostensibly as the non-authenticated sender intended, and while the security processing has failed, the upper layer can still choose to use the seed for the intended purpose of generating an STS.

To clarify that the MLME-COMMS-STATUS.indication delivers the RSKD IE Header IE content in the event of an incoming security processing error, the following resolution is proposed.

**Proposed resolution:** REVISED

**Resolution detail:** Insert the following new paragraph in 6.9.6.8 after paragraph one, i.e. directly after p34 line 7.

When receiving a received frame that contains an RSKD IE header IE, it is intended that the IE is delivered to the next higher layer to allow it to set the *phyHrpUwbStsKey*, *phyHrpUwbStsVCounter* and *phyHrpUwbStsVUpper96* attributes appropriately for STS generation. If a frame containing an RSKD IE header IE fails to pass the incoming security processing, for example if the receiver does not have the key to validate the MIC, the RSKD IE shall be delivered to the next higher layer in the HeaderIeList parameter of the MLME-COMM-STATUS.indication

# Use of “key” etc.

|  |  |
| --- | --- |
| Comment ID | 262289 |
| Comment # | R1-31 |

Re: subclause 6.9.6.5 on page 31 line # 10

**Discussion:**

Commenter objects to the use of “key” in the context of the DRBG used to generate the STS and points out that the NIST document [B23] defines a “Cryptographic Key (Key)” as “A parameter that determines the operation of a cryptographic function”. The quoted “such as” list however should not be treated as an exhaustive list, since using key in the context of a DRBG is clearly within the scope of [B23] as evidenced by the middle box of Figure 12 on p50 of [B23] which shows "key" and "V" (counter) parts of the DRBG. This particular figure maps directly onto the usage in the STS generation in the HRP UWB PHY.

To clarify this usage, it is proposed to define and use term “STS key” to refer to the key used in STS generation which may or may not be “privileged information” depending on the application needs.

In addition, in the context of [B23], the commenter is stating that all of the steps of [B23] are not included, which is true since the PHY is really only doing the core operation of generating the deterministic random STS pulse stream, however the API via PIB attributes allows the higher layers to re-seed as often as deemed necessary and fully follow [B23] in this respect. To address this, the wording saying we are *following the [B23] specification* will be changed to more correctly say *in line with the [B23] recommendation*.

**Proposed resolution:** REVISED

**Resolution detail:**

***In clause 3.2 add the following definitions***:

scrambled timestamp sequence (STS): a sequence of pseudo randomized pulses generated using a deterministic random bit generator (DRBG).

STS key: a parameter that is part of the seed input to the STS generating DRBG and which may be privileged information or not depending on the application needs.

***In the body of the text change every mention of “key” associated with the STS, or its DRBG, with “STS key”, which includes the following occurrences (editor to check for and similarly change any others missed):***

p33 line 26: ….. seed (i.e., STS key and data value V) …

p77 line 17 remove mention of NIST doc by deleting two sentences

~~For non-encrypted transmission of the IE, the DRBG is similar to NIST [B23], but not conforming to NIST due to part of the seed being transmitted non-encrypted. Non-encrypted transmission of the IE is intended to be used exclusively in non-secure (e.g., broadcast) ranging applications.~~

p84 line 10: ...... within a ranging session for which the same STS key is being used, …

p88 line 15: …… to convey and align the seed, (i.e., STS key and data), …

p88 line 17: change Figure 60—RSKD IE Content field format, “Key” field to “STS Key” field

p89 line 13: The STS Key field, if present, contains a 16-octet string to initialize the STS ~~generation~~ key.

p144 description of *phyHrpUwbStsKey*: This attribute specifies the STS key used in…..

p153 change first paragraph of “16.2.8.1 The STS generation DRBG” from:

~~The STS shall be generated using a DRBG following the specification in §10.2.1 of NIST Special Publication 800-90A Rev. 1 (2015) [B23]. The structure of the DRBG is shown in Figure 82. Each time the DRBG is run, it produces a 128-bit pseudo-random number used to form 128 pulses of the STS, as specified in 16.2.8.2. Higher layers should add pre-processing and/or re-seeding if specific levels of backtracking resistance are required and/or a very large number of iterations is performed using a single key~~.

To:

The STS shall be generated using a DRBG with the structure shown in Figure 82. This is using AES in counter mode in line with the recommendation in §10.2.1 of NIST Special Publication 800-90A Rev. 1 (2015) [B23]. Each time the DRBG is run, it produces a 128-bit pseudo-random number used to form 128 pulses of the STS, as specified in 16.2.8.2. Higher layers should add pre-processing and/or re-seeding if specific levels of backtracking resistance are required and/or a very large number of iterations is performed.

p154 line 2: change Figure 82 designation “128-bit Key” to “128-bit STS key”

p154 line 4: change “128-bit key” to “128-bit STS key”

p154 line 9: change “key” to “STS key”

p155 line 2: change “Key” to “STS key”

p155 line 4: delete NOTE paragraph:

~~NOTE—For non-encrypted transmission of the IE, the DRBG is similar to NIST [B23]. but not conforming to NIST due to part of the seed being transmitted non-encrypted. Non-encrypted transmission of the IE is intended to be used exclusively in non-secure (e.g., broadcast) ranging applications~~

p182 line 7: change “Key” to “STS key”

p182 line 14: change “key” to “STS key”

p182 line 15: change “key” to “STS key”

# Annex G related comments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Comment ID | 262296 | 262297 | 262298 | 262299 |
| Comment # | R1-38 | R1-39 | R1-40 | R1-41 |

**Discussion:** As presented in the draft this annex is the only place where the order of the octets of the seed (key and IV data) over the air are related directly to their order as fed into the AES block generating the STS. Part of the proposed resolution aims to capture this in the body text, however as words are sometimes open to interpretation differences, the annex provides a definitive reference to ensure implementers do the same thing and can successfully interwork, and as such it should be kept normative which the proposed resolution aims to clarify. As per the proposed resolution the unnecessary “shall”(s) that were inserted can be removed.

**Proposed resolution:** REVISED

**Resolution detail:**

1. Insert the following new paragraph directly after the paragraph on page 89 line 22:

The STS Key field and the V3, V2, V1 and V Counter fields that together define the seed for STS generation are strings of octets and as such are sent in the octet order typical for any string. When treating these as numbers in the context of [B23] and Figure 82, the octet received first in time is the treated as the most significant octet.

1. Insert the following new paragraph directly after the paragraph on page 154 lines 4 to 7:

A conforming implementation shall produce the output as given in Annex G.

1. Undo the excessive use of “shall” added between D5 and D6 by the following set of changes:

p182 line 8: change “shall be” to “are”

p182 line 30: change “shall” to “will”

p183 line 5: change “shall then be” to “are then”

***[END]***