**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |
| --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) |
| Title | **Proposed Fragment Authentication** |
| Date Submitted | 13 January 2013 |
| Source | [][][PO Box 798 Los Gatos CA 95006] | Voice: [ +1 408 395 7207 ]Fax: [ Deprecated ]E-mail: [ ben @ blindcreek.com ] |
| Re: | 802.14.4k LECIM Sponsor Ballot Comment Submission |
| Abstract | This contribution proposes changes to the balloted draft to provides for authenticating fragments with no additional overhead added to the fragment itself. This is submitted to support comments on the 802.15.4k initial sponsor ballot. |
| Purpose | Improve draft. |
| Notice | This document has been prepared to assist the IEEE P802.15. 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. |

Fragment Authentication with no per-Fragment overhead

Benjamin Rolfe, David Howard, Jonathan Simon

Contents

[Introduction 2](#_Toc345662344)

[Overview 3](#_Toc345662345)

[MPDU Counter 3](#_Toc345662346)

[Text changes to 4k Draft 2 4](#_Toc345662347)

# Introduction

A concern has been raised that a malicious actor may attack a fragmented frame transmission by inserting a bogus fragment that will pass the fragment validation check, but isn't really part of the MPDU. When reassembled MPDU is validated it will most likely be discarded, but the effort was wasted and energy will be spent trying again. The energy needed to perform this attack is less than, say, a brute force denial of service attack. With the potentially short duration of a single fragment, the malicious actor may substantially disrupt communications with very little on-air time and energy, thus making it difficult to detect. If the malicious fragment is detected and ignored, the bad actor would have to transmit many more packets to affect disruption, and would thus be more likely to be detected and dealt with appropriately.

This contribution proposes a solution that provides for authenticating fragments with no additional overhead added to the fragment itself. The overhead is limited to additional fields in the fragment context frame, and needs be present only the authentication protection is required. We suggest this feature be optional when fragmentation is implemented. We feel the complexity addition is small for implementations that include MAC security as defined in 802.15.4-2011.

This scheme assumes, as we do for all 802.15.4 security, that how keys are distributed is out of scope of this standard. The scheme described in this document uses the algorithms for the MIC defined in the base standard. It is assumed that anyone who would implement this security mechanism understands how the MIC-32 is calculated and used.

# Overview

The basic scheme is that when secure fragments are enabled, the MAC replaces the CRC-32 used for fragment validation with the MIC-32 as used in 802.15.4-2011.

To provide for the “used once” counter required for the MIC to be used (the “nonce”) of sufficient length, the fragment context frame will include a counter, which is used as part of a nonce value needed to compute a MIC. The fragment number transmitted with each fragment is combined with the transaction unique counter value to form the nonce. Thus the transaction counter forms the upper bits and the fragment number the low 6 bits.

The transaction setup message is a complete MPDU in a single fragment, as such may be secured when the PHY configuration supports the resulting message length.

# MPDU Counter

For security to work as intended, the nonce must not repeat between re-keying. The frame counter used in 7.3.2 is 32 bits. The MPDU counter field is 26 bits. Upon reception of a fragment this value is used as the most significant 26 bits and combined with the 6 bits of fragment number to form the 32 bit frame counter used in 7.3.2 to form the CCM\* nonce. With each MPDU transaction the MPDU counter value is incremented.

The 26 bit MPDU unique value will, with 1 MPDU per second, allow going for over 2 years between re-keying. In a typical LECIM device one message per hour may be more typical, in which case a device may go over 7000 years between rekeying. For device transmitting more often (not LECIM) and using security, and MPDU fragmentation, they will need to rekey somewhat more often than they might if not using fragmentation, which will depend on the average number of fragments per MPDU. This provides a realistic range of application tradeoffs.



Figure : Building the nonce for MIC calculation

# Text changes to d3P802-15-4k\_Draft\_Standard

The following explains the necessary changes to implement the fragment authentication described above.

**5.2.4.25 MPDU Fragment Sequence Context Description IE**

*In figure 48ww change “Reserved” field to “Secure Fragment”. Add a new field between “Addressing” and “PHY-dependent Parameter” fields named “MPDU Counter” with length shown as 0/4 octets.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Octets: 2** |  |  |  |
| **Bits: 1** | **1** | **10** | **3** | **1** | **10** | **6** |
| FragmentTx Option | Secure Fragment | TID | I-ACK Policy | TID Extension | MPDUSize /SuccessThreshold | Addressing Information |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **0/24/40** | **variable** | **0/4** | **variable** |
|  |  | 26 | 6 |  |
| TIDExtensionParameters | Addressing | MPDU Counter | Reserved | PHY-dependentParameters |

**Figure 48ww**

*Add description of Secure Fragment field after 5.2.4.25.1:*

The Secure Fragment field is used to indicate if the fragments in this transaction will be sent with authentication. When set, the MPDU Counter field shall be present in this IE, and the fragment validation field will be set to the MIC as described in 7.4. The field shall be set to 1 when *macMPDUFragSecure* is set to TRUE.

*Add description of MPDU Counter field after 5.2.4.25.8:*

The MPDU Counter field shall be present when the Secure Fragment field is set to 1 (*macMPDUFragSecure* is set to TRUE). The MAC shall maintain a counter that is incremented with each MPDU fragmentation transaction initiated such that the counter value is not repeated as described in 7.4.2.

**5.4.1.1 Fragment sequence context**

*Insert after last paragraph 5.4.1.1:*

When the fragment context frame is received with the Secure Fragment field set to 1, the MPDU Counter field will be used with the fragment number to form the Frame Counter field used to construct the nonce described in 7.2.1 except that *macFrameCounter* is replaced with *macFragmentFrameCounter*, which shall be comprised of the MPDU counter field, used as the most significant 26 bits, and the fragment number used as the least significant 6 bits.

**5.4.1.2 Fragment format**

*Change last paragraph of 5.4.1.2:*

The FVS field is used to validate the received fragment. When macMPDUFragSecure is FALSE, the length of the field shall be determined by *macFragmentFVSType*, and the FVS shall be calculated according to 5.2.1.9, except that the initial remainder value used for CRC calculation shall be as described in 5.2.4.25.7. When macMPDUFragSecure is TRUE, the length of the field shall be 4 octets and shall contain the MIC-32 as described in 5.4.1.1.

**6.4.3.2 General MAC PIB attributes for functional organization**

*Add new attributes to Table 52:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute  | Type | Range | Description | Default |
| *macMPDUFragSecure* | Boolean | TRUE, FALSE | When set to TRUE, a MIC shall be used as the fragment validation sequence, as described in 5.4.1.1. Otherwise, the FVS shall be calculated as in 5.4.1.2. | FALSE |
| *macFragmentFrameCounter* |  | 0x000000 – 0x3FFFFFF | The outgoing MPDU counter to use when *macMPDUFragSecure* is TRUE. Not used when *macMPDUFragSecure* is FALSE.  | 0x000000 |