**IEEE P802.15**

**Wireless Specialty Networks**

|  |  |
| --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Specialty Networks (WSNs) |
| Title | **TG 802.15.4ae List of changes to the IEEE Std 802.15.4** |
| Date Submitted | 9th September 2024 |
| Source | Tero Kivinen | E-Mail: kivinen@iki.fi |
| Abstract | List of changes needed in the IEEE Std 802.15.4 to add ASCON. |
| Purpose | Getting ready for draft proposal. |
| 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. |

1. Changes to Clause 9

Add ASCON algorithm to Table 9-9.

1. New annex Ba for specifying how to use AEAD ciphers in general

The new annex would specify how to use AEAD just like the Annex B now specifies how to use AES-CCM\* and AES-CCM modes of operation. This might not be even needed if the text in Section 9.3.2 is enough. Or we might just update the text in 9.3.2 to be sufficient.

**Ba.1 Introduction**

AEAD ciphers are defined in document “An Interface and Algorithms for Authenticated Encryption” RFC 5116. This annex defines how AEAD ciphers are going to be used in environments for which the use of variable-length authentication tags, rather than fixed-length authentication tags only, is beneficial. This annex encodes the authentication tag length in part of the nonce given to AEAD algorithm.

**Ba.2 Symmetric-key cryptographic building blocks**

This annex uses the notation, and representation of strings, integers and octets as described in 9.3.1

**Ba.2.1 Cipher**

This annex can use any cipher defined as AEAD algorithm.

**Ba.2.2 Parameters for AEAD algorithms**

Parameters for different AEAD ciphers are defined in Table 1.

|  |  |
| --- | --- |
| **AEAD Cipher** | **Parameters** |
| Ascon | * K length is 16 octets.
* N length is 16 octets.
* Does not need counter.
* MIC length shall be 4, 8, or 16 octets.
 |

**Ba.3 Specification of AEAD operations**

**Ba.3.1 Prerequisites**

The authenticated encryption operation has four inputs, each of which is an octet string (for specifics see RFC 5116 section 2.1):

* A secret key K
* A nonce N. Each nonce provided to distinct invocations of the Authenticated Encryption operation shall be distinct, for any particular value of the key. Nonces used in the IEEE Std 802.15.4 use 13-octet long part derived from addresses as specified in 9.3.2.
* A plaintext P, which contains the data to be encrypted and authenticated (string *m* of length l(*m*) octets).
* The associated data A, which contains the data to be authenticated, but not encrypted (string *a* of length l(*a*) octets).

There is a single output:

* A ciphertext C, which is at least as long as the plaintext, or an indication that the requested encryption operation could not be performed.

**Ba.3.1 AEAD encryption and authentication transform**

Generate N as follows:

1. Form the 1-octet Flags field consisting of the 1-bit Reserved field, the 1-bit Adata field, and particular 3-bit representations of the integers M and L = 2, as follows:

Flags = Reserved || Adata || M || L

Here, the 1-bit Reserved field is reserved for future expansions and shall be set to zero. The 1-bit Adata field is set to zero if length of authenticated additional data is zero and set to one if length of additional authenticated data is not zero. The M field is the 3-bit representation of the integer (M – 2)/2 in most-significant-bit-first order, where M is the length of the authentication tag in octets. The L field is the 3-bit representation of the integer L – 1, in most-significant-bit-first order, where L is always two.

1. Form a N as follows:

N = Flags || AEAD Nonce

Where AEAD Nonce is as defined in 9.3.2.

The generated N is used as fixed part of nonce as specified in the section 3.2 of RFC5116. The counter length used in nonce is always two. If the counter is not needed by the AEAD algorithm then fixed nonce is padded with two octets of 0x00.

 Do authentication encryption AEAD operation as specified in RFC5116 section 2.1.

**Ba.3.2 AEAD decryption and authentication checking transformations**

Calculate the N as in Ba.3.1 and do authenticated decryption AEAD operation as specified in RFC5116 section 2.2.

1. Create test vector document in mentor

Create document providing test vectors using ASCON and add reference to that in Table 9-9.

1. Add ASCON to bibliography

Add Ascon NIST specification, and RFC 5116 to the bibliography.