**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |  |
| --- | --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) | |
| Title | Resolution to SecurityMinimum and AllowedSecurityLevels | |
| Date Submitted | July, 16 2019 | |
| Source | [Tero Kivinen] [] | Voice: [] Fax: [] E-mail: [kivinen@iki.fi] |
| Re: | SECN discussion about how to solve SecurityMinimum AllowedSecurityLevels | |
| Abstract | Proposed resolution how to modify section 9 SecurityMinimum and AllowedSecurityLevels so they will work with multiple algorithms. | |
| Purpose | Solve the issue. | |
| 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. | |

# Description of Problem

In the security processing we have two way of filtering incoming frames. We can set the secSecurityMinimum to certain value, or we can fill in the secAllowedSecurityLevels. The secAllowedSecurityLevels is superset of the features provided by secSecurityMinimum. When we add new algorithms there is no longer distinct order of security of the algorithms and security levels. AES-CCM-128 is no more secure than AES-GCM-128. In the SECN we want to keep this feature but we need to be able to do policy checking based on the algorithm too.

# Solution

Solving the secAllowedSecurityLevels is quite easy, we just defined secAllowedSecurityLevels to be list of tupples, containing (algorithm id, security level) pairs. When matching the list we match both algorithm id and security level against the ones from key desceriptor. To solve the secSecurityMinimum easiest solution is to remove that completely, as the same feature can be delivered by using secAllowedSecurityLevels.

# Changes to 802.15.4

## Section 9.2.7 Incoming IE security level checking procedure

Remove step 1 completely:

~~1) If~~ *~~secIeAllowedSecurityLevels~~* ~~of the~~ *~~secIeSecurityLevelDescriptor~~* ~~is empty, then the procedure shall compare the SecurityLevel (as SEC1) with the~~ *~~secIeSecurityMinimum~~* ~~of the~~ *~~secIeSecurityLevelDescriptor~~* ~~(as SEC2) according to the algorithm described in 9.4.1.1. If this comparison evaluates to TRUE, the procedure shall set the IeStatus in the IeStatusList for this IE to PASSED.~~

Change step 2 as follows:

~~2~~1) ~~If~~ *~~secIeAllowedSecurityLevels~~* ~~of the~~ *~~secIeSecurityLevelDescriptor~~* ~~is not empty, the~~The procedure shall check whether the Algorithm Id, SecurityLevel tupple is equal to any of the elements of the *secIeAllowedSecurityLevels* of the *secIeSecurityLevelDescriptor*. If this check is successful, the procedure shall set the IeStatus in the IeStatusList for this IE to PASSED.

Renumber 3) to 2).

## Section 9.2.9 Incoming security level checking procedure

Remove step a completely:

~~a) If~~ *~~secAllowedSecurityLevels~~* ~~in SecurityLevelDescriptor is empty, then the procedure shall compare the SecurityLevel (as SEC1) with the~~ *~~secSecurityMinimum~~* ~~element of the SecurityLevelDescriptor (as SEC2) according to the algorithm described in 9.4.1.1. If this comparison evaluates to TRUE, the procedure shall return with Status set to PASSED.~~

Change step b as follows:

~~b~~a) ~~If~~ *~~secAllowedSecurityLevels~~* ~~in SecurityLevelDescriptor is not empty, the~~The procedure shall check whether the Algorithm Id, SecurityLevel tupple is equal to any of the elements of the *secAllowedSecurityLevels* of the SecurityLevelDescriptor. If this check is successful, the procedure shall return with Status set to PASSED.

Renumber c) to b) and d) to c).

## Section 9.5 Security-related MAC PIB attributes

In table 9-5 remove *secSecurityMinimum* entry completely.

In table 9-5 change description of *secDeviceOverrideSecurityMinimum* as follows:

Indication of whether originating devices for which the ~~secExempt~~*secExempt* is set may override the security level indicated by the *secAllowedSecurityLevels.* ~~AllowedSecurityLevels or SecurityMinimum.~~ If TRUE, this indicates that for originating devices with Exempt status, the incoming security level zero is acceptable, in addition to the incoming security levels meeting the ~~minimum expected~~ security levels indicated by the ~~SecurityMinimum~~ *secAllowedSecurityLevels* element.

In table 9-5 change description of *secAllowedSecurityLevels* as follows:

A set of allowed algorithm id, and security level~~s~~ tupples, as defined in Table 9-xx, and Table 9-6 respectively, for incoming MAC frames with the indicated frame type, and, if present, Command ID field. ~~If the set is empty, then the SecurityMinimum parameter applies instead.~~

In table 9-16 remove *secIeSecurityMinimum* entry completely.