**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |
| --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) |
| Title | **Text proposal for 15.4ab Secure compressed PSDU**  |
| Date Submitted | July 2023 |
| Sources | Rojan Chitrakar, Lei Huang, Bin Qian, David Xun Yang (Huawei) |  |
| Re: | Contribution to IEEE 802.15.4ab  |
| Abstract |  |
| Purpose | This submission proposes text to for the IEEE Std 802.15.4ab specification framework document.  |
| Notice | This document does not represent the agreed views of the IEEE 802.15 Working Group or IEEE 802.15.4ab Task Group. It represents only the views of the participants listed in the “Sources” field above. It is offered as a basis for discussion and is not binding on the contributing individuals. The material in this document is subject to change in form and content after further study. The contributors reserve the right to add, amend or withdraw material contained herein. |

Baseline for the below text is: IEEE 802.15.4-2020

9. Security

9.1 Overview

9.2 Functional description

9.2.2 Outgoing frame security procedure

*TG4ab editor, modify the subclause as follows (track changes ON):*

…

This procedure involves the following steps:

…

d) Check frame counter value.

1) If TSCH mode or compressed PSDU is not being used and the secFrameCounterPerKey in the KeyDescriptor is set to FALSE and secFrameCounter has the value 0xffffffff, the procedure shall return with a Status of COUNTER\_ERROR.

2) If TSCH mode or compressed PSDU is not being used and the secFrameCounterPerKey in the KeyDescriptor is set to TRUE and secKeyFrameCounter element of the KeyDescriptor has the value 0xffffffff, the procedure shall return with a Status of COUNTER\_ERROR.

e) Insert Auxiliary Security Header field. If compressed PSDU is not being used, the procedure shall insert the Auxiliary Security Header field in the frame to be secured, with the fields set as follows:

1) The Security Level field of the Security Control field shall be set to the SecurityLevel parameter.

2) The Key Identifier Mode field of the Security Control field shall be set to the KeyIdMode parameter.

3) If TSCH mode is being used, the Frame Counter Suppression field in the Security Control field shall be set to one. Otherwise, the Frame Counter Suppression field in the Security Control field shall be set to zero.

4) The Frame Counter field shall be set as follows:

i) If TSCH mode is being used, the Frame Counter field shall be omitted.

ii) If the secFrameCounterPerKey in the KeyDescriptor is set to TRUE, the Frame Counter

field shall be set to secKeyFrameCounter.

iii) Otherwise, the Frame Counter field shall be set to secFrameCounter.

5) If the KeyIdMode parameter is set to a value not equal to zero, the Key Source and Key Index fields of the Key Identifier field shall be set to the KeySource and KeyIndex parameters, respectively.

For compressed PSDUs, set the frame counter and block structure indicator as follows:

If a block structure exists and is known to both the transmitting and the receiving device (e.g., inside block structure), the frame counter is set as the indices of the slot, round and block in which the compressed PSDU is transmitted (see Figure 9-XX2 (Frame Counter field inside block structure)) and block structure indicator is set to 1. If a block structure does not exist and or is not known to the receiving device (e.g., outside block structure), the frame counter is set to 0 and the block structure indicator is set to 0.

f) Secure the frame. For the frames specified in Table 9-1, the Private Payload field and Open Payload field shall be set as indicated in the table. For frames not specified in Table 9-1, the Private Payload shall be set to the MAC Payload field, and Open Payload field shall be empty. For compressed PSDUs, the procedure shall then use the Private Payload field, the Open Payload field, the RPA\_hash field, RPA\_prand used to generate the RPA\_hash and the ID field of the compressed PSDU, the frame counter, the block structure indicator, the SecurityLevel parameter, and the secKey element of the KeyDescriptor to produce the secured compressed PSDU according to the transformation process defined in 9.3.5. For frames other than compressed PSDUs, the procedure shall then use the Private Payload field, the Open Payload field, the macExtendedAddress, the Frame Counter field (if TSCH is not being used), the ASN (if TSCH is being used), the SecurityLevel parameter, and the secKey element of the KeyDescriptor to produce the secured frame according to the transformation process defined in 9.3.5.

g) If not using compressed PSDU, store frame counter.

1) If not using TSCH mode and secFrameCounterPerKey in the KeyDescriptor is set to TRUE, the procedure shall increment secKeyFrameCounter element of the secKeyDescriptor by one.

2) If not using TSCH mode and secFrameCounterPerKey in the KeyDescriptor is set to FALSE, the procedure shall increment secFrameCounter by one.

h) Finish procedure. The procedure shall return with a Status of SUCCESS.

**Table 9-1—Exceptions to Private Payload field and Open Payload field definitions**

|  |  |  |
| --- | --- | --- |
| **Frame type** | **Private Payload field** | **Open Payload field** |
| Beacon(Frame Version < 2) | Beacon Payload field | All other fields in the MAC Payload field |
|
| MAC Command(Frame Version < 2) | Content field | Command ID field |
| SECURE-REPORT(from initiator) | TurnAroundTime | All other fields in the MessageContent field |
| SECURE-REPORT(from responder) | ReplyTime | All other fields in the MessageContent field |
| All other secured compressed PSDUs | MessageContent | Empty |

*TG4ab editor, insert a new subclause 9.2.12 after 9.2.11 as follows:*

**9.2.12 Incoming frame security procedure for compressed PSDUs**

This procedure shall only be used for incoming secured compressed PSDUs. For other frames, the procedure in 9.2.4 or 9.2.5 is used instead.

The input to this procedure is the compressed PSDU to be unsecured and the SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters. The inputs are as follows:

— SecurityLevel shall be set to *secAutoRequestSecurityLevel*

— KeyIdMode shall be set to *secAutoRequestKeyIdMode*

— KeySource shall be set to *secAutoRequestKeySource*

— KeyIndex shall be set to *secAutoRequestKeyIndex*

The outputs from this procedure are the status of the procedure and, if the status is SUCCESS the unsecured compressed PSDU.

All outputs of this procedure are assumed to be invalid unless and until explicitly set in this procedure.

This procedure involves the following steps:

1. **Check for *macSecurityEnabled***. If *macSecurityEnabled* is set to FALSE, the procedure shall return with a Status of UNSUPPORTED\_SECURITY.
2. **Check the SecurityLevel**. If the SecurityLevel is zero, the procedure shall return with a Status of UNSUPPORTED\_SECURITY.
3. **Set Device address**.DevicePanId shall be set to *macPanId*. The DeviceAddressingMode shall be set to NONE and the DeviceAddress shall be set to the address of the peer device.
4. **Obtain KeyDescriptor.** The procedure shall obtain the KeyDescriptor using the KeyDescriptor lookup procedure as described in 9.2.3 with using the KeyIdMode, KeyIndex, KeySource, DeviceAddressingMode, DevicePanId, and DeviceAddress. If KeyDescriptor lookup procedure fails, the procedure shall return with a Status of UNAVAILABLE\_KEY.
5. **Obtain DeviceDescriptor.** The procedure shall obtain the DeviceDescriptor using the DeviceDescriptor lookup procedure described in 9.2.6 using the DeviceAddressingMode, DevicePanId, and DeviceAddress. If that procedure fails, then the procedure shall return with a Status of UNAVAILABLE\_DEVICE.
6. **Set frame counter and block structure indicator.** If a block structure exists and is known to both the transmitting and the receiving device (e.g., inside block structure), the frame counter is set as the indices of the slot, round and block in which the compressed PSDU is received (see Figure 9-XX2 (Frame Counter field inside block structure)) and block structure indicator is set to 1. If a block structure does not exist and or is not known to the receiving device (e.g., outside block structure), the frame counter is set to 0 and the block structure indicator is set to 0.
7. **Unsecure compressed PSDU**. The Private Payload field and Open Payload field shall be set as indicated in the Table 9-1. The procedure shall then use the Private Payload field, the Open Payload field, the RPA\_hash field, RPA\_prand used to generate the RPA\_hash and the ID field of the compressed PSDU, the frame counter, the block structure indicator, the SecurityLevel parameter, and the secKey element of the KeyDescriptor to produce the unsecured compressed PSDU, according to the inverse transformation process described in the security operations, as described in 9.3.6. If the inverse transformation process fails, the procedure shall return with a Status of SECURITY\_ERROR.
8. **Return unsecured compressed PSDU**. The procedure shall return with the unsecured compressed PSDU, and a Status of SUCCESS.

9.3 Security Operations

9.3.3 AEAD Nonce

*TG4ab editor, insert the following subclause after 9.3.3.3 (AEAD nonce for Fragment frames):*

9.3.3.4 AEAD Nonce for compressed PSDU

The nonce for compressed PSDU shall be formatted as shown in Figure 9-XX1 (Nonce for compressed PSDU), with the leftmost field in the figure defining the first octets and the rightmost field defining the last octet of the nonce.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Octets: 3 | 3 | 1 | 4 | Bits: 0 | 1-3 | 4-7 |
| RPA\_hash | RPA\_prand | ID | Frame Counter  | Block Structure Indicator | Security Level | Reserved |

**Figure 9-XX1—Nonce for compressed PSDU**

The RPA\_hash field is set to the RPA\_hash field of the compressed PSDU to be secured.

The RPA\_prand field is set to the RPA\_prand field of the compressed PSDU to be secured, if such a field exist, else it is set as the RPA\_prand that is used to generate the RPA\_hash field of the compressed PSDU to be secured.

NOTE – In order to ensure the freshness of the nonce, it is important that the initiator does not reuse the same RPA\_hash and RPA\_prand for the same identity resolving key (IRK), especially outside the block structure.

The ID field is set to the ID field of the compressed PSDU to be secured.

If a block structure exists and is known to both the transmitting and the receiving device (e.g., inside block structure), the Frame Counter field is formatted as illustrated in Figure 9-XX2 (Frame Counter field inside block structure) and the Slot Index field, the Round Index field and the Block Index field are set as the indices of the slot, round and block in which the frame is transmitted or received respectively. If a block structure does not exist and or is not known to at least one of the transmitting or the receiving device (e.g., outside block structure), the Frame Counter field is reserved and set to 0.

|  |  |  |
| --- | --- | --- |
| Bits: 0-7 | 8-23 | 24-39 |
| Slot Index | Round Index | Block Index |

**Figure 9-XX2—Frame Counter field** **inside block structure**

The Block Structure Indicator field indicates whether the secured compressed PSDU is transmitted (or received) inside or outside of a block structure:

Zero: Outside block structure

One: Inside block structure

NOTE – The purpose the Block Structure Indicator field is to ensure that the nonce used for secured compressed PSDU transmitted inside and outside block structure is never reused and is always set to zero in a secured compressed PSDU transmitted outside the block structure and is set to one in a secured compressed PSDU transmitted inside the block structure.

The Security Level field is an unsigned integer that shall be set to the value of the security level negotiated during session setup. It is described in 9.4.2.2 (Security Level field).

9.3.4 AEAD prerequisites

*TG4ab editor, modify the subclause as follows (track changes ON):*

Securing a frame involves the use of the AEAD encryption and authentication transformation, as described in B.4.2. Unsecuring a frame involves the use of the AEAD decryption and authentication checking transformation, as described in B.4.3.

The length M of the Authentication field for the AEAD forward transformation and the AEAD inverse transformation is determined from Table 9-6, for compressed PSDUs using the security level negotiated during session setup and for frames other than compressed PSDUs using the Security Level field of the Security Control field of the auxiliary security header of the frame.

**9.3.5 AEAD transformation data representation**

**9.3.5.3 *a* data and *m* data**

*TG4ab editor, modify the subclause as follows (track changes ON):*

In the AEAD transformation process, the data fields shall be applied as in Table 9-3.

NOTE—The MHR contains the Auxiliary Security Header field, as defined in 7.2 for frames other than compressed PSDUs. For compressed PSDUs, the MHR is composed of the ID field, the RPA\_hash field, the RPA\_prand field if present and the MessageControl field of the compressed PSDU.

**9.3.6 AEAD inverse transformation data representation**

**9.3.6.3 c data and a data**

*TG4ab editor, modify the subclause as follows (track changes ON):*

In the AEAD inverse transformation process, the data fields shall be applied as in Table 9-5.

NOTE—The MHR contains the Auxiliary Security Header field, as defined in 7.2 for frames other than compressed PSDUs. For compressed PSDUs, the MHR is composed of the ID field, the RPA\_hash field, the RPA\_prand field if present and the MessageControl field of the compressed PSDU.

…

Baseline for the below text is: 15-22-0381-05-04ab-nba-uwb-ranging-text-proposal-for-15-4ab-tfd

* + - 1. Compressed PSDU messages

*TG4ab editor, add the following compressed PSDU messages as follows (track changes ON):*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase** | **Message Name** | **Octet 0 (Msg ID)** | **Octets 1-N [Len]** | **Description** |
| Initialization | SECURE-ADV-RESP | 0x80 | [RPA\_hash[3], MessageControl[1],MessageContent[], MIC[] | Secured version of the advertising response packet used by responder during initialization phase.Note - The frame content is same as that of ADV-RESP except that the CRC16 field is replaced with the MIC and the MessageContent is encrypted if a security level with encryption is negotiated.  |
| SECURE-SOR | 0x81 | [RPA\_hash[3], MessageControl[1],MessageContent[],MIC[] | Secured version of the start of ranging packet used by initiator during initialization phase.Note - The frame content is same as that of SOR except that the CRC16 field is replaced with the MIC and the MessageContent is encrypted if a security level with encryption is negotiated. |
| Control | SECURE-POLL | 0x82 | [RPA\_hash[3], RPA\_prand[3],MessageControl[1],MessageContent[],MIC[] | Secured version of the POLL frame with MessageControl=0x01.Note - The frame content is same as that of POLL frame with MessageControl=0x01 except that the CRC16 field is replaced with the MIC and the MessageContent is encrypted if a security level with encryption is negotiated. |
| SECURE-RESP | 0x83 | [RPA\_hash[3], MessageControl[1],MessageContent[],MIC[] | Secured version of the RESP frame with MessageControl=0x01.Note - The frame content is same as that of RESP frame with MessageControl=0x01 except that the CRC16 field is replaced with the MIC and the MessageContent is encrypted if a security level with encryption is negotiated. |
| SECURE-POLL (One-to-many) | 0x84 | [RPA\_hash[3], RPA\_prand[3],MessageControl[1],MessageContent[],MIC[] | Secured version of the POLL (One-to-many) frame with MessageControl=0x011.Note - The frame content is same as that of POLL (One-to-many) frame with MessageControl=0x011 except that the CRC16 field is replaced with the MIC and the MessageContent is encrypted if a security level with encryption is negotiated. |
| Report | SECURE-REPORT(from initiator) | 0x85 | [RPA\_hash[3],MessageControl[1],MessageContent[],MIC[] | Secured version of the REPORT(from initiator)MessageControl=0x00:MessageContent={PTDataLength[1],PTData[PTDataLength],TurnAroundTime[5]}, where PTDataLength and PTData fields are optionally present and represent pass through data to higher layers.Note - The frame content is same as that of the REPORT(from initiator) except that that the CRC16 field is replaced with the MIC and the TurnAroundTime field is situated at the end of the MessageContent field and is encrypted if a security level with encryption is negotiated. |
| SECURE-REPORT(from responder) | 0x86 | [RPA\_hash[3],MessageControl[1],MessageContent[],MIC[] | Secured version of the REPORT(from responder).MessageControl=0x00:MessageContent={PTDataLength[1],PTData[PTDataLength], ReplyTime[5]}, where PTDataLength and PTData fields are optionally present and represent pass through data to higher layers.MessageControl=0x01:MessageContent={Presence Bitmap[1],PTDataLength[1],PTData[PTDataLength],ReplyTime[5],NbaChannelMap[6], UWB PHY Config[3],UWB MAC Config[2],NB PHY Config[1]}, where PTDataLength and PTData fields are optionally present and represent pass through data to higher layers; and at least one of NbaChannelMap, UWB PHY Config, UWB MAC Config and NB PHY Config fields is present.MessageControl=0x02-0xff: reservedNote - The frame content is same as that of the REPORT (from responder) except that that the CRC16 field is replaced with the MIC and the fields other than the Presence Bitmap, PTDataLength and PTData[PTDataLength are situated at the end of the MessageContent field and are encrypted if a security level with encryption is negotiated. |
|  | Reserved | 0x87 - 0x8f |  | Reserved for future secured compressed frames |
|  | Reserved | 0x90 - 0xff |  | Reserved for future use |

* + - 1. Compressed PSDU message fields

*TG4ab editor, add the following to the end of the table (track changes ON):*

|  |  |  |
| --- | --- | --- |
| **Field name** | **Length in bits** | **Description** |
| MIC | 32 or 64 or 128 | Either the MIC-32 or the MIC-64 or the MIC-128 as described in 9.3 (Security Operations). |

References

[1] 15-23-0216-00-04ab-secured-compressed-frames-for-mms-ranging

[2] 15-22-0381-05-04ab-nba-uwb-ranging-text-proposal-for-15-4ab-tfd