IEEE P802.11
Wireless LANs

|  |
| --- |
| Clean up of FILS Container  |
| Date: 2014-01-22 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Jarkko Kneckt | Nokia | Otaniementie 19B, 02150 Espoo Finland  |  |  |
|  |  |  |  |  |

Abstract

The submission cleans the editorial challenges and poor structure of the FILS Container element, FILS Container TLV and element defragmentation.

The revision 1 also cleans up the clauses 8.4.2.1 and 10.44.3.1 Higherlayer setup using higher layer packet encapsulation.

**8.4.2 Information elements**

**8.4.2.1 General**

***Change the next to last paragraph of 8.4.2.1 as follows:***

The frame body components specified for many management subtypes result in elements ordered by ascending Element ID, with the exception of the MIC Management element (8.4.2.54 (Management MIC element)) and the Fragment element ( 8.4.2.189 (Fragment element)). If present, the MIC Management element appears at the end of the robust management frame body. See 9.24.6 (Element parsing) on the parsing of elements. If present, the Fragment element appears after ~~another~~  the element that it is fragmenting, or after the previous Fragment element. See 9.33(Element fragmentation).

**8.4.2.186 FILS Secure Container element**

***Instructions to the Editor. Make the changes to the clause 8.4.2.186 as shown below. The orginal text is 802.11ai D1.2.***

FILS ~~Secure~~ Container element includes one ~~or more~~ FILS Secure Container ~~Type Length Value(s) (~~TLV~~) (s)~~. The FILS Secure Container element is shown in figure 8-401db(FILS Secure Container element format).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Element ID | Length | FILS Container TLV |
| Octets: | 1 | 1 | Variable |

Figure 8-401db—FILS Secure Container element format

The Element ID and Length fields are defined in 8.4.2.1 (General).

FILS Secure Container TLV carries various purposes such as IP address assignment and GTK transfer. A FILS Secure Container TLV encoding consists of three fields: Type, Length, and Value field as shown in Figure 8-401dc~~v~~ (FILS Secure Container TLV format).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Type | Length | Value |
| Octets: | 1 | 2 | Variable |

Figure 8-401dc~~b~~—FILS Secure Container TLV format

The ~~first field,~~ Type field~~,~~ specifies the type of the data carried by the V~~v~~alue field, and it is unique within the FILS Secure Container TLVs~~element~~. The values of the Type field are shown in Table 8-183d(FILS Secure Container TLV).The ~~second field,~~ Length field~~,~~ specifies the actual length of the V~~v~~alue field in octets. The ~~third field, Length~~ Value field~~,~~ contains the data representing the value for the T~~t~~ype field.

If a FILS Secure Container TLV is too large to fit into a single FILS Secure Container element, the FILS Secure Container element is fragmented as described in 9.33 (Element Fragmentation).

**Table 8-183d—FILS Secure Container TLV**

|  |  |  |  |
| --- | --- | --- | --- |
| ~~Type~~ Name of FILS Secure Container TLV | Type ~~ID~~ | Length (octets) | Extensible |
| FILS HLP Wrapped ~~d~~Data | 1 | V~~v~~ariable but limitedby MMPDU | No |
| FILS IP Address Request | 2 | 4 to 255 | No |
| FILS IP Address Assignment | 3 | 4 to 255 | No |
| FILS DNS Information | 4 | 4 to 255 | No |
| KEY RSC | <ANA> | 19 | No |
| KDE Container | <ANA> | 4 to 255 | No |

|  |  |  |  |
| --- | --- | --- | --- |
|  | ~~Element ID~~ | ~~Length~~ | ~~FILS Secure Container TLV~~ |
| ~~Octets:~~ | ~~1~~ | ~~1~~ | ~~Variable~~ |

~~Figure 8-401db—FILS Secure Container element format~~

~~FILS Secure Container TLVs are used to carry out various purposes such as IP address assignment and GTK transfer.~~

~~If a FILS Secure Container TLV is too large to fit into a single element, the FILS Secure Container element is fragmented by using the Fragment elements (see 8.4.2.189 (Fragment element)).~~

**8.4.2.189 Fragment element**

***Instructions to the Editor. Make the changes to the clause 8.4.2.189 as shown below. The orginal text is 11-14-0003r2***

The payload of e~~E~~ach ~~information~~ element is limited to a maximum of 255 octets since their L~~l~~ength field is a single octet (Figure 8-104). If data to be represented in an element ~~IE~~ is too large ~~and the generic advertisement service (GAS) is not used~~, it is necessary to fragment the data~~, (see~~ as described in ~~section~~ 9.33(Element fragmentation) and9.34(Element defragmentation). The format of the Fragment element ~~IE~~ is indicated in Figure 8-183dx (Fragment element format ~~IE~~).

~~The length of the all but the final Fragment element shall be 255. The length of the final Fragment element depends on the amount of fragmented data left over. The length of a Fragment element shall always be nonzero.~~

|  |  |  |  |
| --- | --- | --- | --- |
|  | Element ID | Length | Fragmented Data |
| Octets: | 1 | 1 | Variable |

Figure 8-183dx—Fragment~~ed Data~~ element format

The Element ID and Length fields are defined in 8.4.2.1 (General).

The Fragmented Data field contains the data of the element that is fragmented as described in 9.33(Element fragmentation.

**9.33 Element Fragmentation**

***Instructions to the Editor. Make the changes to the clause 9.33 as shown below. The orginal text is 11-14-0003r2***

The general format of elements limits the size of each element to 255 octets. Data that is too large for a single element may be fragmented into a series of elements consisting of the original element into which the data would not fit, immediately followed by a number of Fragment elements.

The data to be fragmented is divided into M + N chunks, where

• M is the result of the integer division of the length of the data by 255

• N is equal to 1 if the length of the data modulo 255 is greater than 0, and equal to 0 otherwise

The original element into which the data would not fit is filled with the first chunk of data and is termed the leading element. The length of the leading element shall be 255. This element is immediately followed by M-1 Fragment elements, each containing the next chunk of data in a Fragmented Data field and with a length of 255. If N = 1 these elements are immediately followed by the last chunk of data in a Fragmented Data field of Fragment element which has a length equal to the length of the data modulo 255.

A Fragment element shall only follow another element whose length is 255. A Fragment element shall not be fragmented.

**9.34 Element ~~Reassembly~~ Defragmentation**

***Instructions to the Editor. Make the changes to the clause 9.34 as shown below. The orginal text is 11-14-0003r2***

Elements which have had their information fields fragmented are those that are followed by one or more Fragment elements. To reconstruct the original data the chunk of data from the leading element is concatenated, in order, with the chunks of data from the series of Fragmented Data field of Fragment elements that follow it. The defragmentation ~~reassembly~~ procedure finishes when any element other than a Fragment element is encountered or the end of the MMPDU is reached.

**10.44.3.1 Higherlayer setup using higher layer packet encapsulation**

***Instructions to the Editor. Make the changes to the clause 10.44.3.1 as shown below. The orginal text is 802.11ai D1.2***

The FILS HLP Wrapped Data TLV~~(s)~~ in the FILS Secure Container element (8.4.2.186.1) is used for encapsulating a higher layer protocol (HLP) frame~~(s)~~.

If a non-AP STA uses the higher layer frame encapsulation, the non-AP STA constructs the FILS HLP Wrapped Data TLV(s). When the non-AP STA transmits multiple HLP frames in an Association or a Reassociation Request frame, the non-AP STA shall construct one ~~multiple~~  FILS Secure Container element with FILS HLP Wrapped Data TLV~~s~~ for each HLP frame. The FILS Secure Container elements are included in the transmitted frame. ~~s . Then the non-AP STA transmits the Association/ or Reassociation Request frame including the FILS HLP wrapped data TLV in FILS Secure Container element~~. The FILS Secure Container element may be fragmented as described in 9.33 (Element fragmentation)~~by Fragment element (8.4.2.189)~~ ~~if required~~. The encapsulation procedure is following.

1 The non-AP STA prepares HLP MSDU(s) to transmit.

2) The non-AP STA fills FILS HLP Wrapped Data TLV(s) by the destination MAC address, the source MAC address and the HLP MSDU for each HLP MSDU~~s~~.

3) The non-AP STA encapsulates the FILS HLP Wrapped Data TLV~~(s)~~ into the FILS Secure Container element (8.4.2.186) and ~~F~~fragments the FILS Secure Container element~~(s)~~ if required.

…

If the AP receives HLP frames with the STA's MAC address, multicast address or broadcast address as the destination address from the network before transmitting Association/Reassociation Response, the AP transmits Association/Reassociation Response frame including the HLP frame(s) in the FILS Secure Container element including FILS HLP Wrapped DataTLV ~~of the FILS Secure Container element~~. The encapsulation procedure is described previously. If the AP does not receive HLP frames from the network targeted to the STA before transmitting Association/Reassociation Response, the AP transmits Association/Reassociation frame without the FILS Secure Container element including the HLP Wrapped Data TLV. The status code of Association/Reassociation Response is not affected whether or not the HLP frame is included in the Association/Reassociation Response frame.

When the non-AP STA receives Association/Reassociation Response with FILS Secure Container element including one or more HLP Wrapped Data TLVs, the non-AP STA decapsulates the HLP~~(~~s~~)~~ and generates MA-UNITDATA.indication primitive for each HLP MSDU~~(s)~~.