IEEE P802.11
Wireless LANs

|  |
| --- |
| Fragmenting Large IEs |
| Date: 2013-05-06 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Dan Harkins | Aruba Networks | 1322 Crossman ave, Sunnyvale, CA | +1 408 227 4500 | dharkins at aruba networks dot com  |
|  |  |  |  |  |

Abstract

This submission describes a scheme for fragmentation and reassembly of large data that cannot fit in a single IE.

***Instruct the editor to modify section 8.3.3.1 as indicated:***

**8.3.3.1 Format of Management Frames**

The frame body consists of the fields followed by the elements defined for each management frame subtype. All fields and elements are mandatory unless stated otherwise and appear in the specified, relative order, with the exception of the Fragment IE (8.4.2.188). The Fragment IE has no specified order but, instead, is placed immediately after, and at the same order as, the element that is being fragmented. STAs that encounter an element ID they do not recognize in the frame body of a received Management frame ignore that element and continue to parse the remainder of the management frame body (if any) for additional elements with recognizable element IDs. See 9.25.7 (Vendor specific element parsing). Unused element ID codes are reserved.

***Instruct the editor to modify section 8.4.2.183 as indicated:***

**8.4.2.183 FILS Public Key element**

The FILS Public Key element is used to communicate the device’s (certified) public key for use wth the FILS authentication exchange. The format of the FILS Public Key element is shown in Figure 8-401dd.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Element ID | Length | Key type | FILS public key |
| Octets: | 1 | 1 | 1 | variable (optional) |
|  | **Figure 8-401dd-- FILS Public Key element format(#1248)** |

Where the Key Type subfield is as follows:

 0: Reserved

 1: An X.509v3 certificate encoded according to RFC 5280.

 2. A raw public key encoded according to RFC 5480.

 3: A raw public key encoded according to RFC 3279.

If the length subfield is greater than 1, it indicates that the device’s public key is in the FILS public key field. If the length equals 1, it indicates that immediately following the FILS Public Key element is a series of one or more Fragment IEs (see 8.4.2.188) that are to be reconstructed according to 8.4.2.188.2 to obtain the device’s public key.

***Instruct the editor to add section 8.4.2.188, and subsections, to the draft and replace <ANA-X> with an appropriate figure identifier:***

**8.4.2.188 Fragmented IE and Fragment IE**

Information in Elements is limited to a maximum of 255 octets due to their native format (see Figure 8-104). If data to be represented in an IE is too large, it must be fragmented. Fragmenting an IE consists of encoding the IE with a length set to the length of the fixed-sized components of the element format—in other words, as if there was no data in the IE—and appending one or more Fragment IEs. The format of the Fragment IE is indicated in Figure 8-<ANA-X>.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Element ID | Length | Fragmented data |
| Octets: | 1 | 1 | variable |
| **Figure <ANA-X>-- Fragment IE(#1248)** |

The length of the all but the final Fragment IE shall be 255. The length of the final Fragment IE depends on the amount of fragmented data left over. The length of a Fragment IE shall always be non-zero.

**8.4.2.188.1 Fragmentation of a large Information Element**

Data that is too large for a single IE must be fragmented into a series of Fragment IEs.

First the IE to be fragmented is constructed with a length to indicate it is being fragmented—e.g. equal to the fixed components of the IE format—as defined by the particular IE. Next, the data to be fragmented is divided into *M* chunks, where *M* is the smallest value such that *M*\*255 is larger than the length of the data being fragmented. The length of all chunks except the last shall be 255. The length of the last chunk is equal to the length of the data being fragmented minus ((*M*-1)\*255). (Note: an IE with a particularly large fixed component could require fragmentation even if it only requires a single chunk). Each chunk is copied, in order, into the Fragmented data of its own Fragment IE and the length of each Fragment IE is set to the length of the chunk.

**8.4.2.188.2 Reassembly of a large Information Element**

IEs that are capable of being fragmented define the length that indicates that the IE is fragmented. Typically, this is the length of the fixed components of the IE format (as if it had zero data). When a fragmentable IE is encountered with such a length, the reassembly procedure begins to recover the IE’s data. The fragmented data from each Fragment IE in the series of Fragment IEs following the IE represent the chunks of the IE’s data. Each chunk is concatenated, in order, to reconstruct the data of the original IE. The reassembly procedure finishes when any IE other than the Fragment IE is encountered.

A Fragment IE that does not follow either another Fragment IE or an IE whose definition indicates it can be fragmented, shall be ignored.

**References:**