IEEE P802.11
Wireless LANs

|  |
| --- |
| Resolution of a Few Security Comments |
| Date: 2014-01-14 |
| 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 proposes resolutions to the following CIDs: 2416, 2426, 2436, 2445

***Instruct the editor to modify section 8.2.2 as indicated:***

**8.2.2 Conventions**

Values specified in decimal are coded in natural binary unless otherwise stated. The values in Table 8-2 (Valid type and subtype combinations) are in binary, with the bit assignments shown in the table. Values in other tables are shown in decimal notation.

Bit streams are an unordered string of bits. For evaluation purposes bit streams are interpreted as a sequence of octets with the most significant octet first and the most significant bit of an octet first.

***Instruct the editor to modify section 8.4.2.47 as indicated:***

**8.4.2.47 Fast BSS Transition element (FTE)**

The ANonce field contains a value chosen by the R1KH. It is encoded as a bit stream following the conventions in 8.2.2 (Conventions).

The SNonce field contains a value chosen by the S1KH. It is encoded as a bit stream following the conventions in 8.2.2 (Conventions).

***Instruct the editor to modify section 8.4.2.117 as indicated:***

**8.4.2.117 Authenticated Mesh Peering Exchange element**

**The Local Nonce field contains a nonce value chosen by the mesh STA that is sending the element. It is encoded as a bit steam following the conventions from 8.2.2 (Conventions).**

**The Peer Nonce field contains a nonce value that was chosen by the peer mesh STA or candidate peer mesh STA to which the element is being sent. It is encoded as a bit stream following the conventions from 8.2.2 (Conventions).**

***Instruct the editor to modify section 8.6.8.24 as indicated:***

**8.6.8.24 Public Key frame**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  Category | Action |  Public Key Frame Usage |  Group |  Public Key (optional) |

 Octets 1 1 1 2 variable

 **Figure 8-587—Public Key frame body format**

The Public Key Frame Usage field is set to a number to signify the usage mode of this frame. The Requets Types are shown in Table 8-271 (Request Type definitions).

 **Table 8-271—Request Type Definitions**

|  |  |
| --- | --- |
|  Public Key Frame Usage |  Value |
|  Request |  0 |
|  Response |  1 |
|  NAK |  2 |
|  Reserved |  3-255 |

The Publie Key Frame Usagefield is set to “Request” to indicate that a public key is being requested from a peer AP.

The Pubilc Key Frame Usagefield is set to “Response” to indicate that this frame is in response to a Public Key frame.

The Public Key Frame Usagefield is set to “NAK” to indicate rejection of a received Public Key frame.

***Instruct the editor to modify section 11.6.1.3 as indicated:***

**11.6.1.3 Pairwise key hierarchy**

* The PTK shall be derived from the PMK by

PTK 🡨 PRF-X(PMK, “Pairwise key expansion”, Min(AA,SPA) || Max(AA,SPA) ||

Min(ANonce,SNonce) || Max(ANonce,SNonce))

where X = 256 + TK\_bits. The value of TK\_bits is cipher-suite dependent and is defined in Table 11-4 (Cipher suite key lengths). The Min and Max operations for IEEE Std(#130) 802 addresses are with the address converted to a positive integer treating the first transmitted octet as the most significant octet of the integer. The nonces are bit steams and are evaluated by the Min and Max operations as specified in 8.2.2 (Conventions).

***Instruct the editor to modify section 11.6.1.4 as indicated:***

**11.6.1.4 Group key hierarchy**

1. X = (#1550) TK\_bits. The value of TK\_bits is cipher-suite dependent and is defined in Table 11-4 (Cipher suite key lengths). AA is represented as an IEEE Std(#130) 802 address and GNonce as a bit streamas defined in 8.2.2 (Conventions).

***Instruct the editor to modify section 11.6.1.6 as indicated:***

**11.6.1.6 PeerKey key hierarchy**

1. The STK shall be derived from the SMK by

STK 🡨 PRF-X(SMK, "Peer key expansion", Min(MAC\_I,MAC\_P) || Max(MAC\_I,MAC\_P) ||

Min(INonce,PNonce) || Max(INonce,PNonce))

where X = 256 + TK\_bits. The value of TK\_bits is cipher-suite dependent and is defined in Table 11-4 (Cipher suite key lengths). The Min and Max operations for IEEE 802 addresses are with the address converted to a positive integer treating the first transmitted octet as the most significant octet of the integer. The nonces are bit steams and are evaluated by the Min and Max operations as specified in 8.2.2 (Conventions).

***Instruct the editor to modify section 11.10.1 as indicated:***

**11.10.1 AP PeerKey overview**

The AP PeerKey protocol provides session identification and creation of an AP PeerKey association to provide for security of OBSS management communication between two APs.(#2421) The result of a successful run of the AP Peerkey protocol is an AP PeerKey association. An AP PeerKey association is composed of a Mesh PMKSA and a Mesh TKSA(#1711) .

Two APs perform the AP PeerKey protocol in order to protect HCCA TXOP Advertisement frames in an OBSS. The AP PeerKey protocol is unauthenticated (neither peer has a verified identity of the other peer) but an AP knows that only the peer AP that completed the AP PeerKey protocol is able to send protected HCCA TXOP Advertisement frames protected by the resulting AP PeerKey association. This allows an AP to determine whether a peer AP honors the HCCA TXOPs avoidance schedule that is negotiated. In this manner, an AP is able to honor the negotiated schedule of trusted peer APs and ignore peer APs that are not trustworthy. This allows trustworthy APs to negotiate mutually beneficial schedules while allowing an AP to not disadvantage itself in an OBSS in the presence of untrustworthy APs.

***Instruct the editor to modify section 13.5.7 as indicated:***

**13.5.7 Keys and key derivation algorithm for the authenticated mesh peering exchange (AMPE)**

The temporal key (MTK) shall be derived from the PMK by

MTK 🡨 KDF-X(PMK, “Temporal Key Derivation”, min(localNonce, peerNonce) ||

max(localNonce, peerNonce) || min(localLinkID, peerLinkID) ||

max(localLinkID, peerLinkID) || Selected AKM Suite ||

min(localMAC, peerMAC) || max(localMAC, peerMAC)).

CCMP uses X = 128. The “min” and “max” operations for IEEE 802 addresses are with the address converted to a positive integer, treating the first transmitted octet as the most significant octet of the integer as specified in 11.6.1.3 (Pairwise key hierarchy). The nonces are bit streams and are evaluated by the min and max operations as specified in 8.2.2 (Conventions). The LinkIDs are bit streams and are evaluated by the min and max operations as specified in 8.2.2 (Conventions).

**References:**