IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resolution of AP PeerKey Comments | | | | |
| Date: 2013-11-11 | | | | |
| 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 resolution to CIDs 2417, 2418, 2419, 2420, 2421, and 2422.

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

**8.6.8.24 Public Key frame**

The Public Key frame is transmitted by an AP to provide its public key to a peer AP and to request the peer’s

public key.

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

|  |  |
| --- | --- |
| Name | Usage mode |
| Request | 0 |
| Response | 1 |
| NAK | 2 |
| Refused | 3 |
| Reserved | 4-255 |

When the Request Type field is set to “NAK” or “Refused”, the Public Key field is not included.

***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 data confidentiality forcommunication between two APs. . An AP PeerKey association is composed of a Mesh PMKSA and a Mesh TKSA.

***Instruct the editor to modify section 11.10.2 as indicated:***

**11.10.2 AP PeerKey protocol**

An AP for which dot11ProtectedTXOPNegotiationActivated is true or dot11ProtectedQLoadReportActivated is true shall reply to a Public Key frame. An AP that has both dot11ProtectedTXOPNegotiationActivated is false and dot11ProtectedQLoadReportActivated is false shall drop all received Public Key frames. If the responding AP refuses to perform the AP PeerKey protocol with the initiating AP it shall respond with a Public Key frame, setting the Request Type to “Refused”. If the Group field in the public key request is a group that is supported by the responding AP, the AP shall reply with a public key of the same group as the request, generating such a key pair if required, and setting the Request Type field to “Response”.

If the initiating AP receives a Public Key frame from a peer AP with the Request Type field set to “NAK”, the initiating AP inspects the group field in the received “NAK”. If the group issupported, the initiating AP

should initiate to the peer AP using the indicated group. If the group is notsupported, the AP may choose to initiate to the peer AP using a different group from the dot11RSNAConfigDLCGroup table. Receipt of a

Public Key frame from a peer AP with the Request Type field set to “NAK” terminates the PeerKey

protocol.

If the initiating AP receives a Public Key frame from a peer AP with the Request Type field set to “Reject”, the initiating AP shall terminate the PeerKey protocol.

Note: an initiating AP that receives a “Reject” should delay sending another Request for at least 5 seconds, and should perform exponential back-off on all subsequent Requests by doubling the delay with each subsequent rejection. This standard does not specify a limit on the number of attempts the initiating AP makes to communicate with another AP.

If the initiating AP receives a Public Key frame from a peer AP with the Request Type field set to “Request” prior to receiving a “Response”, it checks the Group field in the request. If the Group in the received “Request” is the same as the Group the initiating AP sent it its request, the AP shall construct a Public Key frame with the Request Type field set to “Response” containing the public key it sent in its “Request” and transmit it to the peer. The AP shall then generate a PMK and a Mesh PMKSA, see below, and terminate the PeerKey protocol. If the Group field differs and the group indicated is supported, the AP shall respond to the peer AP by replying with a public key from that group, generating a key pair if required, and setting the Request Type field to “Response”. The AP shall generate a PMK and a Mesh PMKSA, see below, and terminate this run of the PeerKey protocol.

Once the AP and peer AP have exchanged public keys from the same finite cyclic group they can compute

the Diffie-Hellman shared secret using scalar-op() and function F from 11.3.4

(Finite cyclic groups):

Upon creation of the PMK, an AEK shall be created per 13.5.7 (Keys and key derivation algorithm for the authenticated mesh peering exchange (AMPE)). The Mesh PMKSA for this instance of the AP PeerKey protocol shall then be created using the AP’s BSSID as the STA’s MAC address, the peer AP’s BSSID as the peer STA’s MAC address, the AEK, the lifetime, and the PMKID. If a Mesh PMKSA created by a prior run of the AP PeerKey protocol exists it shall be deleted upon creation of the new PMKSA. All MeshPTKSAs created by the old Mesh PMKSA shall also be deleted.

Upon creation of the Mesh PMKSA, the APME protocol (as defined in 13.5 (Authenticated mesh peering exchange (AMPE)) shall be used to prove possession of the PMK (and implicitly the private key that corresponds to the peer’s public key) and generate the Mesh PTKSA.

**References:**