IEEE P802.11
Wireless LANs

|  |
| --- |
| Resolution of Security Comments from the 4th Sponsor Ballot Recirculation |
| Date: 2011-05-10 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Dan Harkins | Aruba Networks | 1322 Crossman ave, Sunnyvale, CA | +1 408 227 4500 | dharkins at arubanetworks dot com |
|  |  |  |  |  |

Abstract

This document proposes changes to draft 11.01 to address CIDs 4030, 4032, 4033, 4034, 4035, 4036, 4037, 4038, 4053, 4062, 4067, 4103, 4104, 4105, 4106, 4108, 4157, 4158, 4159, 4160, 4161, 4162, 4163, 4271, 4272, 4273, 4279, 4280, 4281

***Modify section 3 as indicated:***

* Definitions

**mesh peering management:** A group of protocolsto facilitate the mesh peering establishment and closure of the mesh peerings.

***Modify section 5.4.3.2 as indicated:***

* Deauthentication

When the deauthentication service is terminating SAE authentication any PTKSA,GTKSA, Mesh TKSA or Mesh GTKSA related to this SAE authentication is destroyed. If PMK caching is not enabled, deauthentication also destroys any PMKSA created as a result of this successful SAE authentication

***Modify section 7.3.2.98.1 as indicated:***

* General

The Mesh Configuration element shown in Figure 7-95o130 (Mesh Configuration element format) is used to advertise mesh services. It is contained in Beacon frames and Probe Response frames transmitted by mesh STAs, and is also contained in Mesh Peering Open and Mesh Peering Confirm frames.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ElementID | Length | Active Path Selection Protocol Identifier | Active Path Selection Metric Identifier | Congestion ControlMode Identifier | Synchronization Method Identifier | Authentication Protocol Identifier |  | Mesh Formation Info | Mesh Capability |
| Octets:1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 |
| * Mesh Configuration element format
 |

***Delete section 7.3.2.98.7 and renumber the subsequent subclauses in 7.3.2.98 appropriately***

***Modify section 7.3.2.102 as indicated:***

* Mesh Peering Management element

The Mesh Peering Management element is used to manage a mesh peering with a neighbor mesh STA. The format of the Mesh Peering Management element is shown in Figure 7-95o137 (Mesh Peering Management element format).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Element ID | Length | Mesh Peering Protocol Identifier | Local Link ID | Peer Link ID (conditional) | Reason Code (conditional) | Chosen PMK (optional) |
| Octets: 1 | 1 | 1 | 2 | 2 | 2 | 16 |
|  | * Mesh Peering Management element format
 |

The Element ID is set to the value given in Table 7-26 (Element IDs) for this element.

The Length field is set to the number of octets in the Mesh Peering Management element following the Length field itself.

The Mesh Peering Protocol Identifier field indicates the type of mesh peering protocol that is currently used to establish mesh peerings. Table 7-43bj6 (Mesh Peering Protocol Identifier field values) provides the mesh peering protocol identifier values defined by this standard.

|  |
| --- |
| Table 7-43bj6-- Mesh Peering Protocol Identifier field values  |
| Value | Meaning |
| 0 | Mesh peering management protocol |
| 1 | Authenticated mesh peering exchange protocol |
| 2–254 | Reserved |
| 255 | Vendor specific(The active mesh peering protocol is specified in a Vendor Specific element) |

When the Mesh Peering Protocol Identifier field is 255, the active mesh peering protocol is specified by a Vendor Specific element that is present in the frame. The content of the Vendor Specific element is beyond the scope of this standard. (See 7.3.2.26.)

The Local Link ID field is the unsigned integer value generated by the local mesh STA to identify the mesh peering instance.

The conditional components of the Mesh Peering Management element are present depending on the Action field value of the frame in which the Mesh Peering Management element is conveyed.

The Peer Link ID field is the unsigned integer value generated by the peer mesh STA to identify the mesh peering instance. This field is not present for the Mesh Peering Open frame, is present for the Mesh Peering Confirm frame, and is optionally present for the Mesh Peering Close frame. The presence or absence of the Peer Link ID in a Mesh Peering Close is inferred by the Length field.

The Reason Code field enumerates reasons for sending a Mesh Peering Close. It is present for the Mesh Peering Close frame and is not present for Mesh Peering Open or Mesh Peering Confirm frames. The reason code is defined in 7.3.1.7 (Reason Code field).

The Chosen PMK field is present when dot11MeshSecurityEnabled is true and a PMK is shared between the transmitter and receiver of the frame containing the element. It contains the PMKID that identifies the PMK used to protect the Mesh Peering Management frame.

***Modify section 7.3.2.118 as indicated:***

* Authenticated Mesh Peering Exchange element

The GTKdata field is optional. When present, it contains the bit string of {GTK || Key RSC || GTKExpirationTime} as the GTK data material. When present, the GTKdata field is protected by the exchange in which it is contained (see 11C.5 (Authenticated mesh peering exchange (AMPE))). The Key RSC denotes the last frame sequence number sent using the GTK and is specified in Table 8-3 of 8.5.2. GTKExpirationTime denotes the key lifetime of the GTK in seconds and the format is specified in Figure 8-31 of 8.5.2.

***Insert the following to the end of 7.4.14.1***

* Self-protected(Ed) Action fields

The Mesh Peering Open frame, the Mesh Peering Confirm frame, and the Mesh Peering Close frame are referred to as Mesh Peering Management frames.

***Modify 7.4.14.2.2 as indicated:***

* Mesh Peering Open frame details

The Mesh Peering Open frame is used to open a mesh peering using the procedures defined in 11C.3.6 (Mesh peering open) and in 11C.5.5 (Mesh Peering Management frames for AMPE). The Mesh Peering Open frame is also, together with Mesh Peering Confirm and Mesh Peering Close frames, referred to as a Mesh Peering Management frame. The format of the Mesh Peering Open frame Action field is shown in Table 7-57v25 (Mesh Peering Open frame Action field format).

|  |
| --- |
| * Mesh Peering Open frame Action field format
 |
| Order | Information | Notes |
|  | Category |  |
|  | Self-protected(Ed) Action |  |
|  | Capability |  |
|  | Supported Rates |  |
|  | Extended Supported Rates | The Extended Supported Rates element is present if there are more than eight supported rates, and is optionally present otherwise. |
|  | Power Capability | The Power Capability element is present if dot11SpectrumManagementRequired is true. |
|  | Supported Channels | The Supported Channels element is present if dot11SpectrumManagementRequired is true and dot11ExtendedChannelSwitchEnabled is false. |
|  | RSN | The RSN element is present only if dot11MeshSecurityActivated is true. |
|  | Mesh ID | The Mesh ID field is set as described in 7.3.2.99 (Mesh ID element). |
|  | Mesh Configuration  | The Mesh Configuration field is set as described in 7.3.2.98 (Mesh Configuration element). |

***Modify section 8.1.3 as indicated:***

* RSNA Establishment

Change the contents of item b) and c) in 8.1.3 as follows:

* If an RSNA is based on a PSK or password in an ESS, the SME establishes an RSNA as follows:
* It identifies the AP as RSNA-capable from the AP’s Beacon or Probe Response frames.
* If the RSNA-capable AP advertises support for SAE authentication in its Beacon or Probe Response frames, and the STA has a group defined in the dot11RSNAConfigDLCGroupTable and a password for the AP in the dot11RSNAConfigPasswordValueTable, the STA shall invoke SAE authentication to establish a PMK. If the RSNA-capable AP does not advertise support for SAE authentication in its Beacon and Probe Response frames but advertises support for the alternate form of PSK authentication (see 5.8.2.2 (Alternate Ooperations with PSK)), and the STA also supports the alternate form of PSK authentication, the STA may ~~It shall~~ invoke Open System authentication and use the PSK as the PMK with the key management algorithm in step 4) below.

***Modify section 8.2a.1 as indicated:***

* SAE overview

SAE shall be implemented on all mesh STAs to facilitate and promote interoperability.

***Modify section 8.2a.2 as indicated:***

* Assumptions on SAE

When used with AKMs 00-0F-AC:8 or 00-0F-AC:9 from Table 7-34 (AKM suite selectors), H is instantiated as HMAC-SHA256:

H(salt, ikm) = HMAC-SHA256(salt, ikm) .

When used with AKMs 00-0F-AC:8 or 00-0F-AC:9 from Table 7-34 (AKM suite selectors), CN is instantiated as a function that takes a key and a sequence of data. Each piece of data is converted to an octet string and concatenated together before being passed, along with the key, to HMAC-SHA256:

CN(key, X, Y, Z, …) = HMAC-SHA256(key, D2OS(X) || D2OS(Y) || D2OS(Z) || …)

***Modify section 8.4.1.1.3b as indicated:***

* Mesh GTKSA

The mesh GTKSA results from a successful AMPE or mesh group key handshake, and is unidirectional. In an MBSS, each mesh STA defines its own “transmit mesh GTKSA”, which is used to encrypt its group addressed transmissions. Also each mesh STA stores a separate “receive mesh GTKSA” for each peer mesh STA so that encrypted group addressed traffic received from the peer mesh STAs may be decrypted.

***Modify section 8.8.1 as indicated:***

* Keys and key derivation algorithm

To execute the authenticated mesh peering exchange (AMPE), and mesh group key handshake with a candidate peer mesh STA, the mesh STA shall derive an authenticated encryption key (AEK) and a mesh temporal key (MTK) using the PMK it shares with the candidate peer mesh STA.

The AEK is derived statically from the shared PMK. The MTK is derived from the shared PMK and dynamic information provided by the mesh STA and candidate peer mesh STA.

The AEK is mutually derived by the local mesh STA and the peer mesh STA once a new PMK has been selected. The AEK shall be derived from the PMK by

AEK KDF-256(PMK, “AEK Derivation”, Selected AKM Suite ||
min(localMAC, peerMAC) || max(localMAC, peerMAC)).

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 8.5.1.2. The min and max operations for nonces are with the nonces treated as positive integers converted as specified in 7.1.1.

The MTK is used to protect communications between two peer mesh STAs. The local mesh STA and peer mesh STA derive an MTK per peering instance and may rekey the MTK using AMPE.

***Modify section 11A.4.2 as indicated:***

* FT initial mobility domain association in an RSN

A STA indicates its support for the FT procedures by including the MDIE in the (Re)Association Request frame and indicates its support of security by including the RSNIE. The AP responds by including the FTIE, MDIE, and RSNIE in the (Re)Association Response frame. After a successful IEEE 802.1X authentication (if needed) or SAE authentication, the STA and AP perform an FT 4-Way Handshake. At the end of the sequence, the IEEE 802.1X Controlled Port is opened, and the FT key hierarchy has been established. The message flow is shown in Figure 11A-2.

~~Upon successful completion of the IEEE 802.1X authentication,~~ If IEEE 802.1X authentication was performed, then upon successful completion of authentication the R0KH receives the MSK and authorization attributes. If SAE authentication was performed, the R0KH receives the PMK resulting in the successful completion of SAE. If a key hierarchy already exists for this non-AP STA belonging to the same mobility domain (i.e., having the same MDID), the R0KH shall delete the existing PMK-R0 security association and PMK-R1 security associations. It then calculates the PMK-R0, PMKR0Name, and PMK-R1 and makes the PMK-R1 available to the R1KH of the AP with which the STA is associated.

***Modify section 11C.2.3 as indicated:***

* Mesh profile

A mesh profile is a set of parameters that specifies the attributes of a mesh BSS. A mesh profile consists of the following:

* A Mesh ID—specified by dot11MeshID
* A path selection protocol identifier—specified by dot11MeshActivePathSelectionProtocol
* A path selection metric identifier—specified by dot11MeshActivePathSelectionMetric
* A congestion control mode identifier—specified by dot11MeshActiveCongestionControlMode
* A synchronization method identifier—specified by dot11MeshActiveSynchronizationMethod
* An authentication protocol identifier—specified by dot11MeshActiveAuthenticationProtocol

***Modify section 11C.3.1 as indicated:***

* General

The mesh peering management (MPM) protocol is used to establish, maintain, and close mesh peerings between mesh STAs when dot11MeshSecurityActivated is false. When dot11MeshSecurityActivated is true, the peers establish an authenticated mesh peering using the authenticated mesh peering exchange (AMPE) protocol. The AMPE protocol requires an existing mesh PMKSA. If a mesh PMKSA with the candidate peer mesh STA exists it AMPE shall use that PMKSA. If no mesh PMKSA exists the peers shall first authenticate to establish a mesh PMKSA, see 11C.5 (Authenticated mesh peering exchange (AMPE)).

***Modify section 11C.3.3 as indicated:***

* Mesh authentication

In order to create a secure peering, mesh STAs first authenticate each other and create a mesh PMKSA. This can be done using either SAE or IEEE 802.1X. Mesh STAs shall support SAE authentication (see 8.2a (Authentication using a password)) using a pre-shared secret with the candidate peer mesh STA. Optionally, mesh STAs may support IEEE 802.1X authentication (see 5.8 (IEEE Std 802.11 and IEEE Std 802.1X-2004)).

When dot11MeshActiveAuthenticationProtocol is sae (1) the scanning mesh STA shall initiate SAE to the candidate mesh STA. If SAE terminates unsuccessfully, the scanning mesh STA shall terminate the peering establishment procedure. Otherwise, the PMK that results from successful SAE authentication shall be used to create a mesh PMKSA.

When dot11MeshActiveAuthenticationProtocol is ieee8021x (2), then the scanning mesh STA shall initiate the MPM protocol to establish a peering. If the MPM protocol fails then the scanning mesh STA shall terminate the peering establishment procedure. Otherwise, IEEE 802.1X authentication shall be performed between the two peers according to the following:

* If only one mesh STA has the Connected to AS field set to 1, that STA shall act as the IEEE 802.1X authenticator and the other STA shall act as the IEEE 802.1X supplicant;
* If both mesh STAs have the Connected to AS field set to 1, then the mesh STA with the higher MAC address shall act as the IEEE 802.1X authenticator and the other mesh STA shallact as the IEEE 802.1X supplicant (see 8.5.1 (Key hierarchy) for MAC address comparison).

***Modify section 11C.3.4.1 as indicated:***

* Overview

A mesh STA uses a mesh peering instance controller to manage all mesh peering instances.

The mesh peering instance controller performs the following functions:

* Create and destroy MPM finite state machines and AMPE finite state machines
* Manage instance identifiers for each mesh peering instance
* Manage mesh TKSAs for each mesh peering instance when dot11MeshSecurityActivated is true
* Pre-process the incoming Mesh Peering Management frames and pass the frames to the finite state machine with matching instance identifier
* Pass internal commands to the finite state machine with matching instance identifier

A mesh peering instance is identified by a mesh peering instance identifier. The mesh peering instance identifier is the set of: localLinkID, localMAC, and peerMAC.

***Modify section 11C.3.4.2 as indicated:***

* Creating a new mesh peering instance

The mesh peering instance controller creates a new mesh peering instance after either of the following two events:

* The receipt of a Mesh Peering Open frame from a candidate peer mesh STA according to the rules of 11C.3.5 (Mesh peering instance selection).
* The receipt of an MLME-MESHPEERINGMANAGEMENT.request primitive with a Mesh Peering Open frame.

***Modify section 11C.3.4.3 as indicated:***

* Deleting mesh peering instances

The mesh peering instance controller deletes a mesh peering instance after either:

* Expiry of a holding timer (see 11C.4.4 (Timers))
* The acceptance of a peer’s response to an existing request to close the peering (see 11C.4.3 (Events and actions)).

 When the deletion occurs, the mesh TKSA that is bound to the mesh peering shall be deleted.

***Modify section 11C.3.5 as indicated:***

* Mesh peering instance selection

The content of a Mesh Peering Management frame received from a candidate peer mesh STA, and the set of mesh peering instances in the mesh peering instance controller determine whether

* A new mesh peering instance is created (see 11C.3.4.2 (Creating a new mesh peering instance)); or,
* An existing mesh peering instance is updated

If dot11MeshSecurityActivated is true and the mesh STA shares a PMK with the candidate peer mesh STA but the Mesh Peering Protocol Identifier field in the Mesh Configuration element of the frame indicates “mesh peering management protocol,”, the frame shall be silently discarded.

If dot11MeshSecurityActivated is true and the mesh STA shares a PMK with the candidate peer mesh STA but either the Mesh Peering element or the MIC element are not present in the frame, the frame shall be silently discarded.

If dot11MeshSecurityActivated is false but the Mesh Peering Protocol Identifier field in the Mesh Peering Management element of the received frame indicates “authenticated mesh peering exchange”, the frame shall be silently discarded.

If dot11MeshSecurityActivated is false but either the Mesh Peering element or the MIC element are present in the frame, the frame shall be silently discarded.

If the frame contains a group address in TA or RA, it shall be silently discarded.

If the incoming Mesh Peering Management frame is for AMPE and the Chosen PMK from the received frame contains a PMKID that does not identify a valid mesh PMKSA, the frame shall be silently discarded.

If the Mesh Peering Management frame has not been silently discarded, the mesh peering instance controller attempts to locate a matching mesh peering instance identifier. A match is determined by comparing the contents of the Mesh Peering Management frame with each peering instance. A match is found if all the following conditions are true:

***Modify Annex D as indicated:***

Dot11MeshSTAConfigEntry ::=

 SEQUENCE {

 dot11MeshID OCTET STRING,

 dot11MeshNumberOfPeerings Unsigned32,

 dot11MeshAcceptingAdditionalPeerings TruthValue,

 dot11MeshConnectedToMeshGate TruthValue,

 dot11MeshSecurityActivated TruthValue,

 dot11MeshActivePeeringProtocol INTEGER,

 dot11MeshMaxRetries Unsigned32,

 dot11MeshRetryTimeout Unsigned32,

 dot11MeshConfirmTimeout Unsigned32,

 dot11MeshHoldingTimeout Unsigned32,

 dot11MeshConfigGroupUpdateCount Unsigned32,

 dot11MeshActivePathSelectionProtocol INTEGER,

 dot11MeshActivePathSelectionMetric INTEGER,

 dot11MeshForwarding TruthValue,

 dot11MeshTTL Unsigned32,

 dot11MeshGateAnnouncementProtocol TruthValue,

 dot11MeshGateAnnouncementInterval Unsigned32,

 dot11MeshActiveCongestionControlMode INTEGER,

 dot11MeshActiveSynchronizationMethod INTEGER,

 dot11MeshNbrOffsetMaxNeighbor Unsigned32,

 dot11MBCAActivated TruthValue,

 dot11MeshBeaconTimingReportInterval Unsigned32,

 dot11MeshBeaconTimingReportMaxNum Unsigned32,

 dot11MeshDelayedBeaconTxInterval Unsigned32,

 dot11MeshDelayedBeaconTxMaxDelay Unsigned32,

 dot11MeshDelayedBeaconTxMinDelay Unsigned32,

 dot11MeshAverageBeaconFrameDuration Unsigned32,

 dot11MeshSTAMissingAckRetryLimit Unsigned32,

 dot11MeshAwakeWindowDuration Unsigned32,

 dot11MCCAImplemented TruthValue,

 dot11MCCAActivated TruthValue,

 dot11MAFlimit Unsigned32,

 dot11MCCAScanDuration Unsigned32,

 dot11MCCAAdvertPeriodMax Unsigned32,

 dot11MCCAMinTrackStates Unsigned32,

 dot11MCCAMaxTrackStates Unsigned32,

 dot11MCCAOPtimeout Unsigned32,

 dot11MCCACWmin Unsigned32,

 dot11MCCACWmax Unsigned32,

 dot11MCCAAIFSN Unsigned32

 }

dot11RSNAConfigPasswordCredential OBJECT-TYPE

 SYNTAX OCTET STRING

 MAX-ACCESS read-write

 STATUS current

 DESCRIPTION

 "This is a control variable.

 It is written by an external management entity.

 Changes take effect as soon as practical in the implementation.

 This variable is a binary representation of a shared,

 secret, and potentially low-entropy word, phrase, code

 or key used as an authentication credential.

 Any character-based word or phrase shall be converted

 into a canonical binary representation according to

 8.2a.3 before populating the Password Credential."

 ::= { dot11RSNAConfigPasswordValueEntry 2 }

dot11RSNAConfigDLCGroupIdentifier OBJECT-TYPE

 SYNTAX Unsigned32

 MAX-ACCESS read-write

 STATUS current

 DESCRIPTION

 "This is a control variable.

 It is written by an external management entity.

 Changes take effect as soon as practical in the implementation.

 This variable uniquely identifies a domain parameter

 set for a group in the IANA registry `Group Description'

 attributes for RFC 2409 (IKE)."

 ::= { dot11RSNAConfigDLCGroupEntry 2 }

dot11MeshComplianceGroup OBJECT-GROUP

 OBJECTS {

 -- dot11MeshSTAConfigTable

 dot11MeshID,

 dot11MeshNumberOfPeerings,

 dot11MeshAcceptingAdditionalPeerings,

 dot11MeshConnectedToMeshGate,

 dot11MeshSecurityActivated,

 dot11MeshActiveAuthenticationProtocol,

 dot11MeshMaxRetries,

 dot11MeshRetryTimeout,

 dot11MeshConfirmTimeout,

 dot11MeshHoldingTimeout,

 dot11MeshActivePathSelectionProtocol,

 dot11MeshActivePathSelectionMetric,

 dot11MeshForwarding,

 dot11MeshTTL,

 dot11MeshGateAnnouncementProtocol,

 dot11MeshActiveCongestionControlMode,

 dot11MeshActiveSynchronizationMethod,

 dot11MeshNbrOffsetMaxNeighbor,

 dot11MBCAActivated,

 dot11MCCAImplemented,

 dot11MCCAActivated }

 STATUS current

 DESCRIPTION

"This object class provides the objects from the IEEE 802.11 MIB required to manage mandatory mesh functionality. Note that additional objects for managing mesh functionality are located in the dot11MeshOptionGroup, dot11MeshHWMPComplianceGroup, and dot11PasswordAuthComplianceGroup."

 ::= { dot11Groups 56}

**References:**