IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resolution of Some Additional Comments | | | | |
| Date: 2016-03-14 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Dan Harkins | HPE | 1322 Crossman avenue, Sunnyvale, California,  United States of America | +1 408 227 4500 |  |
|  |  |  |  |  |

Abstract

This submission proposes resolutions to 7249, 7318, 7319, 7335, 7339, 7341, 7571, 7710, 7728, 7729, 7730, and 7740.

**802.11 Document Template Instructions**

To properly identify your Word document as an IEEE 802.11 Submission

there are 5 steps that you must complete, and 8 data fields that you must fill in.

Step 1. Obtain a document number (has the form yy/xxxx).

Step 2. Title page (above) - fill in the document subject title text, full date (in the ISO 8601 format of YYYY-MM-DD), the full author(s) details and the abstract text (a total of 4 data fields).

Step 3. Menu select File, Properties. Fill in the 5 data fields:

Title field = document designator

example "doc.: IEEE 802.11-04/9876r0" , or

"doc.: IEEE 802.11-04/9876r2"

Author field = first author’s name

Keywords field = venue date (month year, e.g. January 2005)

Comments field = first author, affiliation

Step 4. Update the header and footer: To do this, menu select View, Normal, then menu select View, Page Layout (called View, Print Layout in some versions of Word). Switching the view of the document automatically updates all fields in the header and footer. Save the file with the final headers and footers. If Automatic updating is turn off you will need to manually open the header and footer view, select each field to be updated and press F9 (“Update”).

Step 5. Delete this page of instructions.

# MS Word Submission Preparation Summary:

Things to do: 5

Fields to fill in: 8

Recommendations:

a) Always create a new document using the template, rather than using someone else's document.

b) For quick and easy creation of new 802.11 submissions, place the 802.11 template files in the template folder area on your computer. Typical locations are:

c:\Program Files\Microsoft Office\Templates\802.11

or,

c:\Documents and Settings\User Name\Application Data\Microsoft\Templates\802.11

To create a new submission, menu select File, New, then select the appropriate 802.11 template file.

c) **When you update or revise your document**, remember to check all 5 fields in step 3 for the correct values and follow steps 3 through 4 again to ensure that the header and footer are updated with the revised field values.

rev: 2012-10-09 (Adrian Stephens)

blah

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Page.Line | Section | Comment | Suggestion | Resolution |
| 7249 | 2016.33 | 12.7.6.7 | "whether to install the temporal keys, the encapsulated GTK, and if management frame protection is negotiated, the IGTK" -- the GTK and the IGTK are TKs | Change to "and whether to install the temportal keys" | Reject: “w, x, y, and z” is better than “w, and x, y, and z” |

Discussion:

The text reads:

“The Authenticator sends an EAPOL-Key frame containing ANonce, the RSNE from its Beacon or

Probe Response frames, MIC, whether to install the temporal keys, the encapsulated GTK, and if

management frame protection is negotiated, the IGTK.”

That is kind of clunky because it’s a thing, a thing, direction, a thing, and a thing depending on an answer. But that’s better than saying “a thing, a thing, and direction, a thing, and a thing depending on an answer.”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Page.Line | Section | Comment | Suggestion | Resolution |
| 7318 | 1994.22 | 12.7.1.7.2 | "(Length+255)/256" potentially results in a non-integer number of iterations | Replace with [ Length / 256 ] where the [] should be the ceil operator symbols and the Length should remain italicised | Revise: make it Ceil(). |
| 7319 | 1986.47 | 12.7.1.2 | "(Len+159)/160" potentially results in a non-integer number of iterations | Replace with [ Len / 160 ] where the [] should be the ceil operator symbols and the Len should remain italicised | Revise: make it Ceil(). |
| 7335 | 1994.22 | 12.7.1.7.2 | "(Length+255)/256" assumes the output size of the has is 256 bits, but that is not the case for e.g. SHA-384 | At the end of line 13 above append "whose output length+F265 is <i>HashLen</i>" and then change "(Length+255)/256" to "(Length+HashLen-1)/HashLen", with all three variables italicised | Revise: make it Ceil() and have it depend on a variable that indicates the digest length of the underlying hash function. |

***Instruct the editor to modify section 12.7.1.2 as indicated:***

**12.7.1.2 PRF**

In the following, *K* is a key; *A* is a unique label for each different purpose of the PRF, treated as a sequence of ASCII-encoded octets without a terminating null; *B* is a variable-length string; *Y* is a single octet containing 0; *X* is a single octet containing the loop parameter *i*:

H-SHA-1(K , A , B , X )  HMAC-SHA-1(K , A || Y || B || X )

*PRF(K, A, B, Len)*

for i 🡨 0 to Ceil(*Len*/160) **do**

*R* 🡨 *R* || H-SHA-1(*K, A, B, i*)

return L(*R*, 0, *Len*)

***Instruct the editor to modify section 12.7.1.7.2 as indicated:***

**12.7.1.7.2 Key derivation function (KDF)**

The KDF for the FT key hierarchy, and for AKMs 00-0F-AC:11 and 00-0F-AC:12, is a variant of the pseudorandom function (PRF) defined in 12.7.1.2 (PRF) and is defined as follows:

Output  KDF-Hash-Length (K, Label, Context) where

Input: K , a derivation key whose length equals the block size of the hash function

Hash , a cryptographically strong hash function

Label , a string identifying the purpose of the keys derived using this KDF, treated as a

sequence of ASCII-encoded octets without a terminating null

Context , a bit string that provides context to identify the derived key

Length , the length of the derived key in bits

Output: a Length -bit derived key

*result* 🡨 ""

*iterations* 🡨 Ceil(*Length*/*dlen*)

**do** *i* = 1 to *iterations*

*result* 🡨*result* || HMAC-Hash(*K , i || Label || Context || Length* )

**od**

**return** first *Length* bits of *result*, and irretrievably delete the other bits

In this algorithm, *i* and *Length* are encoded as 16-bit unsigned integers, represented using the bit ordering conventions of 9.2.2 (Conventions); *K, Label,* and *Context* are bit strings and are represented using the ordering conventions of 9.2.2 (Conventions); and, *dlen* is the length, in bits, of the digest produced by HMAC-Hash.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Page.Line | Section | Comment | Suggestion | Resolution |
| 7339 | 1979.44 | 12.6.18 | "receives or invokes an MLME Disassociation or Deauthentication primitive" is too loose | Change to "receives an MLME-DEAUTHENTICATE.indication or MLME-DISASSOCIATE.indication primitive or issues an MLME-DEAUTHENTICATE.request or MLME-DISASSOCIATE.request primitive" | Accept |
| 7341 | 1979.55 | 12.6.18 | "receives or invokes an MLME Disassociation or Deauthentication primitive" is too loose | Change to "receives an MLME-DEAUTHENTICATE.indication or MLME-DISASSOCIATE.indication primitive or issues an MLME-DEAUTHENTICATE.request or MLME-DISASSOCIATE.request primitive" | Accept |

Discussion:

Text says:

When a non-AP STA’s SME receives a successful MLME-ASSOCIATE.confirm or MLMEREASSOCIATE. confirm primitive that is not part of a fast BSS transition or receives or invokes an MLME Disassociation or Deauthentication primitive, it deletes some security associations.”

Yes, agreed. Tighter is better.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Page.Line | Section | Comment | Suggestion | Resolution |
| 7571 | 1995.32 | 12.7.1.7.4 | "to generate a key whose length is equal to the length of the hash algorithm's digest" breaks the usual pattern and is too indirect | Add a new bullet "--- Length is the hash algorithm's output length" | Reject: length is not a stand-alone item in the key derivation statement |

Discussion:

Text says:

PMK-R1 = KDF-Hash-Length(PMK-R0, "FT-R1", R1KH-ID || S1KH-ID)

where

* KDF-Hash-Length is the KDF as defined in 12.7.1.7.2 (Key derivation function (KDF)) using the hash algorithm defined by the AKM in Table 9-132 (AKM suite selectors) to generate a key whose length is equal to the length of the hash algorithm’s digest.

So Length, just like Hash is not a standalone item, the standalone item is KDF-Hash-Length and the text describes what that item is.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Page.Line | Section | Comment | Suggestion | Resolution |
| 7710 | 1994.07 | 12.7.1.7.2 | “The KDF for the FT key hierarchy, and for the AKMs 00-0F-AC:11 and 00-0F-AC:12, is a variant of the pseudorandom function (PRF) defined in 12.7.1.2 (PRF)” – it’s also used for things other than FT and :11/12, no? | Change to "The  KDF  for  the  FT  key  hierarchy  and  for certain AKMs  is  a  variant  of  the pseudorandom function (PRF) defined in 12.7.1.2 (PRF)" | Revised: it’s used by a bunc of stuff so instead of enumerating them all let’s just call it the KDF. |

***Instruct the editor to modify section 12.1.7.2 as indicated:***

**12.1.7.2 Key derivation function (KDF)**

The KDF is a variant of the

pseudorandom function (PRF) defined in 12.7.1.2 (PRF) and is defined as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Page.Line | Section | Comment | Suggestion | Resolution |
| 7728 | 1962.43 | 12.6.1.1.6 | It says “Because the PTKSA is tied to the PMKSA or to a PMK-R1 security association, it only has the additional information from the 4-way handshake” – forgets FT | Add "or FT Protocol authentication" at the end | Agreed. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Page.Line | Section | Comment | Suggestion | Resolution |
| 7729 | 1963.44 | 12.6.1.1.8 | A GTKSA is created by the Supplicant’s SME when Message 3 of the 4-way handshake is received or when Message 1 of the group key handshake is received” – forgets FT | Add "or when FT Protocol authentication is used" at the end | Revised: since existing text is specific about the HS message, mention when in FT this happens |

***Instruct the editor to modify section 12.6.1.1.8 as indicated:***

**12.6.1.1.8 GTKSA**

The GTKSA results from a successful 4-way handshake, FT 4-way handshake, FT protocol, FT resource request protocol or the group key handshake and is unidirectional. In an infrastructure BSS, there is one GTKSA, used exclusively for encrypting group addressed MPDUs that are transmitted by the AP and for decrypting group addressed transmissions that are received by the STAs. In an IBSS or in a PBSS, each STA defines its own GTKSA, which is used to encrypt its group addressed transmissions, and stores a separate GTKSA for each peer STA so that encrypted group addressed traffic received from other STAs may be decrypted. A GTKSA is created by the Supplicant’s SME when Message 3 of the 4-way handshake is received, when Message 1 of the group key handshake is received, or when the Reassociation response of the FT handshake is received. The GTKSA is created by the Authenticator’s SME when the SME changes the GTK and has sent the GTK to all STAs with which it has a PTKSA. A GTKSA consists of the following elements:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Page.Line | Section | Comment | Suggestion | Resolution |
| 7730 | 1964.04 | 12.6.1.1.9` | "a valid Message 3 of the 4-way handshake or FT 4-way handshake, the Reassociation Response message of the fast BSS transition protocol with a status code indicating success, a Mesh Peering Open Message of the Authenticated Mesh Peering Exchange (AMPE) protocol, or a valid Message 1 of the group key handshake." -- the MPOM has to indicate success too | Add "with a status code indicating success" after "(AMPE) protocol" | Agree |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Page.Line | Section | Comment | Suggestion | Resolution |
| 7740 | 2065.50 | 12.10.2 | " a  256-bit  key" is duplication | Delete the cited text | Agree |

Discussion:

Text says:

“The PMK shall be a 256-bit key derived from keyseed using the key derivation function (KDF) from 12.7.1.7.2 (Key derivation function (KDF)) using the hash algorithm defined for the negotiated AKM in Table 9-132 (AKM suite selectors) according to Equation (12-4), and the PMKID shall be derived according to Equation (12-5).”

The length of the key can be determined by the KDF so length of the key can be determined elsewhere.

**References:**