IEEE P802.11
Wireless LANs

|  |
| --- |
| FILS AEAD and GCM clarifications |
| Date: 2015-05-12 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Jouni Malinen | Qualcomm |  |  | jouni@qca.qualcomm.com |
|  |  |  |  |  |

Abstract

This document discusses P802.11ai/D4.0 CID 7416 and a MDR comment on EAPOL-Key IV byte order and proposes changes to address these comments and number of related AEAD/GCM items that came up during the effort.

### The addressed comments

CID 7416; 11.11.2.6; 132,56

Comment:

"Processing of a received EAPOL-Key frame shall include verification that the received frame contains a counter that is strictly greater than the counter in the last received EAPOL-Key frame," -- and if this verification fails, then what?

Proposed Change:

Add a specification of the behaviour required if the verification fails

Proposed Resolution:

REVISED. Changes shown in 11-15/649r2.

MDR comment on EAPOL-Key IV byte order and bit description

REVISED. Changes shown in 11-15/649r2.

### Discussion

EAPOL-Key IV is actually big endian, not little endian: “The bit and octet convention for fields in the EAPOL-Key frame are defined in 11.2 of IEEE Std 802.1X-2010.” and that 802.1X-2010 clause has this text: “All EAPOL PDUs consist of an integral number of octets, numbered starting from 1 and increasing in the order that they are put into a MAC frame. The bits in each octet are numbered from 1 to 8, where 1 is the low-order bit. When consecutive octets are used to encode a binary number, the lower numbered octet contains the more significant bits of the binary number.” (i.e., 802.1X using big endian encoding by default unlike 802.11).

While going through the details for these comments, it became clear that number of details for AEAD/GCM use are missing from the current draft:

* GCM Authentication Tag length and location in the frame is not defined
* GCM Nonce construction details were not sufficiently described and the text seemed to imply a 13 octet nonce would be used which is not exactly common with AES-GCM (12 octets would be..)
* AEAD counter is described in unnecessarily complex and inflexible manner (e.g., GCM Nonce would likely be 12 octets, not 13 octets; some other algorithms may need different style); use all 16 octets of EAPOL-Key IV to as an unsigned integer to make this simpler

Proposed design at high level:

* use AEAD counter as a simple counter from 0 to 2^128-1 (with limits based on algorithm); do not split this to AP and non-AP STA ranges
* encode EAPOL-Key IV similarly to EAPOL-Key Replay Counter (unsigned integer in big endian encoding)
* enforce unique GCM nonces separately in nonce construction rules (use one octet to indicate sender)
* clarify encoding of AEAD counter and rules for verifying it is being incremented properly

Change history:

* Revision 1: Removed AAD definition for EAPOL-Key (it was already covered in the current draft in 11.6.2) and added “To guarantee uniqueness of GCM nonce values, the STA shall either deauthenticate or reassociate to derive a new PTKSA before the AEAD counter is incremented to 288.” to 11.11.2.6.
* Revision 2: 11.11.2.4.3 simplification, editorial change fix (“non-AP PTKSA” to “non-AP STA”). “AEAD encryption algorithm” to “AEAD algorithm” for consistency, 11.11.2.6: editorial fixes (frame name capitalization), grammar, and clarity.

### Proposed changes to the P802.11ai draft

**11.6.2 EAPOL-Key frames**

*Change the text as shown:*

**EAPOL-Key IV**. This field is 16 octets, represented as an unsigned binary number. It contains the IV used with the KEK. It shall contain 0 when an IV is not required. When AKM negotiated is not 00-0F-AC:14, 00-0F-AC:15,00-0F-AC:16, or 00-0F-AC:17, ~~I~~it ~~should be~~is initialized by taking the current value of the global key counter (see 11.6.11 (RSNA Authenticator key management state machine)) and then incrementing the counter. Note that only the lower 16 octets of the counter value are used. When the AKM negotiated is 00-0F-AC:14, 00-0F-AC:15, 00-0F-AC:16, or 00-0F-AC:17, the current value of the transmitter’s AEAD counter from the PTKSA is encoded in the field.

**11.11.2.4.3 PTK key derivation with FILS authentication**

*Change the text as shown:*

FILS uses two AEAD counters, one for the local STA and one for its peer. The STA shall set both counters to zero when creating a PTKSA.

**11.11.2.5.2 (Re)Association Request for FILS key confirmation**

*Change the text as shown:*

The (Re)Association Response frame shall be encrypted using the AEAD algorithm as defined in 11.11.2.6 (AEAD cipher mode for FILS) with the KEK as the key. The AAD used with the AEAD algorithm for the Association Request frame is constructed by concatenating the following data together in order:

…

*Note: This is for reverting post-D4.3 change in editor’s work document:*

The (Re)Association Request frame shall be encrypted

…

The plaintext passed to the AEAD algorithm is the data that would follow the FILS Session element in an unencrypted frame. The unique counter required by the AEAD algorithm shall be the current value of the AEAD counter for the non-AP STA. The ciphertext output by the AEAD algorithm concatenated with the Authentication Tag becomes the data that follows the FILS Session element in the encrypted and authenticated (Re)Association Request frame. The resulting (Re)Association Request frame shall be transmitted to the AP.

**11.11.2.5.3 (Re)Association response for FILS key confirmation**

*Change the text as shown:*

The plaintext passed to the AEAD algorithm is the data that would follow the FILS Session element in an unencrypted frame. The unique counter required by the AEAD algorithm shall be the current value of the AEAD counter for the AP. The ciphertext output by the AEAD algorithm concatenated with the Authentication Tag becomes the data that follows the FILS Session element in the encrypted and authenticated (Re)Association Response frame. The resulting (Re)Association Response frame shall be transmitted to the STA.

**11.11.2.6 AEAD cipher mode for FILS**

*Change the text as shown:*

FILS authentication uses an AEAD cipher mode to protect (Re)Association Request/Response and EAPOL-Key frames. The AEAD cipher mode is determined by the specific FILS AKM negotiated.

AES-GCM-128 is used when the AKM negotiated is 00-0F-AC:14 or 00-0F-AC:16 and AES-GCM-256 is used when the AKM negotiated is 00-0F-AC:15 or 00-0F-AC:17. AES-GCM-X (in Table 8-113) is GCM with X-bit AES key.

When the AEAD cipher mode used is GCM, the nonce, N, shall be 12 octets in length and shall be constructed as a concatenation of a one octet sender indication (0x00 = non-AP STA, 0x01 = AP) and the 11 least significant octets of the AEAD counter for the local STA (from the PTKSA) in big endian encoding. The AEAD counter is implicit in the (Re)Association Request and (Re)Association Response frames (0) and explicitly identified in the EAPOL-Key frames. Each successive invocation of the encryption operation of GCM shall increment the AEAD counter by 1. To guarantee uniqueness of GCM nonce values, the STA shall either deauthenticate or reassociate to derive a new PTKSA before the AEAD counter is incremented to 288.

 When processing a received EAPOL-Key frame, the STA shall verify that the received frame contains an AEAD counter that is strictly greater than the AEAD counter for the peer in the PTKSA. If the counter is not greater, the STA shall discard the received EAPOL-Key frame. Otherwise, the STA shall update the AEAD counter for the peer in the PTKSA to the value received in the EAPOL-Key frame.