IEEE P802.11
Wireless LANs

|  |
| --- |
| Proposed Resolution for Assigned Security CIDs – FILS Flexible AEAD Mode |
| Date: 2013-05-15 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| René Struik | Struik Security Consultancy | Toronto ON | +1 415 690-7363+1 647 867-5658Skype: rstruik | rstruik.ext@gmail.com |

Abstract

This document presents suggested text to implement “authenticated encryption” with TGai. Suggested text is relative to 802.11-2012 and TGai/D0.5 and takes into account the adopted resolution for fragmenting large objects (11/478r2).

***Modify section D0.5/11.11.2.4 as indicated:***

**11.11.2.4 Key confirmation with FILS authentication**

Key confirmation for FILS Authentication is an Associate Request followed by an Associate Response. The

Association Request and Association Response shall be protected using the KEK2 according to 11.11.2.5 and 11.11.2.6.

Upon the completion of key establishment (11.11.2.2) and key derivation (11.11.2.3) the STA shall construct an 802.11 associate request frame indicating its selected ciphersuite and the FILS AKM, and the FILS Key Confirmation element. The content of the Key Auth field of the Key Confirmation element depends on the type of FILS authentication.

The AP transfers any necessary KDEs to the STA in the Association Response frame. The AP may include one or more KDEs using the FILS KDE container. The format and the rules for transferring the KDE shall follow 11.6.2 (EAPOL Key Frames).

For FILS Authentication using a trusted third party, the Key Auth field of the Key Confirmation element of the Association Request shall be:

Key-Auth = HMAC-SHA256(KCK2, NSTA | NAP | STA-MAC | AP-BSSID).

For FILS Authentication without a trusted third party, the Key Auth field of the Key Confirmation element in the Association Request shall contain a digital signature using the STA's private key, the specific construction of the digital signature depends on the crypto-system of the public/private key pair:

Key-Auth = Sig-STA(gSTA | gAP | NSTA | NAP | STA-MAC | AP-BSSID).

Where Sig-STA indicates a digital signature using the STA's private key, gSTA is the octet-string representation

of the STA's public Diffie-Hellman value, gAP is the octet-string representation of the AP's public

Diffie-Hellman value, NSTA is the nonce selected by the STA, and NAP is the nonce selected by the AP.

The 802.11 Association Request frame shall be secured as follows:

* The input key shall be the KEK2
* The plaintext indicator shall be a set of information elements (including associated fragments, if applicable) contained in the Association Request frame that follow the FILS Session element. The procedure by which STA selects this set of information elements is outside scope of the specification.
* The first input string shall be set to the right-concatenation of the following fields:
* The STA MAC
* The AP BSSID
* The STA's nonce
* The AP's nonce
* The contents of the Association Request frame from the capability (inclusive) to the FILS Session element (inclusive)
* The second input string shall be set to the remainder of the frame body of the Association Request frame
* The input key, the plaintext indicator, and the first and second input string shall be passed to the encrypt-and-authenticate operation specified in 11.11.2.6.
* The frame body of the Association Request frame shall be set to the transformed second input string resulting from 11.11.2.6.

The resulting 802.11 Association Request frame shall be transmitted to the AP.

The received 802.11 Association Request frame shall be processed as follows:

* The input key shall be the KEK2
* The first input string shall be set to the right-concatenation of the following fields:
1. The STA MAC
2. The AP BSSID
3. The STA's nonce
4. The AP's nonce
5. The contents of the Association Request frame from the capability (inclusive) to the FILS Session element (inclusive)
* The second input string shall be set to the remainder of the frame body of the Association Request frame
* The input key and the first and second input strings shall be passed to the decrypt-and-verify operation specified in 11.11.2.7.
* If the output from 11.11.2.7 returns a failure or if the plaintext indicator does not confirm to the local policy of the AP, authentication shall be deemed a failure. Otherwise, the frame body of the Association Request frame shall be set to the transformed second input string resulting from 11.11.2.7.

The Key-Auth from the decrypted Association frame shall be checked. If it is incorrect, authentication shall be deemed a failure. If authentication is deemed a failure, the KCK2, KEK2, KCK, KEK, and TK shall be irretrievably destroyed. If authentication is not deemed a failure, the AP shall check the Key-Auth field in the Key Confirmation element.

For FILS Authentication using a trusted third party, the AP shall construct a verifier as follows:

Key-Auth' = HMAC-SHA256(KCK, NSTA | NAP | STA-MAC | AP-BSSID)

If Key-Auth' differs from the Key-Auth field in the Key Confirmation element, authentication shall be deemed a failure.

For FILS Authentication without a trusted third party, the AP shall use the STA's (certified) public key from the FILS Public Key element in the Association frame to verify the contents of the Key-Auth field of the Key Confirmation element. The specific technique for verification depends on the crypto-system used by the public key. If verification fails, authentication shall be deemed a failure.

If authentication is a failure, the KCK2, KEK2, KCK, KEK, and TK shall be irretrievably destroyed. Otherwise, the AP shall then construct an 802.11 associate response frame confirming the selected ciphersuite and the FILS AKM, and containing the FILS KDE Container, and its own Key-Auth.

For FILS authentication using a trusted third party, the Key Auth field of the Key Confirmation element in the Association Response shall be:

Key-Auth = HMAC-SHA256(KCK2, NAP | NSTA | AP-BSSID | STA-MAC).

For FILS Authentication without a trusted third party, the Key Auth field of the Key Confirmation element in the Association Response shall contain a digital signature using the AP's private key, the specific construction of the digital signature depends on the crypto-system of the public/private keypair:

Key-Auth = Sig-AP(gAP | gSTA | NAP | NSTA | AP-BSSID | STA-MAC ).

Where Sig-AP indicates a digital signature using the AP's private key, and where gSTA, gAP, NSTA, and NAP are the same as in the construction of the Association Request.

The 802.11 Association Response frame shall be protected as follows:

* The input key shall be the KEK2
* The plaintext indicator shall be a set of information elements (including associated fragments, if applicable) contained in the Association Request frame that follow the FILS Session element. The procedure by which AP selects this set of information elements is outside scope of the specification.
* The first input string shall be set to the right-concatenation of the following fields:
1. The AP BSSID
2. The STA MAC
3. The AP's nonce
4. The STA's nonce
5. The contents of the Association Response frame from the capability (inclusive) to the FILS

Session element (inclusive)

* The second input string shall be set to the remainder of the frame body of the Association Response frame
* The input key, the plaintext indicator, and the first and second input string shall be passed to the encrypt-and- authentication operation specified in 11.11.2.6
* The frame body of the Association Response frame shall be set to the transformed second input string resulting from 11.11.2.6.

The resulting 802.11 Association Response frame shall be transmitted to the STA.

The STA shall process the received 802.11 Association Response frame as follows:

* The input key shall be the KEK2
* The first input string shall be set to the right-concatenation of the following fields:
1. The AP BSSID
2. The STA MAC
3. The AP's nonce
4. The STA's nonce
5. The contents of the Association Response frame from the capability (inclusive) to the FILS Session element (inclusive)
* The second input string shall be set to the remainder of the frame body of the Association Response frame
* The input key and the first and second input string shall be passed to the decrypt-and-verify operation specified in 11.11.2.7.
* If the output from 11.11.2.7 returns failure or if the plaintext indicator does not confirm to the local policy of the STA, authentication shall be deemed a failure. Otherwise, the frame body of the Association Response frame shall be set to the transformed second input string resulting from 11.11.2.7.

The Key-Auth from the decrypted Authentication frame shall be checked. If it is incorrect, authentication shall be deemed a failure. If authentication is deemed a failure, the KCK2, KEK2, KCK, KEK, and TK shall be irretrievably destroyed. If authentication is not deemed a failure, the AP shall check the Key-Auth field in the Key Confirmation element.

For FILS Authentication using a trusted third party, the STA shall construct a verifier as follows:

Key-Auth' = HMAC-SHA256(KCK2, NAP | NSTA | AP-BSSID | STA-MAC).

If Key-Auth' differs from the Key-Auth field in the Key Confirmation element, authentication shall be deemed a failure .

For FILS Authentication without a trusted third party, the STA shall use the AP's (certified) public key from the FILS Public Key element in the Association frame to verify the contents of the Key-Auth field of the Key Confirmation element. The specific technique for verification depends on the crypto-system used by the public key. If verification fails, authentication shall be deemed a failure.

If authentication is a failure, the KCK2, KEK2, KCK, KEK, PMK, and TK shall be irretrievably destroyed.

Otherwise authentication succeeds. In that case, STA and AP shall irretrievably destroy the temporary keys

KCK2 and KEK2 and both shall use the TK with the cipher indicated by the negotiated. The KCK, KEK, and PMK shall be used for subsequent key management as specified in clause 11.5. The STA and AP shall set the lifetime of the PMKSA to the value dot11RSNAConfigPMKLifetime.

***Modify section D0.5/11.11.2.6 as indicated:***

**11.11.2.6 Encrypt and authenticate operation for FILS association frames**

**11.11.2.6.1 Input Transformation (Massage Plaintext if Applicable)**

* Determine the total octet length of the plaintext from the plaintext indicator and set the Encryption Length Indicator *LBL* to the 2-octet representation of this value.
* Partition the second input string into two string segments, viz. the plaintext *P* indicated by the plaintext indicator and the remainder *A* of the second input string.

NOTE: if one always encrypts the entire second input string, this transformation simply sets *P* to this string and *A* to the empty string. Similarly, if the to-be-authenticated string is always “at the back” of the second input string, the strings *P* and *A* are simply the left and right parts of the second input string, where the “dividing line” is determined by information on the length of plaintext. Thus, if one uses simple security policies as to which data elements are to be encrypted and authenticated and which only to be authenticated, one can implement this step in a trivial manner. A special case is where one encrypts the entire string, except possibly for a small portion at the end (e.g., the vendor-specific information).

**11.11.2.6.2 Encrypt and authenticate operation**

The AEAD scheme of 11.11.2.5 shall be used with the 802.11 Associate Request frame (for enciphering by

STA) or with the 802.11 Associate Response frame (for enciphering by AP), with the following instantiation:

* The key *K* shall be set to KEK2;
* The associated data string shall be set to the right-concatenation of the first input string and the string *A* determined in 11.11.2.6.1;
* The string *P* shall be set to the string *P* determined in 11.11.2.6.1;
* The nonce *N* shall be set to
1. For processing by STA: use the 13-octet all-zero string;
2. For processing by AP: use the 13-octet all-one string.

If the encryption-authentication process is unsuccessful, output a failure; otherwise, output the string *C* || *A*.

NOTE: if one always encrypts the entire second input string, the associated data string and plaintext string are always equal to the first and the second input string, respectively.

**11.11.2.6.3 Output Transformation (Indicate the Associated Data and Ciphertext)**

* Substitute the second input string by the string *LBL* || *C* || *A*. Here, *LBL* is the Encryption Length Indicator determined in 11.11.2.6.1.

***Modify section D0.5/11.11.2.7 as indicated:***

**11.11.2.7 Decrypt and verify operation for FILS association frames**

**11.11.2.7.1 Input Transformation (Determining the Associated Data and Ciphertext)**

* Set the Encryption Length Indicator *LBL* to the leftmost two octets of the second input string. If this is not possible, the procedure shall output a failure.
* Remove the Encryption Length Indicator *LBL* from the second input string.
* Partition the resulting second input string as *C* || *A*, where the octet string *C* has as length the value indicated by the Encryption Length Indicator *LBL*. If this partitioning is not possible, the procedure shall output a failure.

**11.11.2.7.2 Decrypt and Verify Operation**

The AEAD scheme of 11.11.2.5 shall be used with the 802.11 Associate Request frame (for deciphering by

STA) or with the 802.11 Associate Response frame (for deciphering by AP), with the following instantiation:

* The key *K* shall be set to KEK2;
* The associated data string shall be set to the right-concatenation of the first input string and the string *A* determined in 11.11.2.7.1;
* The string *C* shall be set to the string *C* determined in 11.11.2.7.1;
* The nonce *N* shall be set to
1. For processing by AP: use the 13-octet all-zero string.
2. For processing by STA: use the 13-octet all-one string;
* The function shall output the string *P* || *A* and shall set the plaintext indicator to *P* if the decryption/verification process is successful and shall output a failure otherwise.

**11.11.2.7.3 Output Transformation (Massage Plaintext if Applicable)**

* Parse the output string *P* || *A* and re-order IEs (including associated fragments, if applicable) that are out of order.
* Substitute the second input string by the result of this transformation (which contains all IEs in ascending order).

NOTE: If subsequent processing of unsecured incoming frames in an existing implementation does not assume ordering of IEs (i.e., it considers the frame body as a set of IEs, rather than an ordered sequence of IEs), this step does not need to be implemented (since the implementation does not care about ordering anyway). Similarly, if the originator of the AEAD-secured data always has the to-be-authenticated string “at the back” of the second input string, this step is not required (since the output string *P* || *A* then always is already in order). A special case is where one encrypts the entire string, except possibly for a small portion at the end (e.g., the vendor-specific information).