IEEE P802.11
Wireless LANs

|  |
| --- |
| FILS Authentication Protocol |
| Date: 2012-10-26 |
| 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  |
| George CherianJouni MalinenPhil Hawkes | Qualcomm | 5775 Morehouse Dr, San Diego, CA, USA | +1 858 651 6645 | gcherian@qualcomm.comjouni@qca.qualcomm.comphawkes@qualcomm.com |
| René Struik | Struik Security Consultancy | Toronto ON | +1 415 690-7363+1 647 867-5658Skype: rstruik | rstruik.ext@gmail.com |

Abstract

This document presents text that defines a FILS authentication protocol which satisfies all the relevant requirements in the SFD.

***Insert the following reference into 2:***

FIPS PUB 186-3 Digital Signature Algorithm (DSS)

IETF RFC 3279 – Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, April 2002

IETF RFC 5295, Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK), August 2008

IETF RFC 6696, EAP Extensions for EAP Re-authentication Protocol (ERP), July 2012

IETF RFC 5280 – Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008

RFC 5480 **-** ECC Subject Public Key Information, March 2009

RFC 6090 **-** Fundamental Elliptic Curve Cryptography Algorithms, February 2011

***Insert the following definition into 3.1:***

**3.1 Definitions**

**EAP Reauthentication Protocol (EAP-RP):** A protocol, using the EAP framework, allowing single-round-trip reauthentication with an Authentication Server following an initial EAP authentication.

**Trusted Third Party (TTP):** a non-STA entity that maintains a security association with both a non-AP STA and an AP.

**Perfect Forward Secrecy (PFS)**: a security property such that loss of secrecy of a long-lived secret does not compromise the security of past sessions.

**Certificate Authority (CA)**: entity that vouches for the binding between a device’s identity, its public key, and associated keying material (such as key validity period, key usage, etc.).

***Modify section 4.5.4.2 as indicated:***

* Authentication

IEEE 802.11 authentication operates at the link level between IEEE 802.11 STAs. IEEE Std 802.11 does not provide either end-to-end (message origin to message destination) or user-to-user authentication.

IEEE Std 802.11 attempts to control LAN access via the authentication service. IEEE 802.11 authentication is an SS. This service may be used by all STAs to establish their identity to STAs with which they communicate, in both ESS and IBSS networks. If a mutually acceptable level of authentication has not been established between two STAs, an association is not(#1421) established.

IEEE Std 802.11 defines five802.11(#12858) authentication methods: Open System authentication, Shared Key authentication, FT authentication(11r), simultaneous authentication of equals (SAE), and FILS authentication.(11s) Open System authentication admits any STA to the DS. Shared Key authentication relies on WEP to demonstrate knowledge of a WEP encryption key. FT authentication relies on keys derived during the initial mobility domain association to authenticate the (#1112)stations as defined in Clause 12 (Fast BSS transition).(11r) SAE authentication uses finite field cryptography to prove knowledge of a shared password.(11s) Three FILS methods are defined in this version of the specification: (1) the FILS authentication exchange using a TTP is performed without PFS, (2) the FILS authentication exchange using a TTP is performed with PFS, (3) The FILS authentication exchange without a TTP and with PFS (Refer to table 8.4.2.42b). When a trusted third party is used for FILS authentication, then EAP-RP as defined in [IETF RFC 5295/6696] shall be used. When a trusted third party is used for FILS authentication A STA that discovers a FILS-capable AP that claims a trusted relationship with a mutually-trusted third party it may begin the FILS Authentication protocol to the AP and perform mutual authentication using the trusted third party only if the STA and trusted third party already share a valid rRK, as defined in [IETF RFC 6696] (see section 11.9a.2.1); otherwise the STA may perform full EAP authentication via IEEE 802.1X authentication. The IEEE 802.11 authentication mechanism also allows definition of new authentication methods.

An RSNA might support SAE authentication and/or FILS authentication.(11s) An RSNA also supports authentication based on IEEE Std 802.1X-2004, or preshared keys (PSKs) after Open System authentication(11s). IEEE 802.1X authentication utilizes the EAP to authenticate STAs and the AS with one another. This standard does not specify an EAP method that is mandatory to implement. See 11.5.5 (RSNA policy selection in an IBSS and for DLS) for a description of the IEEE 802.1X authentication and PSK usage within an IEEE 802.11 IBSS.

In an RSNA, IEEE 802.1X Supplicants and Authenticators exchange protocol information via the IEEE 802.1X Uncontrolled Port. The IEEE 802.1X Controlled Port is blocked from passing general data traffic between two STAs until an IEEE 802.1X authentication procedure completes successfully over the IEEE 802.1X Uncontrolled Port.

Either SAE authentication, FILS authentication or(11s) the Open System 802.11 authentication algorithm is used in RSNs based on infrastructure BSS and IBSS, although Open System 802.11 authentication is optional in an RSN based on an IBSS. SAE authentication is used in an MBSS.(11s) RSNA disallows the use of Shared Key 802.11 authentication.(#12858)

***Modify section 4.5.4.3 as indicated:***(11s)

* Deauthentication

The deauthentication service is invoked when an existing Open System, Shared Key, SAE(11s) or FILS authentication is to be terminated. Deauthentication is an SS.

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.(11s)

In an ESS, because authentication is a prerequisite for association, the act of deauthentication causes(#1421) the STA to be disassociated. The deauthentication service may be invoked by either authenticated party (non-AP STA or AP). Deauthentication is not a request; it is a notification. The association at the transmitting STA is terminated when the STA sends a deauthentication notice to an associated STA. Deauthentication, and if associated, disassociation can not be refused by the receiving STA except when management frame protection(#12241) is negotiated and the message integrity check fails.(11w)

In an RSN ESS, Open System 802.11(#12858) authentication is required. In an RSN ESS, deauthentication results in termination of any association for the deauthenticated STA. It also results in the IEEE 802.1X Controlled Port for that STA being disabled and deletes the pairwise transient key security association (PTKSA). The deauthentication notification is provided to IEEE Std 802.1X-2004 via the MAC layer.

In an RSNA, deauthentication also destroys any related pairwise transient key security association(PTKSA)(11w), group temporal key security association (GTKSA), station-to-station link (STSL) master key security association (SMKSA), STSL transient key security association (STKSA), and integrity group temporal key security association (IGTKSA)(11w) that exist in the STA and, if applicable, closes the associated IEEE 802.1X Controlled Port. If pairwise master key (PMK) caching is not enabled, deauthentication also destroys the pairwise master key security association (PMKSA) from which the deleted PTKSA was derived.

In an RSN IBSS, Open System authentication is optional, but a STA is required to recognize Deauthentication frames. Deauthentication results in the IEEE 802.1X Controlled Port for that STA being disabled and deletes the PTKSA.

***Create section 4.10.3.4a***

**4.10.3.4a AKM operations using FILS authentication and a trusted third party**

It is assumed that the authenticator has a secure channel with the trusted third party in a manner outside the scope of this standard.

The following operations (see Figure <ANA-0>) are carried out when FILS authentication is used with a trusted third party:

1. The STA discovers the AP’s policy through passive monitoring of Beacon frames or through active probing. If a FILS-capable STA discovers that the AP supports FILS authentication and the identity of the trusted third party is known (and trusted) by the STA, the STA and AP proceed to FILS authentication
2. The STA initiates FILS authentication by sending an Authentication frame with the FILS information to the AP. The AP forwards the FILS Authentication information to the trusted 3rd party. Upon receiving a response from the trusted 3rd party, the AP responds to the STA with an Authentication frame with FILS information. The STA and AP generate a PMK as a result of this exchange. Exchange of messages (method, procedure, format and content) between AP/Authenticator and the trusted 3rd party is out of scope of this specification.
3. The STA sends an Association Request frame to the AP and receives a Association Response frame from the AP. This exchange provides proof-of-possession of the PMK and enables the creation of a PTKSA and further establishment of FILS state

IEEE 802.11 Probe Request ReRequest

STA/Supplicant

AP/Authenticator

Trusted 3rd Party

IEEE 802.11 Probe Response ReRequest

IEEE 802.11 Authentication

 FILS Authentication Request

 FILS Authentication Response

IEEE 802.11 Authentication

IEEE 802.11 Association Request

IEEE 802.11 Association Response

 **Figure <ANA-0>—FILS Authentication**

***Create section 4.10.3.4b***

**4.10.3.4b AKM operations using FILS authentication without an online trusted third party**

It is assumed that both STAs using FILS have obtained a public key certificate from a Certificate Authority and that each STA is capable of verifying this certificate during execution of the FILS authentication scheme. The manner by which these certificates are obtained is outside the scope of this standard.

The following operations are carried out when FILS authentication is used with a trusted third party:

1. The STA discovers the AP’s policy through passive monitoring of Beacon frames or through active probing. If a FILS-capable STA discovers that the AP supports FILS authentication and the identity of the trusted third party is known (and trusted) by the STA, the STA and AP proceed to FILS authentication
2. The STA initiates FILS authentication by sending a Authentication frame to the AP, after which the AP responds with a Authentication frame. The STA and AP generate a PMK as a result of this exchange.
3. The STA sends an Association Request frame to the AP and receives a Association Response frame from the AP. This exchange provides proof-of-possession of the PMK and enables the creation of a PTKSA and further establishment of FILS state.

***Modify section 6.3.5.2 as indicated:***

(11s)

**6.3.5.2 MLME-AUTHENTICATE.request**

**6.3.5.2.1 Function**

This primitive requests authentication with a specified peer MAC entity.

**6.3.5.2.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-AUTHENTICATE.request(

PeerSTAAddress,

AuthenticationType,

AuthenticateFailureTimeout,

Content of FT Authentication elements,

Content of SAE Authentication Frame,

FILS wrapped data,

VendorSpecificInfo

)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name**  | **Type**  | **Valid range**  | **Description** |
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which to perform the authentication process. |
| AuthenticationType | Enumeration | OPEN\_SYSTEM,SHARED\_KEY,FAST\_BSS\_TRANSITION,SAE,FILS | Specifies the type of authentication algorithm to use during the authentication process. |
| AuthenticationFailureTimeout | Integer | 1 | Specifies a time limit (in TU) after which the authentication procedure is terminated. |
| Content of FT Authentication elements | Sequence of elements | As defined in 12.8 | The set of elements to be included in the first message of the FT authentication sequence, as described in 12.8.2. Present only if dot11FastBSSTransitionActivated is true. |
| Content of SAE Authentication Frame | Sequence of elements and fields | As defined in 8.4.1.37, 8.4.1.38, 8.4.1.39, 8.4.1.40b, 8.4.1.41, and 8.4.1.42 | The set of elements and fields to be included in the SAE Commit Message or SAE Confirm Message. Present only if AuthenticationType indicates SAE authentication. |
| FILS wrapped data | Sequence of elements and fields | As defined in 8.4.1.42a | The FILS wrapped data field is used for the STA and AP to communicate data used by the FILS authentication algorithm |
| VendorSpecificInfo | A set of elements | As defined in 8.4.2.28 | Zero or more elements. |

**6.3.5.3.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-AUTHENTICATE.confirm(

PeerSTAAddress,

AuthenticationType,

ResultCode,

Content of FT Authentication elements,

Content of SAE Authentication Frame,

FILS wrapped data,

VendorSpecificInfo

)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name**  | **Type**  | **Valid range**  | **Description** |
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which to perform the authentication process. |
| AuthenticationType | Enumeration | OPEN\_SYSTEM,SHARED\_KEY,FAST\_BSS\_TRANSITION,SAE,FILS | Specifies the type of authentication algorithm to use during the authentication process. |
| ResultCode | Enumeration | SUCCESS, REFUSED,ANTI-CLOGGINGTOKEN REQUIRED,FINITE CYCLIC GROUPNOT SUPPORTED,AUTHENTICATIONREJECTED | Indicates the result of the MLMEAUTHENTICATE.request primitive. |
| Content of FT Authentication elements | Sequence of elements | As defined in 12.8 (FTauthentication sequence) | The set of elements included in the second message of the FT authentication sequence, as described in 12.8.3 (FT authentication sequence: contents of second message). Present only if dot11FastBSSTransitionActivated is true. |
| Content of SAE Authentication Frame | Sequence of elements and fields | As defined in 8.4.1.37(Send-Confirm field),8.4.1.38 (Anti-CloggingToken field), 8.4.1.39 (Scalarfield), 8.4.1.40 (Elementfield), 8.4.1.41 (Confirmfield), and 8.4.1.42 (FiniteCyclic Group field) | The set of elements and fields to be included in the SAE Commit Message or SAE Confirm Message. Present only if AuthenticationType indicates SAE authentication. |
| FILS wrapped data | Sequence of elements and fields | As defined in 8.4.1.42a | The FILS wrapped data field is used for the STA and AP to communicate data used by the FILS authentication algorithm |
| VendorSpecificInfo | A set of elements | As defined in 8.4.2.28 | Zero or more elements. |

**6.3.5.4.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-AUTHENTICATE.indication(

PeerSTAAddress,

AuthenticationType,

Content of FT Authentication elements,

Content of SAE Authentication Frame,

FILS wrapped data,

VendorSpecificInfo

)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name**  | **Type**  | **Valid range**  | **Description** |
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which the authentication relationship was established. |
| AuthenticationType | Enumeration | OPEN\_SYSTEM,SHARED\_KEY,FAST\_BSS\_TRANSITION,SAE,FILS | Specifies the type of authentication algorithm that was used during the authentication process. |
| Content of FT Authentication elements | Sequence of elements | As defined in 12.8 | The set of elements to be included in the first message of the FT authentication sequence, as described in 12.8.2. Present only if dot11FastBSSTransitionActivated is true. |
| Content of SAE Authentication Frame | Sequence of elements and fields | As defined in 8.4.1.37, 8.4.1.38, 8.4.1.39, 8.4.1.40, 8.4.1.41, and 8.4.1.42 | The set of elements to be included in the SAE Commit Message or SAE Confirm Message. Present only if AuthenticationType indicates SAE authentication. |
| FILS wrapped data | Sequence of elements and fields | As defined in 8.4.1.42a | The FILS wrapped data field is used for the STA and AP to communicate data used by the FILS authentication algorithm |
| VendorSpecificInfo | A set of elements | As defined in 8.4.2.28 | Zero or more elements. |

**6.3.5.5.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-AUTHENTICATE.response(

PeerSTAAddress,

ResultCode,

Content of FT Authentication elements,

Content of SAE Authentication Frame,

FILS wrapped data,

VendorSpecificInfo

)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name**  | **Type**  | **Valid range**  | **Description** |
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity from which the authentication request was received. |
| ResultCode | Enumeration | SUCCESS,REFUSED, ANTICLOGGINGTOKENREQUIRED,FINITE CYCLICGROUP NOT SUPPORTED,AUTHENTICATIONREJECTED | Indicates the result response to the authentication request from the peer MAC entity. |
| Content of FT Authentication elements | Sequence of elements | As defined in 12.8 | The set of elements to be included in the first message of the FT authentication sequence, as described in 12.8.2. Present only if dot11FastBSSTransitionActivated is true. |
| Content of SAE Authentication Frame | Sequence of elements and fields | As defined in 8.4.1.37, 8.4.1.38, 8.4.1.39, 8.4.1.40, 8.4.1.41, and 8.4.1.42 | The set of elements to be included in the SAE Commit Message or SAE Confirm Message. Present only if AuthenticationType indicates SAE authentication. |
| FILS wrapped data | Sequence of elements and fields | As defined in 8.4.1.42a | The FILS wrapped data field is used for the STA and AP to communicate data used by the FILS authentication algorithm |
| VendorSpecificInfo | A set of elements | As defined in 8.4.2.28 | Zero or more elements. |

***Modify table 8-22 in section 8.3.3.5 by inserting a new orders 8-10, incrementing the orders of subsequent rows:***

|  |
| --- |
| Table 8-22—Association Request frame body |
| Order | Information | Notes |
| 8 | FILS TAG | A field that contains an authentication tag used to secure FILS frames. |
| 9 | FILS session | The FS IE is an identifier for the FILS session  |
| 10 | FILS Key Confirmation | A field that performs a cryptographic proof of authentication for the FILS Authentication protocol. Present if FILS authentication is used. |
| Last | Vendor Specific | One or more vendor-specific (#1684)elements are optionally present(#29). These (#1684)elements follow all other (#1684)elements(#1221). |

***Modify table 8-23 in section 8.3.3.6 by inserting a new order 6-9, and incrementing the orders of subsequent rows:***

|  |
| --- |
| Table 8-23—Association Response frame body |
| Order | Information | Notes |
| 6 | FILS TAG | A field that contains an authentication tag used to secure FILS frames. |
| 7 | FILS session | The FS IE is an identifier for the FILS session  |
| 8 | FILS KDE Container | A field that contains the KDE information. |
| 9 | FILS Key Confirmation | A field that performs a cryptographic proof of authentication for the FILS Authentication protocol |
| Last | Vendor Specific | One or more vendor-specific (#1684)elements are optionally present(#29). These (#1684)elements follow all other (#1684)elements(#1221). |

***Modify section 8.3.3.11 as indicated:***

* Authentication frame format

The  frame  body of a management frame of subtype Authentication contains the information shown in Table 8-28 (Authentication frame body). (#29)FT authentication is used when FT support is advertised by the AP and dot11FastBSSTransitionActivated(#1005) is(#1217) true(#1535) in the (#1112)STA.(11r) SAE authentication is used when dot11MeshActiveAuthenticationProtocol is sae (1).(11s)      FILS authentication is used when support for FILS authentication is advertised by the AP and dot11FILSAuthenticationActivated is true in the STA.

|  |
| --- |
| Table 8-28-- Authentication frame body |
| Order | Information | Notes |
| 16 | FILS session | The FS IE is an identifier for the FILS session  |
| 17(11s) | FILS identity | The FI IE identity of a STA performing FILS authentication |
| 18 | FILS authentication type | The FA field is an indicator of the type of FILS authentication a particular session will perform |
| 19 | FILS nonce | The FN IE is a random, or pseudo-random, octet string used by the FILS authentication protocol. |
| 20 | FILS wrapped data | An encrypted and authenticated series of fields used for FILS authentication. |
| Last | Vendor Specific | One or more vendor-specific (#1684)elements are optionally present(#29). These (#1684)elements follow all other (#1684)elements(#1221). |

|  |
| --- |
| Table 8-29-- Presence of fields and(11s) elements in Authentication frames(11r)  |
| Authentication algorithm | Authentication transaction sequence no. | Status code | Presence of fields 4-15 (11r)(11s) |
| FILS(11s) | 1 | Status | FILS identity is presentFILS authentication type is present.FILS nonce is present. FILS wrapped data is present if FILS authentication uses a TTP. Finite cyclic group is present if FA type field indicates PFS.Element is present if FA type field indicates PFS. |
| FILS(11s) | 2 | Status | FILS identity is present if Status is zero.FILS authentication type is present if Status is zero.FILS nonce is present if Status is zero. FILS wrapped data is present if Status is zero and a TTP is used. Finite cyclic group is present if FA type field indicates PFS.Element is present if FA type field indicates PFS. |

***Modify section 8.4.1.1 as indicated:***

* Authentication Algorithm Number field

The Authentication Algorithm Number field indicates a single authentication algorithm. The length of the Authentication Algorithm Number field is 2 octets. The Authentication Algorithm Number field is illustrated in Figure 8-35 (Authentication Algorithm Number field). The following values are defined for authentication algorithm number:

Authentication algorithm number = 0: Open System

Authentication algorithm number = 1: Shared Key

Authentication algorithm number = 2: Fast BSS Transition(11r)

Authentication algorithm number = 3: simultaneous authentication of equals (SAE)

Authentication algorithm number = <ANA-1>: Fast Initial Link Setup authentication (11s)

Authentication algorithm number = 65 535: Vendor specific use

NOTE—The use of this value implies that a Vendor Specific element(Ed) is included with more information.(#10081)

All other values of authentication algorithm number are reserved.

***Create section 8.4.1.42a, 8.4.1.42b, and 8.4.1.42c***

**8.4.1.42a FILS wrapped data field**

The FILS wrapped data field is used for the STA and AP to communicate data used by the FILS authentication algorithm. See figure <ANA-2> FILS wrapped data.

|  |  |
| --- | --- |
|  | FILS wrapped data |
| Octets: | variable |
| * Figure <ANA-2> FILS-wrapped data(11s)
 |

**8.4.2.42b FILS authentication type field**

The FILS authentication type field is used for indicating the type of FILS authentication exchange, either with PFS or without PFS. The format of the FILS authentication field is shown in Figure <ANA-3>.

|  |  |
| --- | --- |
|  | FILS authentication type |
| Octets: | 1 |

**Figure <ANA-3>-- FILS authentication type format**

The value of the FILS authentication type is taken from table <ANA-4>.

|  |  |
| --- | --- |
|  Value |  Description |
|  0 | The FILS authentication exchange using a TTP is performed without PFS. |
|  1 | The FILS authentication exchange using a TTP is performed with PFS. |
|  2 | The FILS authentication exchange without a TTP and with PFS. |
|  3-255 | Reserved. |

|  |
| --- |
| **Table <ANA-4>-- Values of FILS authentication type(#1248)** |

**8.4.2.42c FILS nonce field**

The FILS nonce field is used for exchanging an additional source of randomness to the FILS authentication exchange. The nonce data shall be 16 octets and shall be chosen in a random manner. The format of the FILS nonce field is shown in Figure <ANA-5> FILS nonce field.

|  |  |
| --- | --- |
|  | FILS nonce |
| Octets: | 16 |

**Figure <ANA-5>-- FILS nonce field format**

***Create sections 8.4.2.121a, 8.4.2.121b, 8.4.2.121c, 8.2.4.121d, 8.2.4.121e, 8.2.4.121f as indicated:***

**8.4.2.121a FILS Identity element**

The FILS identity element is used for conveying an identity to use with the FILS authentication protocol (see 11.9a). The FILS identity element is included in Beacon and Probe Response frames by APs that support FILS authentication and is included in Authentication frames sent by STAs to initiate the FILS authentication protocol. The format of the FILS identity element is shown in Figure <ANA-6> FILS identity element.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Element ID | Length | ID type | FILS identity |
| Octets: | 1 | 1 | 1 | variable |
|  | **Figure <ANA-6>-- FILS identity element format(#1248)** |

The ID type subfield is set as follows:

* 0: Reserved
* 1: Trusted Third Party identity
* 2: STA identity

When using a trusted third party for authentication, the semantics of the FILS identity depend on the ID type as well as the namespace used by the Trusted Third Party to identify itself and entities with which it has a trusted relationship; they are therefore out of scope of this specification. When authenticating without a trusted third party, the ID type subfield shall be 2 (STA identity) for both the STA and AP, and the contents of the FILS identity field shall be an X.500 distinguished name (DN) that identifies either a certified or a raw public key.

**8.4.2.121b FILS Key Confirmation element**

The FILS Key Confirmation element is used to convey a cryptographic proof of authentication between a STA and an AP. The format of the FILS Key Confirmation element is shown in Figure <ANA-7> FILS Key Confirmation.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Element ID | Length | FILS Auth |
| Octets: | 1 | 1 | variable |
|  | **Figure <ANA-7>-- FILS Key Confirmation element format(#1248)** |

The FILS Auth field contains the cryptographic authentication information (see 11.9a.2.4).

**8.4.2.121c FILS KDE container element**

The FILS KDE container element is used to communicate one or more KDEs in a FILS authentication exchange. The FILS KDE container may contain one or more KDEs. The format of the FILS KDE container element is shown in Figure <ANA-8> FILS KDE container element.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | One or more entries |
|  | Element ID | Length | KDE(s) |
| Octets: | 1 | 1 | Variable |
| **Figure <ANA-8>-- FILS KDE container element format (#1248)** |

The Length field of the FILS KDE container specifies the total number of octets of all the KDEs. The encoding of the KDE field is defined in Table 11-6 (KDE) of 11.6.2 (EAPOL-Key frames) of this specification

**8.4.2.121d FILS TAG element**

The FILS TAG element is used to convey an authenticating tag which is used to protect FILS Association Request and Association Response frames. The format of the FILS TAG element is shown in Figure <ANA-9> FILS TAG.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Element ID | Length | TAG |
| Octets: | 1 | 1 | variable |
|  | **Figure <ANA-9>-- FILS TAG element format(#1248)** |

The TAG field contains the tag used by the encrypt-and-authenticate (see 11.9a.2.5) and decrypt-and-verify (see 11.9a.2.6) functions.

**8.4.2.121e FILS session element**

The FILS session element is used for conveying the (unique) identifier of an in-progress FILS authentication protocol. The session identifier is chosen randomly by the non-AP STA in the FILS authentication protocol. The format of the FILS session element is shown in Figure <ANA-10> FILS session element.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Element ID | Length | FILS session |
| Octets: | 1 | 1 | 8 |
| **Figure <ANA-10>-- FILS session element format(#1248)** |

**8.4.2.121f FILS public key element**

The FILS public key element is used to communicate the device’s (certified) public-key for use with the FILS authentication exchange. The format of the FILS certificate element is shown in Figure <ANA-11> FILS certificate element.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Element ID | Length |  Key Type |  FILS public key |
| Octets: | 1 | 1 | 1 | variable |
|  | **Figure <ANA-11>-- FILS public key element format(#1248)** |

Where the Key Type subfield is as follows:

* 0: Reserved
* 1: An X.509v3 certificate encoded according to RFC 5280.
* 2: A raw public key encoded according to RFC 5480
* 3: A raw public key encoded according to RFC 3279

***Modify section 8.4.2.27.3 as indicated:***

* AKM suites

The AKM Suite Count field indicates the number of AKM suite selectors that are contained in the AKM Suite List field.

The AKM Suite List field contains a series of AKM suite selectors contained in the RSN (#1684)element. In an IBSS(#13085) only a single AKM suite selector may be specified because STAs in an IBSS (#10287)use the same AKM suite and because there is no mechanism to negotiate the AKMP in an IBSS (see 11.5.5).

Each AKM suite selector specifies an AKMP. Table 8-101 gives the AKM suite selectors defined by this -standard. An AKM suite selector has the format shown in Figure 8-187.(#11242)

|  |
| --- |
| * Table 8-101-- AKM suite selectors
 |
| OUI | Suite type | Meaning |
| Authentication type | Key management type | Key derivation type (11w) |
| 00-0F-AC | <ANA-12> | FILS | FILS key management as defined in 11.9a  | Defined in 11.9.a |
| 00-0F-AC | <ANA-12 > +1 –255  | Reserved | Reserved | Reserved |
| Vendor OUI | Any | Vendor-specific | Vendor-specific | Vendor-specific |
| Other | Any | Reserved | Reserved | Reserved |

***Modify section 10.3.2.2 as indicated:***

* + - 1. Authentication—originating STA

Upon receipt of an MLME-AUTHENTICATE.request primitive, the originating STA(#3097) shall authenticate with the indicated STA using the following procedure:(11r)

* If the STA is in an IBSS the SME shall delete any PTKSA and temporal keys held for communication with the indicated(#11069) STA by using the MLME-DELETEKEYS.request primitive (see 11.5.12 (RSNA security association termination)).(#10600)
* (#1342)The STA(#10600) shall execute one of the following:(11r)
* For the Open System or Shared Key authentication algorithm, the authentication mechanism described in 11.2.3.2 (Open System authentication) or 11.2.3.3 (Shared Key authentication), respectively.(11r)
* For the FT authentication algorithm in an ESS, the authentication mechanism described in 12.5 (FT Protocol), or, if resource requests are included, 12.6 (FT Resource Request Protocol).(#10600)(11r)
* For SAE authentication in an ESS, IBSS, or MBSS, the authentication mechanism described in 11.3 (Authentication using a password).(11s)

4) For FILS authentication in an ESS, the authentication mechanism described in 11.9a (FILS Authentication).

* If the authentication was successful within the AuthenticateFailureTimeout(#1342), the state(#1342) for the indicated STA shall be set to State 2 if it was State 1; the state shall remain unchanged if it(Ed) was other than State 1.(#10600)
* The MLME(#1342) shall issue an MLME-AUTHENTICATE.confirm primitive to inform the SME of the result of the authentication.

***Modify section 10.3.2.3 as indicated:***

* Authentication—destination STA

Upon receipt of an Authentication frame with authentication transaction sequence number equal to 1, the destination STA(#3097) shall authenticate with the originating(#1342) STA using the following procedure:

1. If FILS authentication is being used in an ESS, the MLME shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request, including the FILS authentication element, and the SME shall execute the procedure described in 11.9a (Authentication for fast link setup)

***Modify section 11.5.1.1.1 and 11.5.1.1.2 as indicated:***

* Security association definitions
* General(#2119)

IEEE Std 802.11 uses the notion of a security association to describe secure operation. Secure communications are possible only within the context of a security association, as this is the context providing the state—cryptographic keys, counters, sequence spaces, etc.—needed for correct operation of the IEEE 802.11 cipher suites.

A security association is a set of policy(ies) and key(s) used to protect information. The information in the security association is stored by each party of the security association, needs to(#10380) be consistent among all parties, and needs to(#10380) have an identity. The identity is a compact name of the key and other bits of security association information to fit into a table index or an MPDU. The following types of security associations are supported by an RSN STA(11w):

* PMKSA: A result of a successful IEEE 802.lX exchange, SAE authentication, FILS authenticaiton,(11s) preshared PMK information, or PMK cached via some other mechanism.
* PMKSA

When the PMKSA is the result of a successful IEEE 802.1X authentication, it is derived from the EAP authentication and authorization parameters provided by the AS. When the PMKSA is the result of a successful SAE authentication, it is generated as a result of the successful completion of the SAE exchange.(11s) When the PMKSA is the result of a successful FILS authentication, it is generated as a result of the successful completion of the FILS authentication protocol. This security association is bidirectional. In other words, both parties use the information in the security association for both sending and receiving. The PMKSA is created by the Supplicant’s SME when the EAP authentication, or FILS authentication completes successfully or the PSK is configured. The PMKSA is created by the Authenticator’s SME when the PMK is created from the keying information transferred from the AS, when IEEE 802.1X authentication is utilized, or when the SAE exchange or FILS authentication exchange successfully completes(11s) or the PSK is configured. The PMKSA is used to create the PTKSA. PMKSAs are cached for up to their lifetimes. The PMKSA consists of the following elements:

***Modify section 11.5.1.3.2 as indicated:***

* Security association in an ESS

In an ESS there are two cases:

* Initial contact between the STA and the ESS
* Roaming by the STA within the ESS

A STA and AP establish an initial security association via the following steps:

* The STA selects an authorized ESS by selecting among APs that advertise an appropriate SSID and FILS capabilities.
* The STA then performs(11s) IEEE 802.11(11s) authentication followed by association to the chosen AP. Confirmation(11s) of security parameters takes place during association. A STA performing IEEE 802.1X authentication uses Open System authentication. A STA performing secure password-based, or PSK, authentication uses SAE authentication.(11s) A STA performing authentication for fast initial link set-up performs FILS authentication.

NOTE 1—It is possible for more than one PMKSA to exist. As an example, a second PMKSA might(#10381) come into existence through PMKSA caching. A STA might leave the ESS and flush its cache. Before its PMKSA expires in the AP’s cache, the STA returns to the ESS and establishes a second PMKSA from the AP’s perspective.

NOTE 2—An attack altering the security parameters is(#10369) detected by the key derivation procedure.

NOTE 3—IEEE 802.11 Open System authentication provides no security, but is included to maintain backward compatibility with the IEEE 802.11 state machine (see 10.3 (STA authentication and association)).

* SAE authentication and FILS authentication provide mutual authentication and derivation of a PMK. If Open System authentication is chosen instead,(11s) the (#3098)Authenticator or the (#3098)Supplicant initiates IEEE 802.1X authentication. The EAP method used by IEEE Std 802.1X-2004(#10369) needs to support mutual authentication, as the STA needs assurance that the AP is a legitimate AP.

NOTE 1—Prior to the completion of IEEE 802.1X authentication and the installation of keys, the IEEE 802.1X Controlled Port in the AP blocks(#10369) all data frames. The IEEE 802.1X Controlled Port returns to the unauthorized state and blocks all data frames before invocation of an MLME-DELETEKEYS.request primitive. The IEEE 802.1X Uncontrolled Port allows IEEE 802.1X frames to pass between the Supplicant and Authenticator. Although IEEE Std 802.1X-2004 does not require a Supplicant Controlled Port, this standard assumes that the Supplicant has a Controlled Port in order to provide the needed level of security. Supplicants without a Controlled Port compromise RSN security and are not(#10382) used.

NOTE 2—Any secure network cannot support promiscuous association, e.g., an unsecured operation of IEEE Std 802.11. A trust relationship is needed(#10383) between the STA and the AS of the targeted SSID prior to association and secure operation, in order for the association to be trustworthy. The reason is that an attacker can deploy a rogue AP just as easily as a legitimate network provider can deploy a legitimate AP, so some sort of prior relationship is necessary to establish credentials between the ESS and the STA.

* The last step is key management. The authentication process, whether SAE authentication or FILS authentication utilizing IEEE 802.11 authentication frames or IEEE 802.1X authentication utilizing data frames post association, creates cryptographic keys shared between the cryptographic endpoints—the AP and STA,(11s) or the IEEE 802.1X AS and the STA, when using SAE/FILS or IEEE 802.1X, respectively. When using IEEE 802.1X(11s) the AS transfers these keys to the AP, and the AP and STA uses one of the key confirmation handshakes, e.g., the 4-Way Handshake or FT 4-Way Handshake,(#1038) to complete security association establishment. When using SAE authentication there is no AS and therefore no key transfer; the 4-way Handshake is performed directly between the AP and STA.(11s) The key confirmation handshake indicates when the link has been secured by the keys and is ready to allow normal data traffic and protected (#13074)robust management frames(11w). FILS authentication performs key confirmation as part of the exchange and no additional handshake is necessary.

When FT is not enabled, a STA roaming within an ESS establishes a new PMKSA by one of the four(11s) schemes:(#1039)

* In the case of (re)association followed by IEEE 802.1X or PSK authentication, the STA repeats the same actions as for an initial contact association, but its Supplicant also deletes the PTKSA when it roams from the old AP. The (#3098)Supplicant also deletes the PTKSA when it disassociates/deauthenticates from all BSSIDs in the ESS.
* In the case of SAE authentication followed by (re)association, the STA repeats the same actions as for initial contact association, but the non-AP STA also deletes the PTKSA when it roams from the old AP. Note that a STA can take advantage of the fact that it can perform SAE authentication to multiple APs while maintaining a single association with one AP, and then use any of the PMKSAs created during authentication to effect a fast BSS transition.(11s)
* In the case of FILS authentication, the STA repeats the same actions as for initial contact and authentication. Note that a STA can take advantage of the fact that it can initiate FILS authentication to multiple APs while maintaining a single association with one AP, and finalize the FILS authentication with one AP.

***Modify section 11.5.9.1 as indicated:***

11.5.9 RSNA authentication in an ESS

11.5.9.1 General(#28)

When establishing an RSNA in a non-FT environment or during an FT initial mobility domain association,(#1040) a STA shall use IEEE 802.11 SAE authentication, FILS authentication or(11s) Open System authentication prior to -(re)association.

SAE authentication is initiated when a STA’s MLME-SCAN.confirm primitive finds another AP within the current ESS that advertises support for SAE in its RSN element.

FILS authentication is initiated when a STA’s MLME-SCAN.confirm primitive finds an AP that advertises support for FILS in its RSN element and indicates support for a trusted third party known to the STA.(11s)

IEEE 802.1X authentication is initiated by any one of the following mechanisms:

* If a STA negotiates to use IEEE 802.1X authentication during (re)association, the STA’s management entity may(#12694) respond to the MLME-ASSOCIATE.confirm (or indication) primitive by requesting the (#3098)Supplicant (or (#3098)Authenticator) to initiate IEEE 802.1X authentication. Thus, in this case, authentication is driven by the STA’s decision to associate and the AP’s decision to accept the association.
* If a STA’s MLME-SCAN.confirm primitive finds another AP within the current ESS, a STA may signal its Supplicant to use IEEE Std 802.1X-2004 to preauthenticate with that AP.

NOTE—A roaming STA’s IEEE 802.1X Supplicant can(#1520) initiate preauthentication by sending an EAPOL-Start message via its old AP, through the DS, to a new AP.

* If a STA receives an IEEE 802.1X message, it delivers this to its Supplicant or Authenticator, which may initiate a new IEEE 802.1X authentication.

***Modify section 11.5.12 as indicated:***

* + 1. RSNA key management in an ESS

When the IEEE 802.1X authentication completes successfully, this standard assumes that the STA’s IEEE 802.1X Supplicant and the IEEE 802.1X AS (#10369)share a secret, called a PMK. In a non-FT environment, the(#1042) AS transfers the PMK, within the MSK, to the AP, using a technique that is outside the scope of this standard; the derivation of the PMK from the MSK is EAP-method-specific. With the PMK in place, the AP initiates a key -confirmation handshake with the STA. The key confirmation handshake sets the IEEE 802.1X state variable portValid (as described in IEEE Std 802.1X-2004) to TRUE.

When SAE authentication completes, both STAs share a PMK. With this PMK in place, the AP initiates the key confirmation handshake with the STA.

Key confirmation is part of the FILS authentication exchange and no further handshakes are needed to satisfy key management requirements in an ESS.(11s)

When FILS authentication is not used, the key confirmation handshake is implemented by the 4-Way Handshake. The purposes of the 4-Way Handshake are as follows:

* Confirm the existence of the PMK at the peer.
* Ensure that the security association keys are fresh.
* Synchronize the installation of temporal keys into the MAC.
* Transfer the GTK from the Authenticator to the Supplicant.
* Confirm the selection of cipher suites.

NOTE 1—It is possible to forge message 1 of the 4-Way Handshake.(#12703) However, the forgery attempt is(#10369) detected in the -failure of the 4-Way Handshake.

NOTE 2—Neither the AP nor the STA can use the PMK for any purpose but the one specified herein without compromising the key. If the AP uses it for another purpose, then the STA can masquerade as the AP; similarly if the STA reuses the PMK in another context, then the AP can masquerade as the STA.

***Modify Table 11-9 in section 11.6.3 as indicated:***

**11.6.3 EAPOL-Key frame construction and processing**

|  |  |  |  |
| --- | --- | --- | --- |
|  AKM |  Integrity Algorithm |  Size of MIC | Key Wrap Algorithm |
| 00-0F-AC:6 | AES-128-CMAC |  16 |  AES Key Wrap |
| <ANA-12> |  TBD |  TBD |  TBD |

***Create section 11.9a and its component subsections***

**11.9a Authentication for Fast Initial Link Set-up**

STAs, both AP STAs and non-AP STAs, who share a means of authentication—either mutual trust of a third party, or based on trust in the other party’s public keys—may use that shared trust to mutually authenticate and derive a shared key in a more efficient manner than using IEEE 802.1X.

The FILS Authentication protocol authenticates STAs to each other, optionally using a TTP. The authentication exchange can optionally be performed with PFS. When a TTP is not used, PFS shall be used. When the FILS authentication protocol is performed with PFS, the STA and AP derive ephemeral public and private keys with respect to a particular set of domain parameters that define a finite cyclic group and then exchange public keys. When a TTP is not used, PFS shall be used. In either case, the result of the FILS Authentication protocol is a PTKSA. FILS Authentication is an RSNA authentication protocol.

**11.9a.1 Assumptions on FILS Authentication**

The security of FILS authentication depends on the following assumptions:

* Communication between the STAs and the trusted third party, when applicable, is protected with a secure deterministic authenticated encryption function.
* When using a TTP, each STA shares a symmetric key (or keys) with the trusted third party that is (are) capable of being used with ERP; when not using a TTP, each STA shall have a means to trust the public key of the other STA.
* When PFS is used, a finite cyclic group is negotiated for which solving the discrete logarithm problem is computationally infeasible.
* When PFS is used, both the STA and AP have at least one finite cyclic group from the dot11RSNAConfigDLCGroupTable in common.

All FILS Association frames shall be encrypted and authenticated (see 11.9a.2.5 and 11.9a.2.6).

**11.9a.2 FILS Authentication protocol**

The STA and the AP communicate using Authentication frames to perform key establishment and 802.11 association frames to perform key confirmation.

After exchanging Authentication frames, the STA and AP derive a shared and secret key which will be used to derive a set of secret keys that are authenticated after exchanging 802.11 association frames.

When a trusted third party is used for FILS authentication, then EAP-RP as defined in [IETF RFC 5295/6696] shall be used.

**11.9a.2.1 Discovery with FILS Authentication**

An AP indicates that it is capable of performing FILS Authentication by constructing a FILS-capable Beacon or Probe response. FILS-capable 802.11 Beacons or Probe responses shall contain an AKM indicating support for FILS Authentication as well as FILS Identity IEs indicating the identity of the AP and, when applicable, the identity(-ies) of the trusted third party(-ies) with whom the AP maintains a relationship.

A STA that discovers a FILS-capable AP that claims a trusted relationship with a mutually-trusted third party may begin the FILS Authentication protocol to the AP and perform mutual authentication using the trusted third party only if the STA and trusted third party already share a valid rRK, as defined in [IETF RFC 6696]. A STA that discovers a FILS-capable AP that advertises an identity for which the STA has a trusted public key may begin the FILS Authentication protocol to the AP and perform mutual authentication using trusted public keys.

**11.9a.2.2 Key Establishment with FILS Authentication**

A FILS-capable STA and AP establish a shared key by exchanging Authentication frames. The specific contents of the Authentication frame depend on the particular authentication technique—whether a TTP is being used or whether digital signatures are being used—and whether PFS is obtained in the exchange or not.

**11.9a.2.2.1 FILS Key Establishment with a Trusted Third Party**

When using a trusted third party, the STA first chooses a random 16 octet nonce, and constructs an EAP-Initiate/Re-auth packet as specified in [IETF RFC6696], with the following additional clarification:

* Regarding ERP Flags
	+ The ‘B’ flag shall be set to 0, indicating that this is not an ERP bootstrap message.
	+ The ‘L’ flag shall be set to 1, indicating that the trusted third party is to provide the lifetimes of rRK and rMSK in the EAP-Finish/Re-auth Packet.
* The “Cryptosuite” field shall not be set to 1.

If PFS is desired, the STA selects a finite cyclic group from the dot11RSNAConfigDLGGroupTable, generates an ephemeral secret private key, and performs the group’s scalar-op (see 11.3.4.1) with its random ephemeral private key and the generator from the selected finite cyclic group to compute an ephemeral public key.

The STA then then constructs an Authentication frame with the Authentication algorithm number set to <ANA-1> and the Authentication transaction sequence number set to one (1). The STA’s FILS Identity shall be indicated using the FILS Identity IE (see 8.4.2.121a), the random nonce shall be encoded as the FILS nonce field (see 8.4.2.121c), the FILS authentication type shall be set to indicate the specific type of FILS authentication, and the EAP-Initiate/Re-auth packet shall be encoded as FILS Wrapped Data (see 8.4.1.42a). If PFS is desired, the chosen finite cyclic group shall be encoded in the Finite Cyclic Group field (see 8.4.1.42) and the ephemeral public key shall be encoded into the Element field (see 8.4.1.40) according to the element to octet-string conversion in 11.3.7.2.4.

The STA shall transmit the Authentication frame to the AP.

If Authentication frame includes a Finite Cyclic Group field, then the AP shall first determine whether the indicated finite cyclic group in the received FILS authentication frame is supported. If not, it shall respond with an Authentication frame with the Authentication algorithm number set to <ANA-1> and the Status set to 77 (Authentication is rejected because the offered finite cyclic group is not supported) and shall terminate the exchange. If the group is supported or if PFS is not being used in this exchange, the AP shall forward the STA’s EAP-Initiate/Re-auth packet to the TTP. When applicable, the AP communicates with the trusted third party using the same protocols with which it uses when authenticating with EAP. Suitable protocols include, but are not limited to, remote authentication dial-in user service (RADIUS) (IETF RFC 2863-2000) and Diameter (IETF RFC 3588-2003).

If PFS is being used, the AP shall also generate an ephemeral private key and perform the group’s scalar-op (see 11.3.4.1) to produce its own ephemeral public key. The AP may delay the generation of its ephemeral public/private key pair until after receiving a response from the TTP.

The TTP processes the EAP-Initiate/Re-auth packet as specified in RFC6696 and returns an EAP-Finish/Re-auth packet to the AP. In the case of successful authentication by the TTP, the TTP returns the associated EAP-RP rMSK with the EAP-Finish/Re-auth packet.

If the TTP responds with an failure indication, then the AP shall produce an Authentication frame with the Authentication algorithm number set to <ANA-1> and the Status set to 15 (Authentication rejected because of challenge failure). If the TTP responds with an success indication (including the associated EAP-RP rMSK), then the AP shall generate its own nonce and construct an Authentication frame for the STA. This frame shall contain the EAP-Finish/Re-auth packet received from the TTP, and, if PFS is being used, the AP’s ephemeral public key. The AP shall set the Authentication sequence number to (2) and shall transmit the frame to the STA.

If PSF is being used for the exchange, then the AP shall perform the group’s scalar-op (see 11.3.4.1) with the STA’s ephemeral public key and its own ephemeral private key to produce an ephemeral Diffie-Hellman shared secret, *ss*.

Upon transmission of the FILS Authentication response, the AP shall perform key derivation per section 11.9a.2.3.

The STA processes the received Authentication frame.

1. If the received Authentication frame does not include the Authentication algorithm number set to <ANA-1>, or if the received Authentication frame does not include a EAP-Finish/Re-auth packet, then the STA shall abandon the FILS authentication
2. If the received Authentication frame includes the Status set to 15 (Authentication rejected because of challenge failure), then the STA shall abandon the FILS authentication
3. The STA ensures that the AP transmitted PFS parameters consistent with the desire of the STA (indicated by whether or not the STA transmitted an ephemeral public key.
	1. If the STA transmitted an ephemeral public key, and the received Authentication frame does not include a well-encoded ephemeral public key, then the STA shall abandon the FILS authentication.
	2. If the STA did not transmit an ephemeral public key desired PFS, and the received Authentication frame includes an ephemeral public key, then the STA shall abandon the FILS authentication.
4. The STA processes the EAP-Finish/Re-auth packet as per RFC6696 -
	1. If the ‘R’ flag = 0, indicating success, then the STA shall generate rMSK.
	2. If the ‘R’ flag = 1, indicating failure, then the STA shall abandon the FILS authentication..
5. If PSF is being used for the exchange, then the STA shall perform the group’s scalar-op (see 11.3.4.1) with the AP’s ephemeral public key and its own ephemeral private key to produce an ephemeral Diffie-Hellman shared secret, *ss*.
6. The STA shall perform key derivation per section 11.9a.2.3.

**11.9a.2.2.2 FILS Key Establishment without a Trusted Third Party**

When not using a trusted third party, the non-AP STA begins FILS Key Establishment by first selecting a finite cyclic group from the dot11RSNConfigDLCGroup table. It then chooses a random, ephemeral private key, uses the selected group’s scalar-op (see 11.3.4.1) with its private key to generate its ephemeral public key, and chooses a random nonce.

The STA then constructs an 802.11 authentication frame with the Authentication algorithm number set to <ANA-1> and the Authentication transaction sequence number set to one (1). The STA’s FILS Identity shall be indicated using the FILS Identity IE (see 8.4.2.121a), the random nonce shall be encoded as the FILS nonce field (see 8.4.2.121c), the FILS authentication type shall be set to indicate FILS authentication without a trusted third party (2), the chosen finite cyclic group shall be encoded in the Finite Cyclic Group field (see 8.4.1.42), and the STA’s public key shall be encoded into the Element field (see 8.4.1.40) according to the element to octet-string conversion in 11.3.7.2.4.

The STA shall transmit the 802.11 authentication frame to the AP.

The AP processes the STA’s 802.11 authentication frame. First, if the finite cyclic group indicated by the Finite Cyclic Group field is not acceptable, the AP shall respond with an 802.11 authentication frame with the status code of 77 (“Authentication is rejected because the offered finite cyclic group is not supported”) and terminate the FILS authentication protocol. If the finite cyclic group is acceptable, the AP shall verify the validity of the STA’s public key.

First, the public key shall be converted from an octet string to an element according to the conversion in 11.3.7.2.5. Then the public key, as a group element, shall be verified in a group-specific fashion according to SP 800-56a section 5.6.2.3.

The AP then shall choose a random nonce, and random, ephemeral private key, and then use the agreed-upon group’s scalar-op (see 11.3.4.1) with its private key to generate its ephemeral public key. The AP then constructs an 802.11 authentication frame with the Authentication algorithm number set to <ANA-1>, the Authentication transaction sequence number set to two (2), and the FILS authentication type to indicate FILS authentication without a trusted third party (2). The AP’s identity shall be indicated using the FILS Identity IE (see 8.4.2.121a), its random nonce shall be encoded into the FILS nonce field (see 8.4.2.121c), the finite cyclic group shall be encoded in the Finite Cyclic Group field (see 8.4.1.42), and the AP’s public key shall be encoded into the Element field (see 8.4.1.40) according to the element to octet-string conversion in 11.3.7.2.4.

The AP shall transmit the 802.11 authentication frame to the STA. The AP may choose to derive the Diffie-Hellman shared secret, *ss*, at this point or it may choose to delay those computations until Key Confirmation (see 11.9a.2.4). If it chooses to derive *ss* at this point, the AP shall use the STA’s ephemeral public key and its private key with the chosen group’s scalar-op to derive *ss*, and the AP shall then perform Key Derivation (see 11.9a.2.3).

The STA processes the AP’s 802.11 authentication frame. First in ensures that the finite cyclic group in the AP’s response matches the group selected by the STA. If they differ, the STA shall terminate the authentication exchange. If they match, the STA shall verify the validity of the AP’s public key.

First, the public key shall be converted from an octet string to an element according to the conversion in 11.3.7.2.5. Then the public key, as a group element, shall be verified in a group-specific fashion according to SP 800-56a section 5.6.2.3. If public key validation fails the STA shall terminate the authentication exchange. Otherwise it computes the Diffie-Hellman shared secret, ss, by using the AP’s ephemeral public key and its private key with the chosen group’s scalar-op to derive ss. The STA then performs Key Derivation (see 11.9a.2.3) and begins Key Confirmation (see 11.9a.2.4).

**11.9a.2.3 Key Derivation with FILS Authentication**

Key derivation with FILS Authentication uses the KDF from section 11.6.1.7.2 to produce three keys, a key encryption key (KEK), a confirmation key (KCK), and a traffic key (TK). The inputs to the KDF are the two 16 octet nonces produced by the STA and AP, a constant label, the ERP secret result if a TTP is being used, and, the Diffie-Hellman shared secret, *ss*, if PFS is being used. The length of the KEK and KMK shall be 128 bits, and the KCK shall be 256 bits, and therefore the output from the KDF shall be 512+TK\_bits, where TK\_bits is determined from table 11-4.

KEK | KMK | KCK | TK = KDF-X(NSTA | NAP, “FILS KECK PTK Derivation”, [rMSK][ | *ss*])

Where X is 512+TK\_bits from table 11-4, rMSK is the output of the ERP exchange if a trusted third party was used, and *ss* is the shared secret resulting from the Diffie-Hellman exchange if PFS was used. The KEK and KMK shall be used with the encrypt-and-authenticate (see 11.9a.2.5) and decrypt-and-verify (see 11.9a.2.6) functions.

**11.9a.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 KEK and KMK according to section 11.9a.2.5 and 11.9a.2.6.

Upon the completion of key establishment (11.9a.2.2) and key derivation (11.9a.2.3) the STA shall construct a nascent 802.11 associate request frame indicating its selected ciphersuite and the FILS AKM, and the FILS Key Confirmation element. The FILS TAG field shall be set to zero. 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 section 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(KCK, 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 keys shall be the KEK and KMK
* The input plaintext shall be the contents of the Association Request frame that follow the FILS TAG element
* The input AAD shall be:
	+ 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 TAG element (exclusive)
* The input keys, the plaintext, and the AAD shall be passed to the encrypt-and-authenticate operation specified in 11.9a.2.5.
* The output authenticating tag from 11.9a.2.5 shall be copied into the TAG field of the FILS TAG element
* The output ciphertext from 11.9a.2.5 shall become the remainder of the Association Request frame that follows the FILS TAG element.

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 keys shall be the KEK and KMK
* The input tag shall be taken from the TAG field of the FILS TAG element
* The input ciphertext shall be the contents of the Association Request frame that follow the FILS TAG element
* The input AAD shall be:
	+ 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 SIV element (exclusive)
* The input keys, the TAG, the ciphertext, and the AAD shall be passed to the decrypt-and-verify operation specified in 11.9a.2.6.

If the output from 11.9a.2.6 returns a failure, authentication shall be deemed a failure. If the output returns plaintext, 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 KEK, KMK, KCK, PMK, and all shared secrets 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 Public Key IE 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 KEK, KMK, KCK, PMK, and all shared secrets shall be irretrievably destroyed. Otherwise, the AP shall then construct a nascent 802.11 associate response frame confirming the selected ciphersuite and the FILS AKM, and containing the FILS KDE Container, and its own Key-Auth. The FILS TAG element shall be set to zero.

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(KCK, 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 keys shall be the KEK and KMK
* The input plaintext shall be the contents of the Association Request frame that follow the FILS TAG element
* The input AAD shall be:
	+ 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 TAG element (exclusive)
* The input keys, the plaintext, and the AAD shall be passed to the encrypt-and-authentication operation specified in 11.9a.2.5.
* The output TAG shall be copied into the TAG field of the FILS TAG element
* The output ciphertext shall become the remainder of the Association Response frame that follows the FILS TAG element.

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

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

* The input keys shall be the KEK and KMK
* The tag shall be taken from the TAG field of the FILS TAG element
* The input ciphertext shall be the contents of the Association Response frame that follow the FILS TAG element
* The input AAD shall be:
	+ 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 TAG element (exclusive)
* The input keys, the tag, the ciphertext, and the AAD shall be passed to the decrypt-and-verify operation specified in 11.9a.2.6.

If the output from 11.9a.2.6 returns failure, authentication shall be deemed a failure. If the output returns plaintext, 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 KEK, KMK, KCK, PMK, and all shared secrets 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(KCK, 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 Public Key IE 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 KEK, KMK, KCK, PMK, and all shared secrets shall be irretrievably destroyed. Otherwise authentication succeeds and both the STA and AP shall use the TK generated in 11.9a.2.3 with the cipher indicated by the ciphersuite in the Association Request and Association Response.

**11.9a.2.5 Encrypt and Authenticate operation for FILS Association frames**

The specific operation for authenticated encryption is TBD but the interface shall be:

* + The function shall take an encryption key, an authentication key, AAD, and plaintext
* The function shall perform authenticated encryption on the plaintext and shall authenticate, but not encrypt, the AAD
* The function shall output ciphertext and an authenticating tag.

**11.9a.2.6 Decrypt and Verify operation for FILS Association frames**

The specific operation for verified decryption is TBD but the interface shall be:

* The function shall take an encryption key, an authentication key, AAD, a tag, and ciphertext
* The function shall perform verified decryption on the ciphertext and shall verify the integrity of the AAD
* The function shall output plaintext if both the plaintext and AAD is verified and otherwise shall output a failure

**References:**

**Motion-1:** To authorize the Editor to incorporate the text changes proposed in contribution 11-12-1045r5 (*11-12-1045-05-00ai-fils-auth-protocol*) to the draft TGai Draft Specification Document.

Yes: \_\_\_\_\_\_\_\_\_\_\_\_; No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; Abstain: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_