IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Certificate-Based Authentication Protocol for IEEE 802.11 TGai | | | | |
| Date: August 31, 2012 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| René Struik | Struik Security Consultancy | Toronto ON | USA: +1 (415) 690-7363  Toronto: +1 (647) 867-5658  Skype: rstruik | rstruik.ext@gmail.com |

Abstract

This document presents suggested text to define how to implement FILS authentication using a certificate-based authenticated key agreement scheme. Suggested text is relative to 802.11REVmb\_D12.0.

NOTE:

The scheme proposed complies with adopted motions within 802.11ai to-date.

ACKNOWLEDGEMENT:

Thanks to Dan Harkins for gracefully suggesting me to reuse his original 11/1488r0 submission, so as to provide textual changes similar in style as his.

REVISION NOTES:

* R03 adds note as to compliance with TGai motions so far (no update to 802.11-2011 yet [still pending – will end up with R04, once I have Word version hereof]).
* R02 cleans up some change markers, so as to improve readability (no other changes)
* R01 adds a note (NOTE-2) to Section 11.9a.2.2, so as to tackle the scenario where STA and AP do have certificates, but at least one of these cannot verify the other’s certificate (this is public-key authentication scheme (d) in 11/1408r05). NOTE: The scenario where they can verify each other’s certificate is public-key authentication scheme (a) in 11/1408r05.

***Insert the following references into 2:***

**RFC 5480 - ECC Subject Public Key Information, replaces RFC 3279 (March 2009).**

**RFC 6090 - Fundamental Elliptic Curve Cryptography Algorithms (February 2011)**

**FIPS Pub 186-2, DSS - Digital Signature Standard (with change notice), October 5, 2001.**

**FIPS 180-3, Secure Hash Standard, October 2008.**

***Insert the following informative references into Annex A (Bibliography):***

R.L. Rivest, “Can We Eliminate Certificate Revocation Lists?,” in *Financial Cryptography - FC’98*,

R. Hirschfeld, Ed., Lecture Notes in Computer Science, Vol. 1465, pp. 178-183, Springer, 1998.

P. McDaniel, A. Rubin, “A Response to “Can We Eliminate Certificate Revocation Lists?”,” in

*Financial Cryptography - FC 2000*, Y. Frankel, Ed., Lecture Notes in Computer Science, Vol. 1962,

pp. 245-258, Springer, 2001.

P. Gutmann, “PKI: It’s Not Dead, Just Resting,” IEEE Computer, Vol. 35, No. 8, pp. 41-48, 2002.

P. Gutmann, “Everything You Never Wanted to Know About PKI, but Were Forced to Find Out,”

University of Auckland, 2004. http://www.cs.auckland.ac.nz/~pgut001/pubs/pkitutorial.pdf

S. Micali, “Efficient Certificate Revocation,” Technical Report TM-542b, MIT Laboratory for

Computer Science, March 22, 1996.

D. Hein, J. Wolkerstorfer, N. Felber, “ECC Is Ready for RFID – A Proof in Silicon,” in *SAC 2008*, R.

Avanzi, L. Keliher, F. Sica, Eds., Lecture Notes in Computer Science, Vol. 5381, pp. 401-413, 2009.

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

**3.1 Definitions**

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) FILS authentication allows two STAs to authenticate each other without requiring the online involvement of a trusted third party (although the latter may assist in authorization). The IEEE 802.11 authentication mechanism also allows definition of new authentication methods.

An RSNA might support SAE authentication and 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 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 theidentity 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 FILS authentication request to the AP, after which the AP responds with a FILS authentication response. The STA and AP generate a PMK as a result of this exchange.
3. The STA sends a FILS association request to the AP and receives a FILS association response 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 Request

IEEE 802.11 Authentication Response

IEEE 802.11 Association Request

IEEE 802.11 Association Response

**Figure 4-TBD—Public-Key Based FILS Authentication with Certificates**

***Modify section 8.3.3.5 as indicated:***

**8.3.3.5 Association Request frame format**

The frame body of a management frame of subtype Association Request contains the information shown in

Table 8-22.

|  |  |  |
| --- | --- | --- |
| Table 8-22—Association Request frame body | | |
| Order | Information | Notes |
| <ANA-a>(11s) | FILS MIC | A string indicating a message authentication tag used during key confirmation of FILS authentication. |
| <ANA-b> | FILS signature | An octet string indicating a signature used during key confirmation and device authentication of FILS authentication. |
| <ANA-2> | FILS session | The FS IE is an identifier for the FILS session |
| 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 8.3.3.5 as indicated:***

**8.3.3.6 Association Response frame format**

The frame body of a management frame of subtype Association Response contains the information shown in

Table 8-23.

|  |  |  |
| --- | --- | --- |
| Table 8-23—Association Response frame body | | |
| Order | Information | Notes |
| <ANA-a>(11s) | FILS MIC | A string indicating a message authentication tag used during key confirmation of FILS authentication. |
| <ANA-b> | FILS signature | An octet string indicating a signature used during key confirmation and device authentication of FILS authentication. |
| <ANA-2> | FILS session | The FS IE is an identifier for the FILS session |
| 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 |
| <ANA-1>(11s) | FILS identity | The FI IE identity of a STA performing FILS authentication |
| <ANA-2> | FILS session | The FS IE is an identifier for the FILS session |
| <ANA-3> | FILS certificate | The device certificate used by 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). |

|  |  |  |  |
| --- | --- | --- | --- |
| 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 present.  FILS session is present.  FILS ephemeral public key is present.  FILS certificate is present.  Finite cyclic group is present. |
| FILS(11s) | 2 | Status | FILS session is present if Status is zero.  FILS ephemeral public key is present if Status is zero.  FILS certificate is present if Status is zero. Finite cyclic group is present. |

***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-5>: 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***

**8.4.1.42a FILS certificate field**

The FILS certificate field is used by the STA and AP to communicate their respective public-key certificate used by the FILS authentication algorithm. See figure <ANA-1> FILS certificate.

|  |  |
| --- | --- |
|  | FILS certificate |
| Octets: | variable |
| * Figure <ANA-1> FILS- (11s)certificate | |

***Create section 8.4.2.121a, 8.4.2.121b, and 8.4.2.121c as indicated:***

**8.4.2.121a FILS Identity element**

The FILS identity element is used for conveying the identity to be used with the FILS authentication protocol (see 11.9a). The FILS identity element is included in Beacons and Probe responses by APs that support FILS authentication and is included in 802.11 authentication requests by STAs to initiate the FILS authentication protocol. The format of the FILS identity element is shown in Figure <ANA-2> FILS identity element.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Element ID | | Length | ID type | FILS identity |
| Octets: | 1 | | 1 | 1 | variable |
|  | | **Figure <ANA-2>-- FILS identity element format(#1248)** | | | |

The ID type subfield is set as follows:

* 0: Reserved
* 1: Certificate Authority identity
* 2: STA identity

The semantics of the FILS identity depend on the ID type as well as the namespace used by the Certificate Authority to identify itself and entities for which it certified public keys. The STA identity is equal to the device’s MAC address. The Certificate Authority’s identity is out of scope of this specification.

**8.4.2.121b 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-3> FILS session element.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Element ID | Length | FILS session |
| Octets: | 1 | 1 | 8 |
| **Figure <ANA-3>-- FILS session element format(#1248)** | | | |

**8.4.2.121c FILS certificate element**

The FILS certificate element is used to communicate the device’s public-key certificate with the FILS authentication exchange. The nonce data shall have variable length and shall be compliant with the PKIX format specified in RFC 5480.. The format of the FILS nonce element is shown in Figure <ANA-4> FILS nonce element.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Element ID | Length | FILS certificate |
| Octets: | 1 | 1 | variable |
| **Figure <ANA-3>-- FILS nonce element format(#1248)** | | | |

***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-6> | FILS | FILS key management as defined in 11.9a | Defined in 11.9.a |
| 00-0F-AC | <ANA-6>+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:***

* 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 or IBSS, 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 or IBSS, 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 authenticaiton 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 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) An STA performing authentication for fast initial link set-up uses 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. An 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, thus obviating the need for an additional handshake.

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:***

* + 1. 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 possession of a public key certificate issued by a Certificate Authorityknown 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.

***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 have public keys certified by entities that are known and trusted by the other STA may mutually authenticate and derive a PMK in a more efficient manner than using IEEE 802.1X.

The FILS Authentication protocol executes an ephemeral Diffie-Hellman key agreement scheme, where the Diffie-Hellmann exponents are signed by each communicating party, thus obviating the need for online involvement of a third party. The STA and AP each derive ephemeral public and private keys with respect to a particular set of domain parameters that define a finite cyclic group and then exchange the resulting ephemeral public keys. The result of the FILS Authentication protocol is a PMKSA. FILS Authentication is an RSNA authentication protocol.

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

The security of FILS authentication depends on the following assumptions:

* Key confirmation between the STAs is protected with a secure authenticated encryption function.
* Each STA has a device certificate that is issued by a Certificate Authority that is trusted by the other STA..
* A finite cyclic group is negotiated for which solving the discrete logarithm problem is computationally infeasible.

For interoperability reasons, we will assume the following:

* Both the STA and AP have at least one finite cyclic group from the dot11RSNAConfigDLCGroupTable in common. This cyclic group shall correspond to the P-256 curve as specified by NIST [FIPS Pub 186-2].
* Both the STA and AP shall support ECDSA certificates defined over this P-256 curve, with uncompressed points, and the SHA-256 hash function.
* Preferably, certificates should be short-lived, thereby obviating the need for CRLs, OCSP, etc, and not contain cross-certification chains. For a motivation, please see [Rivest, Gutmann, Rubin, Micali].

**11.9a.2 FILS Authentication protocol**

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

After exchanging 802.11 authentication frames, the STA and AP derive a shared secret key, which will then be used to derive a PMK that is authenticated via the exchange of 802.11 association frames. This provides for mutual key confirmation between both parties. Mutual device authentication is realized via the exchange by either party of a digital signature over the communicated ephemeral keys and verification hereof by the other party.

**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 suite element indicating support for FILS Authentication as well as FILS Identity IEs indicating the identity of the AP.

An STA that discovers a FILS-capable AP may begin the FILS Authentication protocol to the AP.

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

A FILS-capable STA and AP establish a shared key by exchanging 802.11 authentication frames. The STA first generates a random public-private key pair corresponding to the finite cyclic group from the dot11RSNAConfigDLCGroupTable with which to perform the exchange.

It then constructs an 802.11 authentication frame with the Authentication algorithm number set to <ANA-5> and the Authentication transaction sequence number set to one (1). The chosen finite cyclic group shall be encoded in the Finite Cyclic Group field (see 8.4.1.42), the STA’s FILS Identity shall be indicated using the FILS Identity IE (see 8.4.2.121a), the ephemeral public key shall be indicated using the element field (see 8.4.1.40), the device’s public key certificate shall be indicated using the FILS Certificate Element (see 8.4.2.121c), and the FILS session identifier shall be indicated using the FILS session element (see 8.3.3.11).

The STA shall transmit this message as the 802.11 authentication request frame to the AP.

Upon receipt of the 802.11 authentication request frame, the AP determines whether the indicated finite cyclic group is supported. If not, it shall respond with an 802.11 authentication frame with the Authentication algorithm number set to <ANA-5> and the Status set to FINITE\_CYCLIC\_GROUP\_NOT\_SUPPORTED and shall terminate the exchange.

The AP may decide to reject the authentication request based on criteria that are outside the scope of the standard. If so, it shall generate an 802.11 authentication response frame with the Authentication algorithm number set to <ANA-5>, the Authentication transaction sequence number set to two (2), and the Status set to AUTHENTICATION\_REJECTED. The AP shall transmit this frame to the STA and terminate the exchange.

Otherwise, the AP shall generate an ephemeral public-private key pair corresponding to the same finie cyclic group. On its own turn, it shall construct an 802.11 authentication response frame similar in format to that just received, but now including its own FILS identity and its own ephemeral public key and device certificate. It shall then transmit this message as the authentication response frame to the STA.

NOTE 1– Upon receipt of the authentication request frame from the STA, the AP may exchange information with a third device, e.g., so as to assist in authorization decisions regarding admission of STA to the network. These communications, however, are outside scope of the FILS authentication protocol and the standard, since involving authorization, rather than authentication, messaging. Similarly, any state updates by the AP that solely depend on return messaging by such a third device are outside scope of the FILS protocol and the standard. This is motivated by the observation that, from the STA’s perspective, the mechanism by which the AP arrives at authorization decisions is a unknown (i.e., it has no way of verifying whether these took place using localized knowledge only or would also involve intelligence as part of the network infrastructure). As a final note, authorization decisions as to which services a device may perform on the network may very well depend on details of higher-layer protocols that cannot be vetted at the the network level.

NOTE 2 – Upon receipt of the authentication request frame from the STA, the AP may determine that it cannot verify the certificate of STA, since it was signed by a Certificate Authority that is unknown to AP. In that case, AP may exchange information with a third device, so as to get this third device to vouch for the authenticity of STA’s public key or so as to get a “compatible” certificate itself. In this scenario, involvement of the third party would assist in successful authentication. Even so, whether or not the AP involves a third party in this certificate translation or in certificate verification is unknown to STA.

The AP shall then compute the Diffie-Hellman key by performing a scalar multiplication on the STA’s ephemeral public key using as scalar its own private key and then execute the KDF function with the resulting shared secret, the STA’s and the AP’s FILS identifier to produce the FILS Authentication keys (see 11.9a.2.3).

The STA shall check that the session identifier and selected group in the received 802.11 authentication response frame match those it sent to the AP. Moreover, it shall check that the FILS identifier of the AP corresponds to the AP it sent the authentication request to. If there is a mismatch, the STA shall drop the frame and terminate the protocol. Otherwise, the AP shall then compute the Diffie-Hellman key by performing a scalar multiplication on the AP’s ephemeral public key using as scalar its own private key and then execute the KDF function with the resulting shared secret, the STA’s and the AP’s FILS identifier to produce the FILS Authentication keys (see 11.9a.2.3).

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

Key derviation with FILS Authentication uses the KDF from section 11.6.1.7.2 to produce two keys, a key confirmation key (KCK) used for key confirmation, and a pairwise master key (PMK). The inputs to the KDF are the FILS Identifiers of the STA and AP, a constant label, and the x-coordinate of the shared secret they produced as a result of the Diffie-Hellman key exchange. The KCK shall be 256 bits (32 octets) and the PMK shall be 256 bits (32 octets).

KCK | PMK = KDF-512(FILS ID STA | FILS ID AP, “FILS KCK PMK Derivation”, *ss*)

Where *ss* is the x-coordinate of the shared secret (in affine representation) resulting from the FILS Authentication.

**11.9a.2.4 Key Confirmation with FILS Authentication**

Upon the completion of key establishment (11.9a.2.2) and key derivation (11.9a.2.3) the STA shall construct an 802.11 associate request frame, with FILS session identifier Sid set to the value used during key establishment (see 11.9a.2.2), where the MIC element shall be constructed as follows:

MIC-data-STA = HMAC-SHA256(KCK, FILS-Sid | FILS-ID STA | FILS-ID-AP),

and where the signature element Sign-STA shall be constructed using the public key contained in STA’s device certificate as follows:

Sign-STA = ECDSA-SHA-256(Eph.PublicKey-AP | Eph.PublicKey-STA | FILS-ID AP | FILS-ID STA | FILS-Sid).The STA shall transmit the 802.11 Association Request frame to the AP.

The AP shall verify the correctness of the received MIC-data from the 802.11 Associate Request frame. Subsequently, it shall verify STA’s certificate and the signature Sign-STA from the 802.11 Associate Request frame, including policy checks (as in RFC5480) and checking that the certificate’s Subject field of the public key corresponds to the STA. If any of these verifications fail, FILS Authentication shall fail and the KCK, PMK and shared secret shall be irretrievably destroyed. Otherwise, the AP shall construct an 802.11 Associate Response frame similar in format to that just received, but now with the role of STA and AP reversed and using his own KCK key and public key to construct the authentication tag MIC-Data-AP, respectively the signature element Sign-AP. Thus, theMIC-data shall be constructed as follows:

MIC-data-AP = HMAC-SHA256(KCK, FILS-Sid | FILS-ID AP | FILS-ID STA),

and the signature element Sign-AP shall be constructed using the public key contained in AP’s device certificate as follows:

Sign-AP = ECDSA-SHA-256(Eph.PublicKey-STA | Eph.PublicKey-AP | FILS-ID STA | FILS-ID AP | FILS-Sid).

The AP shall transmit the 802.11 Association Response frame to the STA.

The STA shall verify the correctness of the received MIC-data from the 802.11 Associate Response frame. Subsequently, it shall verify AP’s certificate and the signature Sign-AP from the 802.11 Associate Response frame, including policy checks (as in RFC5480) and checking that the certificate’s Subject field of the public key corresponds to the AP. If any of these verifications fail, FILS Authentication shall fail and the KCK, PMK, and shared secret shall be irretrievably destroyed.

Both the STA and AP shall generate a PTK as follows:

PTK = KDF-Len(PMK, “FILS PTK Derivation”, Sid | min(FILS-ID STA, FILS-ID AP) |

max(FILS-ID STA, FILS-ID AP).

Where Len is taken from table 11-4 for the selected pairwise ciphersuite and min and max are as defined in 11.6.1.3 (Pairwise key hierarchy).

Both the STA and AP shall irretrievably destroy their ephemeral private key used during the execution of the FILS authentication protocol and shall similarly destroy the Diffie-Hellman key, the PMK key, and KCK key.