Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**Enhancement of 802.16e to Support EAP-based Authentication / Key Distribution Rev. 3**
Date Submitted	**2004-01-15**
Source(s)	Jeff Mandin Streetwaves Networking Amatzia 5 Jerusalem, Israel Voice: 972-50-724-587 Fax: 972-50-724-587 mailto:jeff@streetwaves-networks.com
Re:	Call for contributions to 802.16e security adhoc (11/17/2003)
Abstract	Description of requirements, mechanism, and security considerations for EAP in 802.16e
Purpose	Update IEEE C802.16-71/r2
Notice	This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.
Patent Policy and Procedures	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:chair@wirelessman.org> as early as possible, in written or electronic form, if patented technology (or technology under patent application) might be incorporated into a draft standard being developed within the IEEE 802.16 Working Group. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>.

Enhancement of 802.16e to Support EAP-based Authentication / Key Distribution

Jeff Mandin

on behalf of the Security Ad Hoc group (Seok Heon Cho, Jeff Mandin, Ae Soon Park, Chulsik Yoon) and incorporating comments from Vladimir Yanover.

1 Scope of this document

This document outlines how to incorporate Extensible Authentication [2] and compatible key management into 802.16e.

For the purposes of this document, the actual EAP authentication exchange and "inner method" can be regarded as a "black box".

2 Background

2.1 Motivations

To enable mobile operators to use other forms of credentials in addition to, or instead of, the current PKI-based device certificates. Examples include:

Credentials based on EAP-TLS as are widely used in 802.11 environments.

various forms of provider-supplied credentials to be installed in an off-the-shelf SS device eg. Subscriber Information Module (SIM) card using a mechanism such as EAP-SIM or EAP-AKA. These schemes have been embraced in 3G environments.

facilitation of handover to other media (ie. 802.11) by providing a shared set of credentials, as well as hooks for preauthentication or other functions

2.2 Requirements

Interoperability with non-EAP enabled 802.16d/e systems

Maintenance of the current functional definition of the PKM module and support for all higher-level services currently provided

Support for 802.16 primary, static, and dynamic security associations. These include SAs for both unicast and non-unicast MAC-layer connections.

Provision for ciphersuite selection and reauthentication

Compatibility with standard EAP methods. Similarity to 802.1x is desirable, as is reuse of work from 802.1x.

Appropriate compliance with security recommendations for EAP as outlined in RFC 2254bis section 7.

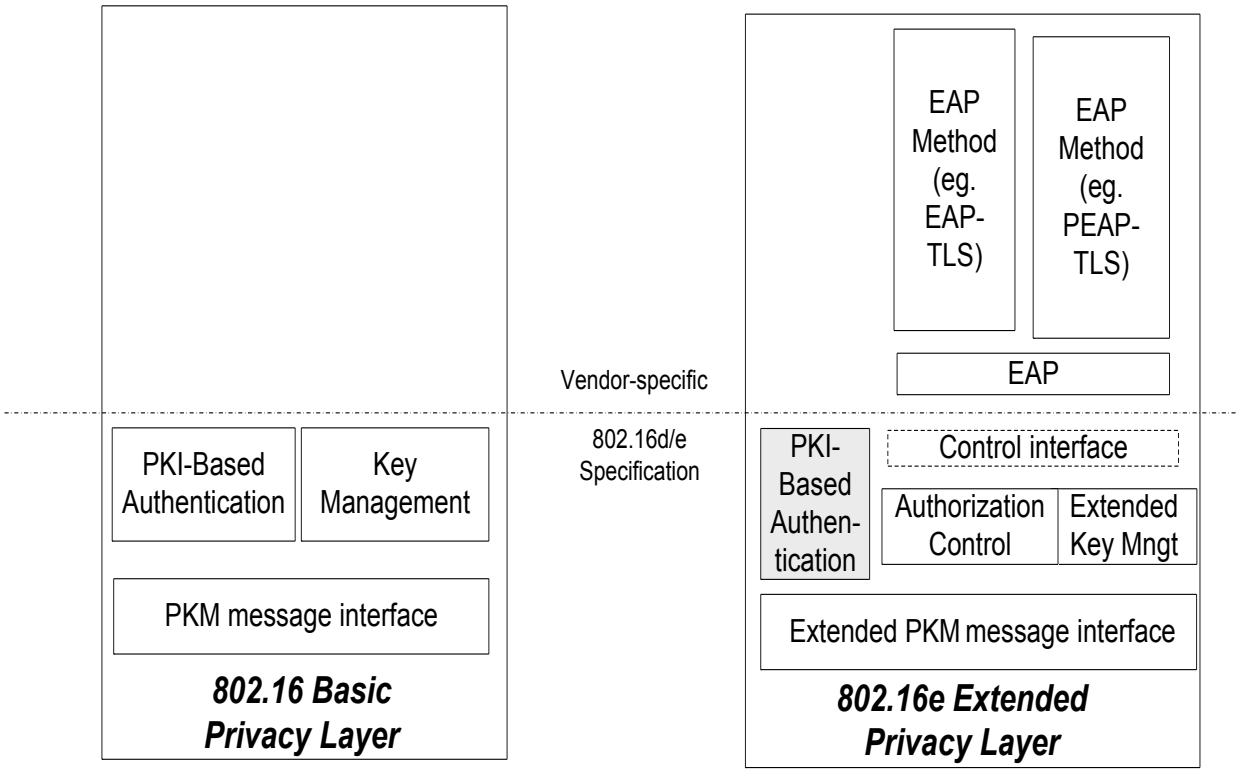
3 Description of Solution for Extensible Authentication in 802.16e

To incorporate EAP-based authentication we must address:

Network model

authentication flows
key distribution and management
coexistence with 802.16 PKM

3.1 Network Model



Comparison of components in Basic and Extended Privacy Layer

3.1.1 Overview of Components of the Extended Privacy Layer

The .16e "Extended Privacy" Layer contains:

EAP methods – these are outside the scope of the current specification, and would typically include one or more strong, well-understood authentication algorithms such as EAP-TLS.

In the case of a privacy layer that performs the role of an EAP Authenticator, the EAP methods may either reside locally, or on an Authentication Server that communicates with the Authenticator via an AAA protocol such as RADIUS.

EAP Layer – conforming to RFC 2284 (or successor RFC). The EAP layer includes the state machines for the EAP supplicant and/or authenticator roles.

Control Interface (Logical) – This is the *logical* interface that defines the signals and data that travel between 802.16e–specific modules and the generic EAP methods.

Authorization Control – This is the logical entity that modifies behaviour of the link depending on authentication status (ie. permits link initialization to continue after success or resets link after failure).

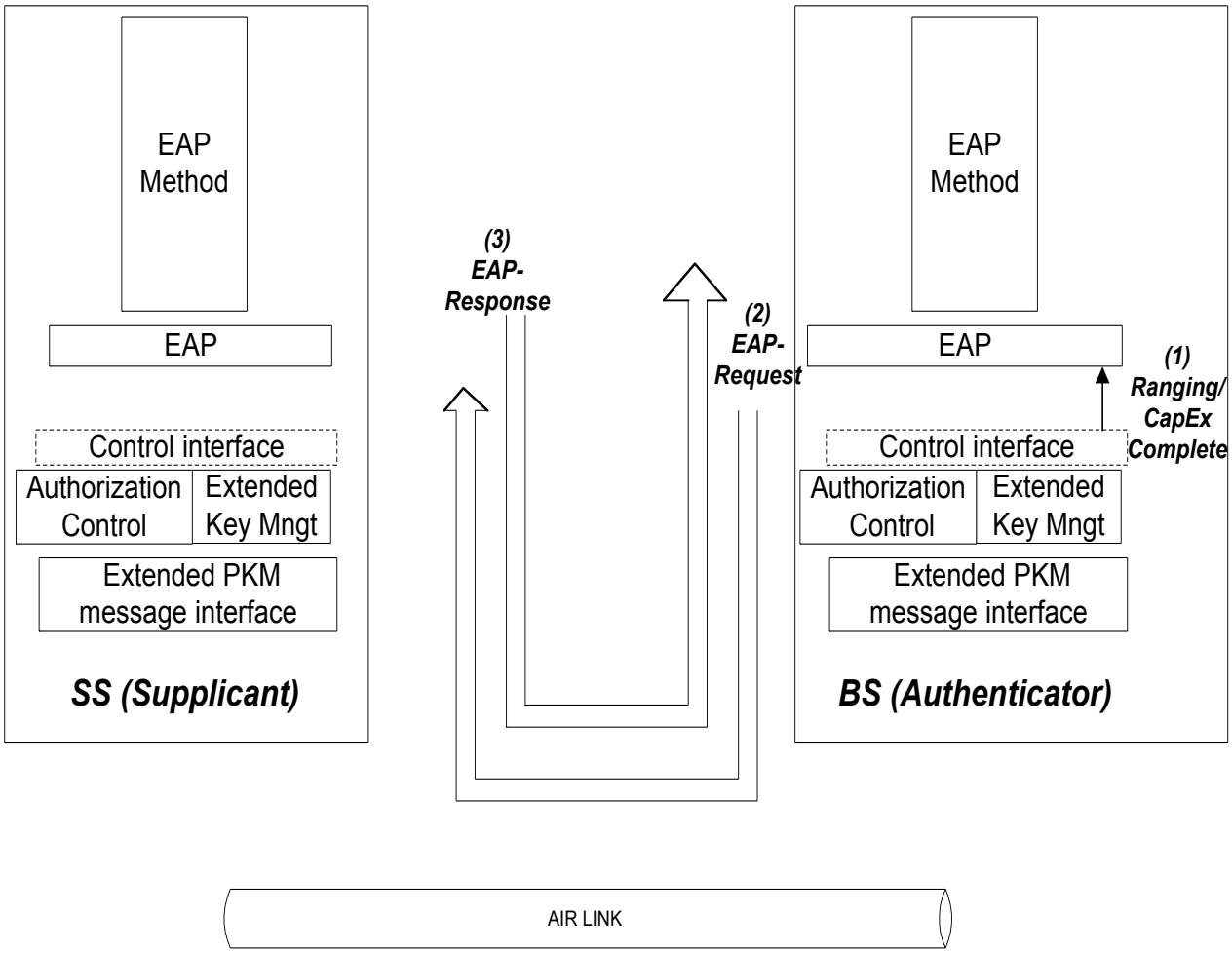
Extended Key Management – In the EAP-enabled model, the module responsible for Security Association setup and key management receives both the “authentication success” and “master key” information from the EAP layer above.

Extended PKM message interface – The MAC messages must now include a new MAC management header type for carrying encapsulated EAP traffic between the SS and BS.

3.2 Authentication Flows

In this section we illustrate a typical authorization flow for 802.16e/EAP (note that there may be exchanges between the BS and the AAA server but that these are not shown here).

3.2.1 Initial stages



Authentication Flows #1: Initial steps of EAP/802.16e Authentication

The first steps of the authorization flow are as follows:

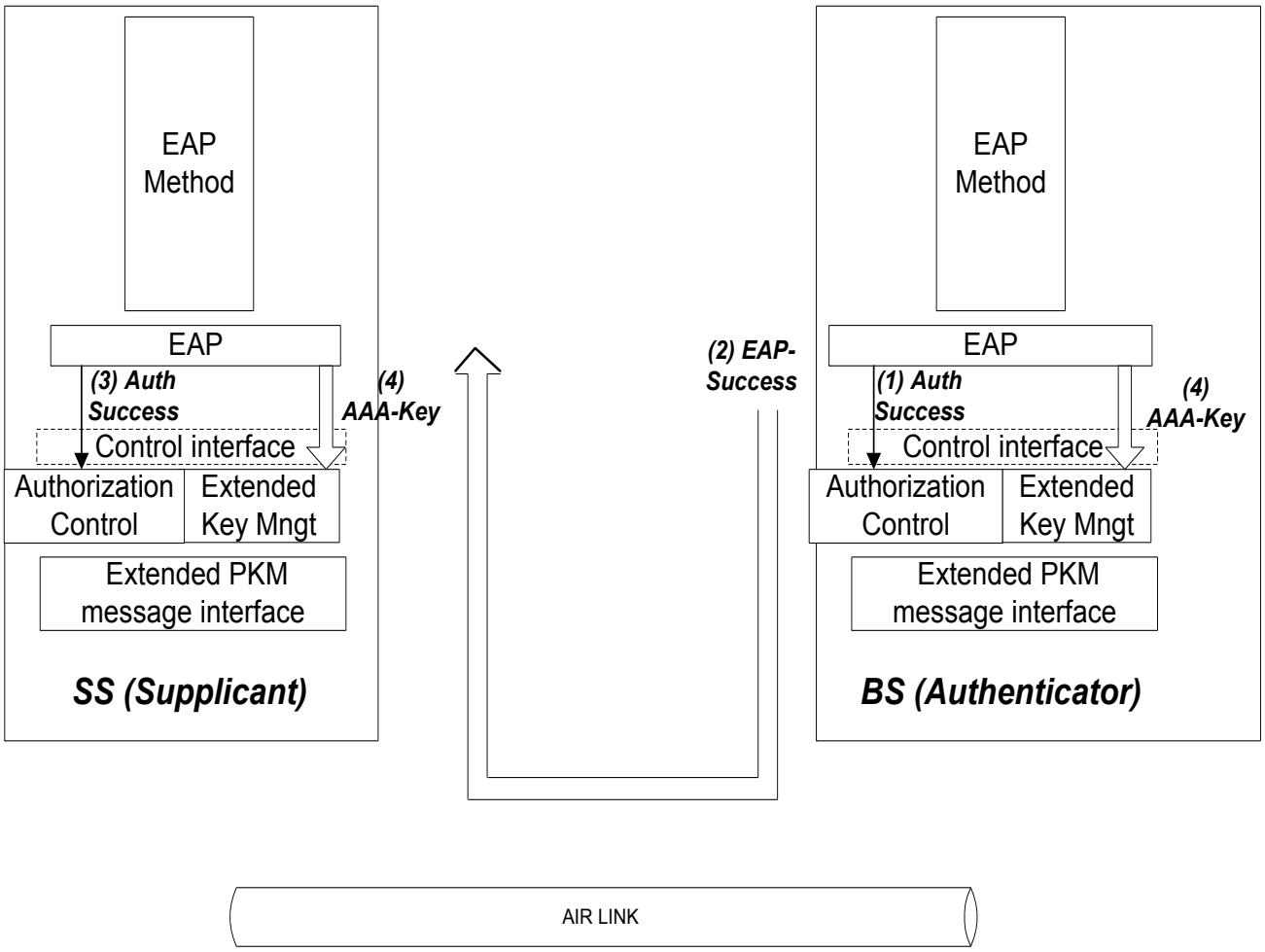
Upon successful completion of ranging (and SBC capabilities exchange), a logical signal (ie. “link activation”) is sent upwards on the Logical Control Interface at the BS (ie. the EAP authenticator). This will cause the authenticator to begin the authentication sequence.

EAP on the Authenticator sends an *EAP-Request* message to the supplicant. This Request might be an identity request or the beginning of an EAP method. The message is encapsulated in a MAC management PDU and transmitted.

EAP on the supplicant receives *EAP-Request*, passes it to the local EAP method for processing, and transmits *EAP-Response*.

Steps 2 and 3 (*EAP-Request/Response* exchange) continue as many times as needed.

3.2.2 Authentication Completion



Authentication Flows #2: Authentication Success and Export of Master Key

After one or more EAP-Request/Response exchanges, the authentication server (whether local to the Authenticator or connected remotely via an AAA protocol) determines whether or not the authentication is successful.

Upon success, EAP on the authenticator transmits a “success” signal on the logical control interface to fully activate the airlink.

EAP on the authenticator transmits EAP-success, which is then encapsulated in a MAC management message and transmitted to the supplicant.

EAP on the supplicant transmits a “success” indication on the logical control interface to fully activate the airlink.

Both EAPs (authenticator and supplicant) export the AAA-key across the logical control interface. As detailed in [3], the AAA-key is the shared “master key” that is derived by the two sides in the course of executing the EAP inner method.

At this point the authentication (and thus the involvement of the generic EAP layer) is complete.

3.2.3 Authentication Refresh

Authentication refresh will be initiated by a component on the Authenticator that resides *above* the EAP layer. Details TBD.

3.3 Security Association and Key Management in EAP/802.16e

Requirements for SA establishment and key distribution:

Subsequent to completion of authentication, the BS and SS must prove to each other that they have possession of theAAA-key - in 802.11i this is known as the "4-way handshake".

The BS and SS must negotiate the ciphersuite and the BS must provide the the SS with the list of SAIDs available to it.

Traffic Encryption Key management should be done using the current KeyReq/KeyRsp messages

Support for fast reauthentication might make it preferable to use a EAPOL-KEY-based scheme for traffic keys. Details are TBD.

3.3.1 Primary SA establishment message exchange

Immediately after authentication, the SS and BS do a simple message exchange that proves possession of the AAA-key and negotiates the ciphersuite. In this exchange, the SS provides the SAID list to the BS.

Details are simple but TBD.

3.4 Coexistence of EAP-based and Legacy-PKM-based authentication

Each BS and SS MUST support Legacy-PKM-based authentication. Support for EAP-based authentication is optional in both the BS and SS.

A particular instance of a SS's network entry procedure will use either EAP-based or Legacy-PKM-based authentication, as indicated by the SBC capabilities exchange. It will not use both EAP and Legacy-PKM in the same network entry procedure, as this would require tunnelling one authentication protocol within the other (cf. [2] section 2.1)

3.4.1 Interoperability

When an SS supports optional EAP-authentication, its product description will also list the EAP methods that it supports (eg. "EAP-TLS acc. to RFC 2716"). A compatible BS would then claim support either for RFC 2716 (for standalone support), or more likely RFC 3579 (for EAP via RADIUS, with EAP-TLS to be supported on the AAA server) etc.

The EAP protocol itself selects which supports determination of which authentication scheme should be used.

3.5 Summary – comparison of EAP-based and Basic PKM

The following shows the functions of Legacy PKM with the corresponding Extended PKM functionality:

Basic PKM	Extended PKM
Authentication in Auth Request/Auth Reply	EAP Inner Method
AK transmission in AuthRsp Auth Reply	AAA-key derivation/export
Ciphersuite negotiation and SAID assignment in Auth Request/Auth Reply	Primary SA establishment message exchange
KeyReq/Rsp Key Request / Key Reply	No change

4 Detailed Descriptions of Component Modules

4.1 Logical Control Interface

This is the *logical* interface that defines the signals and data that travel between 802.16e–specific modules and the generic EAP methods.

These Extended PKM Logical Control Interface will follow the logical interface definitions given in the EAP state machine draft [5] (or successor document) for the lower-layer interfaces of the supplicant [section 4.1] and authenticator [section 5.1].

4.2 Authorization Control

The authorization control model is the logical entity that controls link and initialization state based on the authentication state:

on Network entry:

Authentication success -> continue initialization
Failure: reset link

on Reauthentication:

Authentication success -> continue operation
Failure: either reset link, or permit flow of PKM packets only (ie. behave as “controlled port”) – details TBD.

4.3 Extended Key Management Module

The AK is derived from the AAA-Key using a TBD function.

The KEK and associated keying material is derived from the AK using the same function as the one described in the current PKM.

4.4 Cryptographic Protection of EAP exchanges

The specific threats against EAP traffic transmitted over "insecure media" (eg. Wireless) are as follows (from [2]):

[1] An attacker may try to discover user identities by snooping authentication traffic.

[2] An attacker may try to modify or spoof EAP packets.

[3] An attacker may launch denial of service attacks by spoofing lower layer indications or Success/Failure packets; by replaying EAP packets; or by generating packets with overlapping Identifiers.

[4] An attacker may attempt to recover the pass-phrase by mounting an offline dictionary attack.

[5] An attacker may attempt to convince the peer to connect to an untrusted network, by mounting a man-in-the-middle attack.

[6] An attacker may attempt to disrupt the EAP negotiation in order cause a weak authentication method to be selected.

[7] An attacker may attempt to recover keys by taking advantage of weak key derivation techniques used within EAP methods.

[8] An attacker may attempt to take advantage of weak ciphersuites subsequently used after the EAP conversation is complete.

[9] An attacker may attempt to perform downgrading attacks on lower layer ciphersuite negotiation in order to ensure that a weaker ciphersuite is used subsequently to EAP authentication.

[10] An attacker acting as an authenticator may provide incorrect information to the EAP peer and/or server via out-of-band mechanisms (such as via a AAA or lower layer protocol). This includes impersonating another authenticator, or providing inconsistent information to the peer and EAP server.

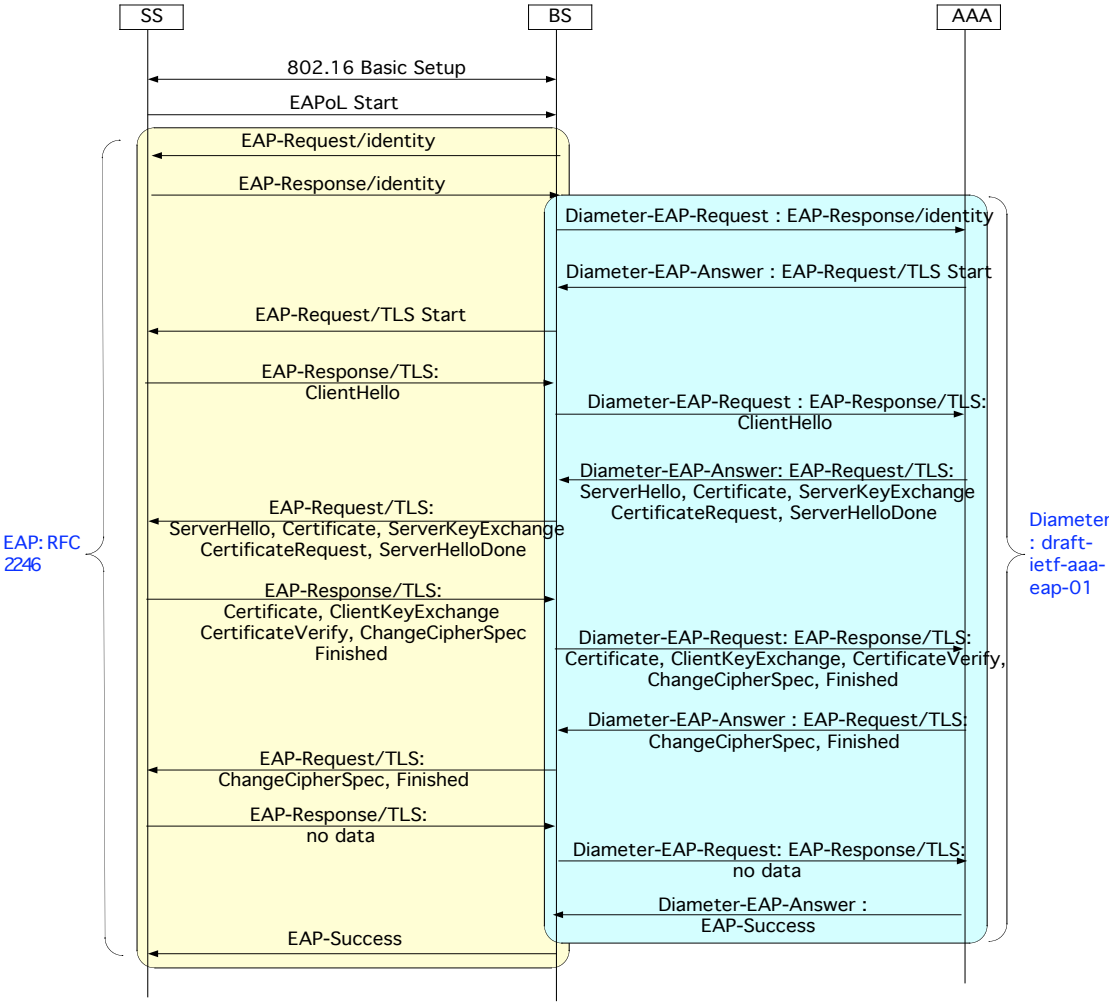
Of the above, [3] would appear to be not relevant as DoS can be easily accomplished via interference with the RF. Whereas [4]-[10] involve using EAP to exploit weaknesses elsewhere in the security architecture which we take care to prevent.

Hence it appears acceptable to utilize EAP in the 802.16e environment.

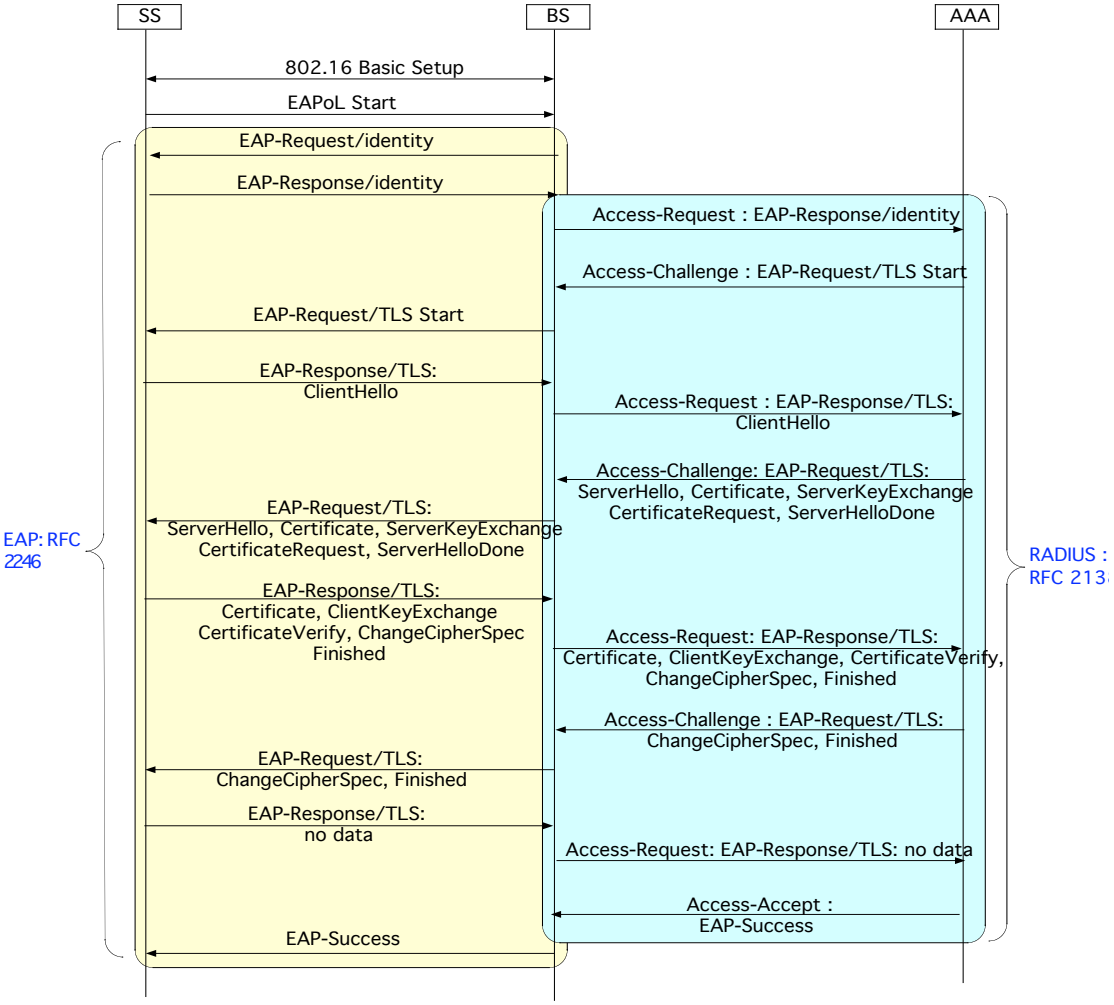
Appendix A – Example Call flow (for EAP-TLS with DIAMETER)

< diagram>

[Reference] Call flow



AAA is Diameter Server



AAA is RADIUS Server

< diagram>

Insert the State Diagram (refer to C80216e-03_63)

5 Appendix B – Background: comparision with EAP/802.1x and 802.11i

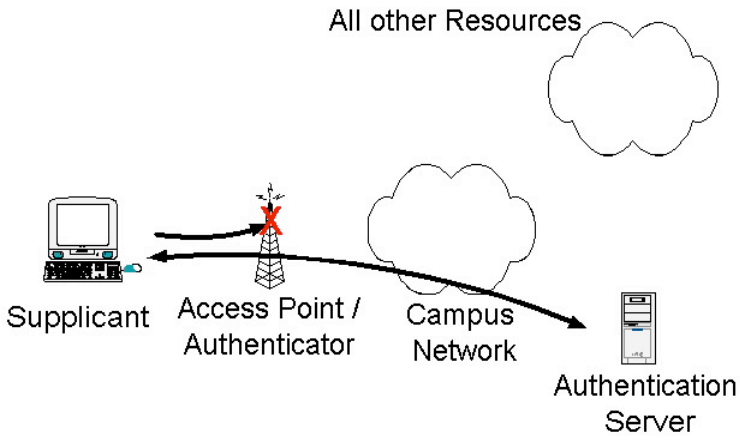
5.1.1 Model in Standard EAP/802.1x and 802.11i

5.1.1.1 Authentication flows in standard EAP/802.1x

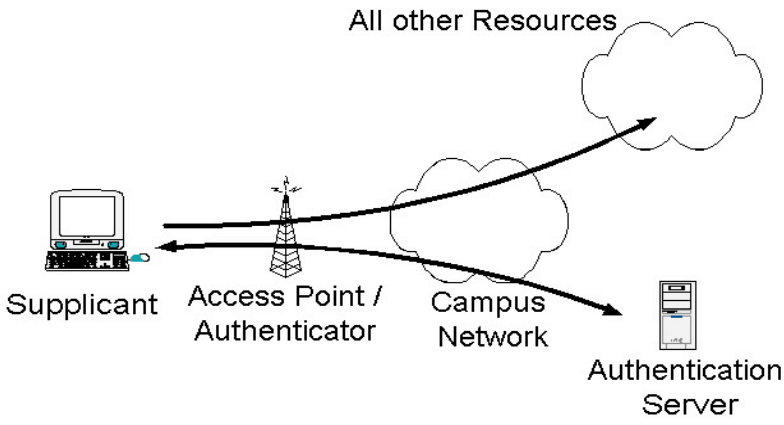
In EAP [2]/802.1x [1], authentication exchanges take place in the data plane, and *above the MAC layer*. Accordingly, the MAC layer itself must support the logical port model described in section xx of the 802.1x specification: the MAC link between the station and access point consists of two distinct **logical** links. These are termed the "controlled port" and "uncontrolled port".

<Diagram>

According to the model, EAP/802.1x packets initially flow to and from the logical uncontrolled port.



After authentication is successful, the logical “controlled port” becomes enabled – and regular data can then flow across the MAC link.



5.1.1.2 Key Distribution in 802.1xRev and 802.11i

As described in [3], successful completion of the EAP/802.1x inner method will (if desired) result in a shared-secret **AAA-Key** being exported to the Client and Authenticator from their respective EAP modules. Methods for derivation of traffic keys from the AAA-Key is ciphersuite-specific and not within the scope of the 802.1x standard.

802.11i [4] specifies mechanisms for using the AAA-Key to establish and maintain the required security associations for the 802.11 environment (including encryption key and hash derivation). Specifically, these include:

“4-way handshake” for installing unicast session keying material
“Group Key handshake” to “push” the data broadcast keying material to the client in EAPOL-KEY messages

5.1.1.3 Differences with 802.16

Simply running 802.1x on top of 802.16 would obviously be the most appealing approach. The problem is that doing this would require significant changes in the MAC because the authenticated identity and the Security Associations (for which authentication is a prerequisite step) are used in link initiation.

Specifically:.

- DSx messages reference the SAIDs which are (currently) provisioned in the PKM step.
- Registration requires data origin integrity (ie. HMAC) and uses the authenticated Identity to provision connections

.

- Handover completion should occur only after the user has been authenticated.

Moving authentication above the MAC would require reworking these mechanisms. As well, 802.16 is a bit of a strange beast in that it supports "convergence layers" at different levels of the OSI stack. Relying on 802.1x would not provide a solution for non-802.3-style convergence layers ie. IP.

The alternative is to perform the EAP-based authentication as part of the link initiation (ie. the way that it's done in the current Privacy Layer - which is also the "traditional" way to do link-layer authentication).

This approach avoids all the problems mentioned above and is compatible with the EAP interface to a lower-layer formalized in draft-ietf-eap-statemachine and Appendix F of 802.1xRev ie. (broadly):

o EAP receives a logical indication that the channel (ie. secondary mgt connection) is available for

EAP message exchange

o EAP sends and receives EAP messages (in the formats described in RFC 2284bis), which are in turn transmitted or received in PKM-encapusulated msgs.

o EAP provides the 802.16 control layer with indication of authentication success/failure, and possibly provides master key data upon authentication success

6 References

[1] IEEE 802.1Xrev
[2] RFC 2284bis IETF draft
[3] EAP Keying Framework IETF draft
[4] IEEE 802.11i
[5] State Machines for EAP Peer and Authenticator IETF draft

7 Specific 802.16e text changes

[6.4.2.3.9 Change Table 26 – PKM Message codes]

	PKM Message Type	MAC Message Type
0 ~2	Reserved	
3	SA Add	PKM-RSP
4	Auth Request	PKM-REQ
5	Auth Reply	PKM-RSP
6	Auth Reject	PKM-RSP
7	Key Request	PKM-REQ
8	Key Reply	PKM-RSP
9	Key Reject	PKM-RSP
10	Auth Invalid	PKM-RSP
11	TEK Invalid	PKM-RSP
12	Auth Info	PKM-REQ
13	EAP Transfer Request	PKM-REQ
14	EAP Transfer Reply	PKM-RSP
15 ~ 255	reserved	

[Add section 6.4.2.3.9.11 EAP Transfer Request message]

When an SS has an EAP message received from an EAP method for transmission to the BS, it encapsulates it in an EAP Transfer Request message.

Code : 13

Attributes are shown in Table xx.

Table xx EAP Transfer Request attributes

Attribute	Contents
EAP Payload	Contains the EAP authentication data, not interpreted in the MAC

The EAP Payload field carries data in the format described in RFC2254bis (or successor RFC) section 4.

[Add section 6.4.2.3.9.11 EAP Transfer Response message]

When an BS has an EAP message received from an EAP method for transmission to the SS, it encapsulates it in an EAP Transfer Request message.

Code : 14

Attributes are shown in Table xx.

Table xx EAP Transfer Request attributes

Attribute	Contents
EAP Payload	Contains the EAP authentication data, not interpreted in the MAC

The EAP Payload field carries data in the format described in RFC2254bis (or successor RFC) section 4.

[Change section 7.1.2 Key Management protocol]

The PKM protocol facilitates mutual authentication of the SS and BS, as well as distribution of traffic keying material from the BS to the SS. It also supports periodic reauthentication/reauthorization and key refresh. The key management protocol uses either EAP [IETF RFC 2284], or X.509 digital certificates [IETF RFC 3280] together with RSA public-key encryption algorithm [PKCS #1] to perform authentication. It uses strong symmetric algorithms to perform key exchanges between SS and BS.

The PKM's authentication protocol establishes a shared secret (i.e., an AK) between SS and BS. The shared secret is then used to secure subsequent PKM exchanges of TEKs. This two-tiered mechanism for key distribution permits refreshing of TEKs without incurring the overhead of computation-intensive public-key operations.

A BS authenticates a client SS during the initial authorization exchange. Each SS presents its credentials, which will be a unique X.509 digital certificate issued by the SS's manufacturer (in the case of RSA authentication) or a vendor-specific credential (in the case of EAP-based authentication).

The BS associates an SS's authenticated identity to a paying subscriber, and hence to the data services that subscriber is authorized to access. Thus, with the AK exchange, the BS establishes an authenticated identity of a client SS and the services (i.e., specific TEKs) the SS is authorized to access.

Since the BS authenticates the SS, it can protect against an attacker employing a *cloned* SS, masquerading as a legitimate subscriber's SS.

The traffic-key management portion of the PKM protocol adheres to a client/server model, where the SS (a PKM "client,") requests keying material, and the BS (a PKM "server") responds to those requests, ensuring that individual SS clients receive only keying material for which they are authorized.

The PKM protocol uses MAC management messaging, i.e., PKM-REQ and PKM-RSP messages defined in 6.4.2.3. The PKM protocol is defined in detail in 7.2.

[Add new section 7.1.3 Authentication protocol}

An SS uses the PKM protocol to obtain authorization and traffic keying material from the BS, and to support periodic reauthorization and key refresh.

PKM supports two distinct authentication protocol mechanisms:

- RSA [PKCS #1] (support is mandatory in all devices)

- Extensible Authentication Protocol (support is optional as described in xx)

7.1.3.1 PKM RSA Authentication

The PKM RSA authentication protocol uses X.509 digital certificates [IETF RFC 3280], the RSA public-key encryption algorithm [PKCS #1].

A BS authenticates a client SS during the initial authorization exchange. Each SS carries a unique X.509 digital certificate issued by the SS's manufacturer. The digital certificate contains the SS's Public Key and SS MAC address. When requesting an AK, an SS presents its digital certificate to the BS. The BS verifies the digital certificate, and then uses the verified Public Key to encrypt an AK, which the BS then sends back to the requesting SS.

All SSs shall have factory-installed RSA private/public key pairs or provide an internal algorithm to generate such key pairs dynamically. If an SS relies on an internal algorithm to generate its RSA key pair, the SS shall generate the key pair prior to its first AK exchange, described in 7.2.1. All SSs with factory-installed RSA key pairs shall also have factory-installed X.509 certificates. All SSs that rely on internal algorithms to generate an RSA key pair shall support a mechanism for installing a manufacturer-issued X.509 certificate following key generation.

7.1.3.2 PKM EAP Authentication

PKM EAP Authentication uses Extensible Authentication Protocol [IETF RFC 2284bis] in conjunction with a vendor-selected standardized *EAP Method* (eg. EAP-TLS [IETF RFC 2716]). The EAP method will use a particular kind of credential – such as an x.509 certificate in the case of EAP-TLS, or a Subscriber Information Module in the case of EAP-SIM (Draft xxxxx).

The particular credentials and EAP methods that are to be used are outside of the scope of this specification, but they should be selected with awareness of the security issues described in [IETF RFC 2284bis] section 7.

Figure xx shows the relationship between the lower levels of the 802.16 MAC and the generic EAP components (and the interface between them).

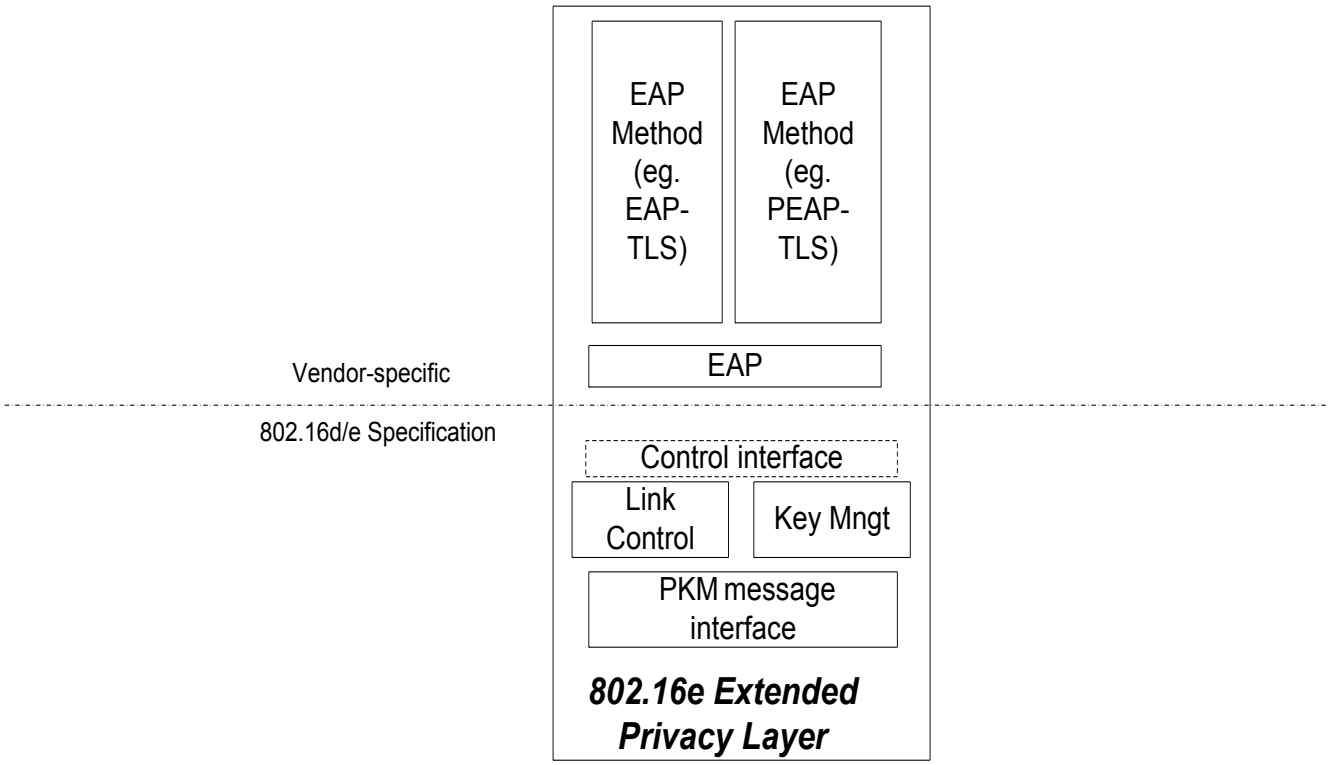

EAP Method (eg. EAP-TLS)
EAP Method (eg. PEAP-TLS)
EAP
Vendor-specific
802.16d/e Specification
Control interface
Link Control
Key Mngt
PKM message interface
802.16e Extended Privacy Layer

Figure xx

[Change 7.2.1 SS Authorization and AK derivation overview]

SS authorization, controlled by the Authorization state machine, is the process of

a) the BS authenticating a client SS's identity[1]

b) the BS and SS establishing a shared AK, from which a key encryption key (KEK) and message authentication keys are derived

c) the BS providing the authenticated SS with the identities (i.e., the SAIDs) and properties of primary and static SAs the SS is authorized to obtain keying information for

After achieving initial authorization, an SS periodically reauthorizes with the BS; reauthorization is also managed by the SS's Authorization state machine. TEK state machines manage the refreshing of TEKs.

7.2.1.1 Authorization via PKM RSA Authentication Protocol

An SS begins authorization by sending an Authentication Information message to its BS. The Authentication Information message contains the SS manufacturer's X.509 certificate, issued by the manufacturer itself or by an external authority. The Authentication Information message is strictly informative; i.e., the BS may choose to ignore it. However, it does provide a mechanism for a BS to learn the manufacturer certificates of its client SS.

The SS sends an Authorization Request message to its BS immediately after sending the Authentication Information message. This is a request for an AK, as well as for the SAIDs identifying any Static Security SAs the SS is authorized to participate in. The Authorization Request includes

a) a manufacturer-issued X.509 certificate

b) a description of the cryptographic algorithms the requesting SS supports; an SS's cryptographic capabilities are presented to the BS as a list of cryptographic suite identifiers, each indicating a particular pairing of packet data encryption and packet data authentication algorithms the SS supports

c) the SS's Basic CID. The Basic CID is the first static CID the BS assigns to an SS during initial ranging—the primary SAID is equal to the Basic CID

In response to an Authorization Request message, a BS validates the requesting SS's identity, determines the encryption algorithm and protocol support it shares with the SS, activates an AK for the SS, encrypts it with the SS's public key, and sends it back to the SS in an Authorization Reply message. The authorization reply includes:

a) an AK encrypted with the SS's public key

b) a 4-bit key sequence number, used to distinguish between successive generations of AKs

c) a key lifetime

d) the identities (i.e., the SAIDs) and properties of the single primary and zero or more static SAs the SS is authorized to obtain keying information for

[1] The SS may optionally authenticate the BS in EAP authentication

While the Authorization Reply shall identify Static SAs in addition to the Primary SA whose SAID matches the requesting SS's Basic CID, the Authorization Reply shall not identify any Dynamic SAs.

The BS, in responding to an SS's Authorization Request, shall determine whether the requesting SS, whose identity can be verified via the X.509 digital certificate, is authorized for basic unicast services, and what additional statically provisioned services (i.e., Static SAIDs) the SS's user has subscribed for. Note that the protected services a BS makes available to a client SS can depend upon the particular cryptographic suites SS and BS share support for.

An SS shall periodically refresh its AK by reissuing an Authorization Request to the BS. Reauthorization is identical to authorization with the exception that the SS does not send Authentication Information messages during reauthorization cycles. Subclause 7.2.4's description of the authorization state machine clearly indicates when Authentication Information messages are sent.

To avoid service interruptions during reauthorization, successive generations of the SS's AKs have overlapping lifetimes. Both SS and BS shall be able to support up to two simultaneously active AKs during these transition periods. The operation of the Authorization state machine's Authorization Request scheduling algorithm, combined with the BS's regimen for updating and using a client SS's AKs (see 7.4), ensures that the SS can refresh.

7.2.1.2 Authorization via PKM Extensible Authentication Protocol

The first steps of the authorization flow are as follows:

1) Upon successful completion of ranging (and capabilities exchange), a logical signal (ie. "link activation") is sent upwards on the Logical Control Interface at the BS (ie. the EAP authenticator). This will cause the authenticator to begin the authentication sequence.

2) EAP on the Authenticator sends an *EAP-Request* message to the supplicant. This Request might be an EAP identity request or the beginning of an EAP method. The message is encapsulated in a MAC management PDU and transmitted.

3) EAP on the supplicant receives *EAP-Request*, passes it to the local EAP method for processing, and transmits *EAP-Response*.

Steps 2 and 3 (*EAP-Request/Response* exchange) continue as many times as needed.

After one or more EAP-Request/Response exchanges, the authentication server (whether local to the Authenticator or connected remotely via an AAA protocol) determines whether or not the authentication is successful.

The next steps of the authorization flow are as follows:

1) Upon success, EAP on the authenticator transmits a "success" signal on the logical control interface to fully activate the airlink.

2) EAP on the authenticator transmits EAP-success, which is then encapsulated in a MAC management message and transmitted to the supplicant.

3) EAP on the supplicant transmits a "success" indication on the logical control interface to fully activate the airlink.

4) Both EAPs (authenticator and supplicant) export the AAA-key across the logical control interface. As detailed in [3], the AAA-key is the shared "master key" that is derived by the two sides in the course of executing the EAP inner method

The authentication part of the authorization flow (and the involvement of the generic EAP layer) is now complete.

The final steps of the authorization flow:

1) BS sends Auth-handshake msg (structure TBD) to SS to supply a nonce, and includes its ciphersuite capabilities and a list of SAIDs that are available to the SS.

2) SS sends Auth-handshake reply msg (structure TBD) to supply its nonce and includes an HMAC based on its TBD-function-derived AK

The Authorization Key (AK) is derived from the AAA-Key (Derivation algorithm TBD)

[in 7.4.1.2]
[Figure 99 Modified]

Figure 99— Authorization procedure in BS and SS

[Add to Table 129] PKM Attribute types

Type	PKM Attributes
28	EAP Payload

[Under 11.2.19.7]
[11.2.20] EAP Payload
Description : The EAP Payload attribute is not interpreted in this MAC layer, which contains a data payload for EAP-TLS or EAP-TTLS. This attribute uses only an EAP Transfer Request and an EAP Transfer Reply.

Type	Length	Value (string)
28	n	EAP payload data