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. 2**
Date Submitted	**2003-12-29**
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/r1
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

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, PKI-based device certificates (eg. various forms of provider-supplied smartcards to be installed in an off-the-shelf SS device)
- facilitation of handover to other media (ie. 802.11) by providing hooks for preauthentication or other functions

2.2 Requirements

The solution must satisfy the following:

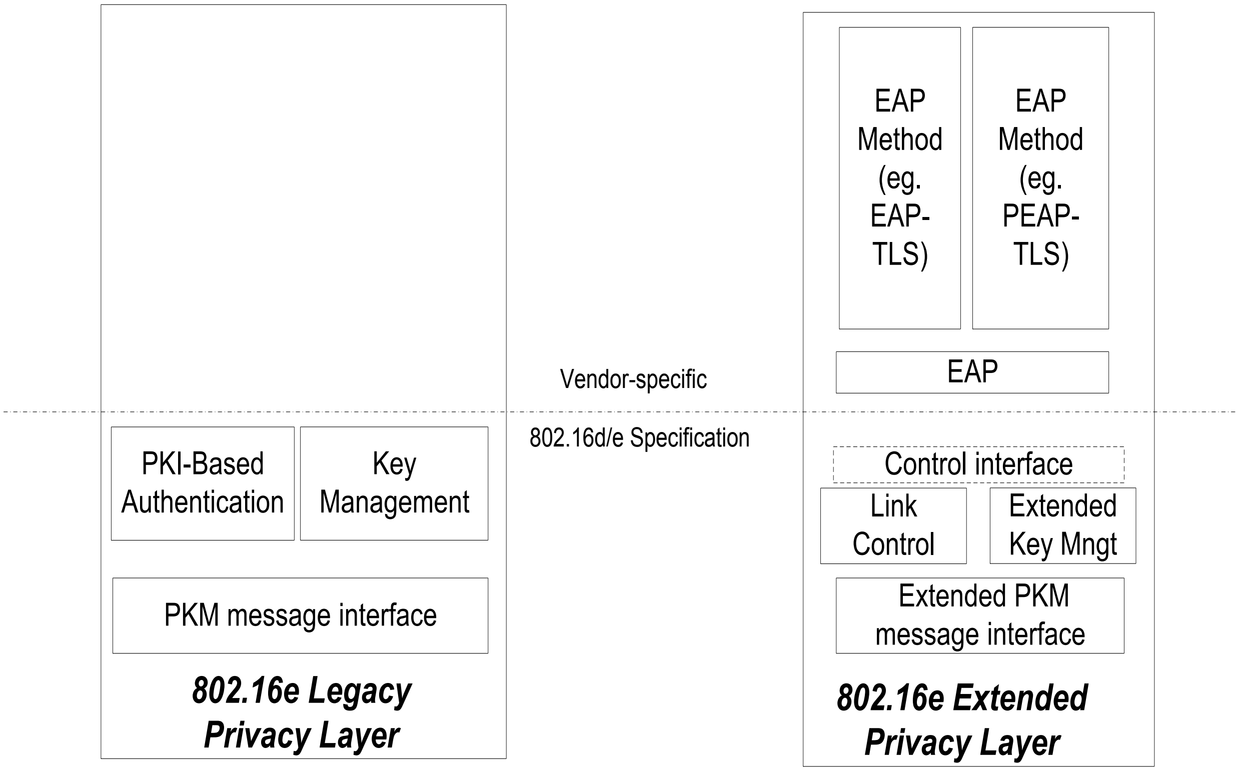
- Interoperability with non-EAP enabled 802.16d/e systems
- Compatibility with the standard EAP/802.1x model **so that methods and analysis pertaining to standard EAP will be applicable** (though not necessarily recommended) for 802.16e
- Support for 802.16 primary, static, and dynamic security associations. These include SAs for both unicast and non-unicast MAC-layer connections.
- Provision for ciphersuite selection and authorization refresh
- Appropriate compliance with security recommendations for EAP in a wireless environment

3 Description of Solution for 802.16e/EAP

To describe 802.16/EAP we must address:

- Network model
- authentication flows
- key distribution and management
- coexistence with 802.16 PKM

3.1 Network Model



Comparison of components in Legacy and Extended Privacy Layer

3.1.1 Overview of Components of the Extended Privacy Layer

The .16e “Extended Privacy” Layer contains:

1. 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.

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

3. 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.

4. Link Control – This is the logical entity that restricts the flow of most data packets until authentication has completed.

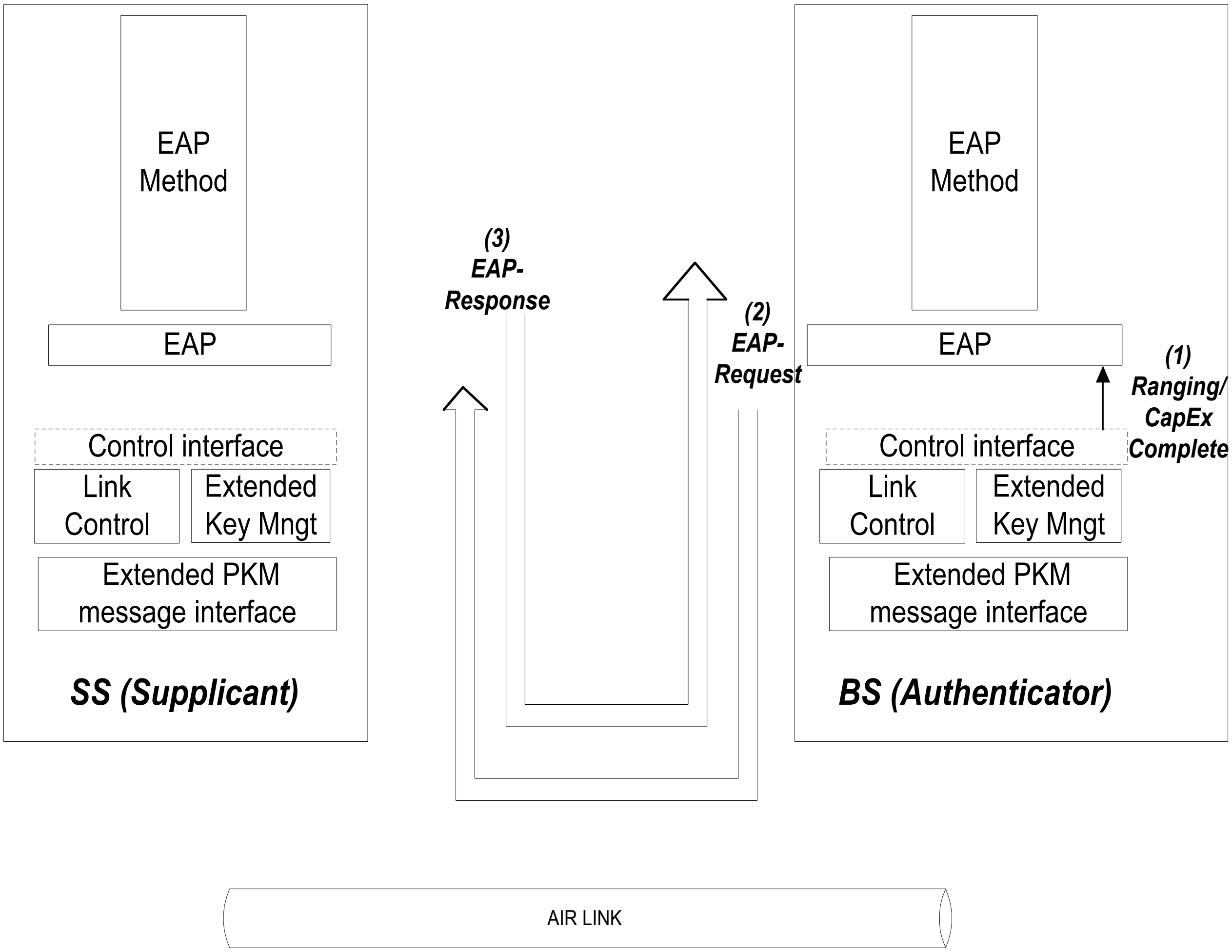
5. Extended Key Management – In the EAP-enabled model, the module responsible for Security Association setup and key management receives both the "authentication success" and "master key" information from the EAP layer above.

6. 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:

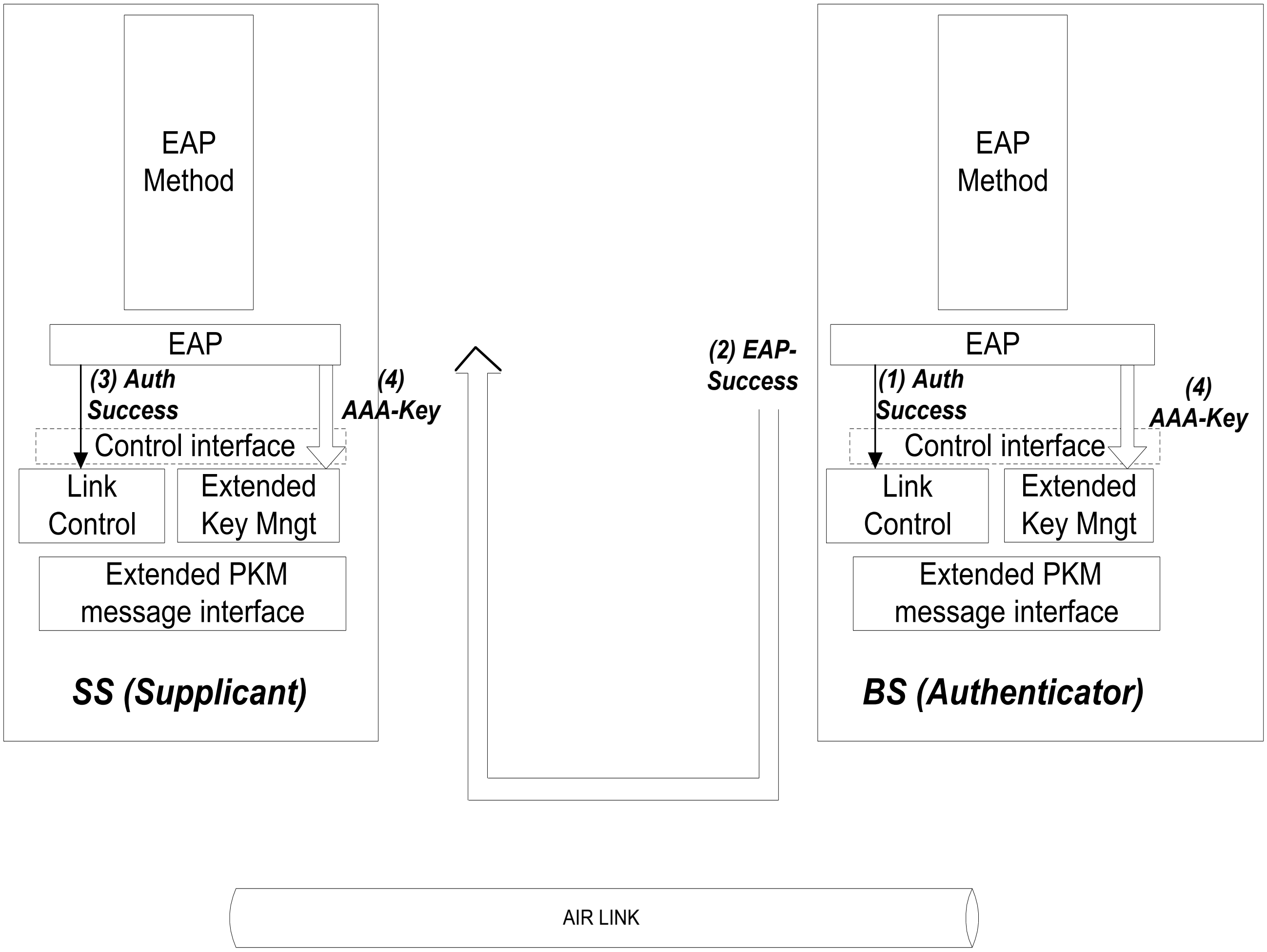
1) Upon successful completion of ranging (and capabilities exchange), a logical signal (ie. “link activation”) is sent upwards on the Logical Control Interface at the BS (ie. the EAP authenticator). This will cause the authenticator to begin the authentication sequence.

2) EAP on the Authenticator sends an *EAP-Request* message to the supplicant. This Request might be an identity request or the beginning of an EAP method. The message is encapuslated 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.

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.

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.

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 correspond to those in 802.11i:

- Subsequent to completion of authentication there is a mechanism similar to the 802.11i "4-way handshake". Details TBD.

- Ciphersuite and SAID information is provided by the BS to the SS. Details TBD

- Traffic key management is 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.

3.4 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)

3.5 Summary – comparison of EAP-based and Legacy PKM

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

Legacy PKM	**Extended PKM**
AuthReq/Rsp Auth Request/Auth Reply	EAP Inner Method
AK transmission in AuthRsp Auth Reply	AAA-key derivation/export

KeyReq/Rsp Key Request / Key Reply	Key Distribution in EAPOL-key Or Key Request / Key Reply (compatibility with 802.16

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 Link Control

The link control model is the logical entity that restricts the flow of most data packets until authentication has completed.

4.2.1 Controlled/Uncontrolled Port

Associated with the link control module are the notions of a logical "controlled port" and "uncontrolled port".

The logical "uncontrolled port" carries the packets which can flow when authentication state is "not authenticated" – ie. ranging, sbc, and pkm. The logical "controlled port" carries the packet traffic which is permitted by 802.16 to flow only after authentication has completed successfully.

Upon successful authentication, the link control module sets the controlled port to active state.

4.3 Extended Key Management Module

TBD

4.4 Format for transmission of EAP packets in 802.16 MAC Layer

<import from ETRI document>
SBC-REQ/RSP Message Format (referrer to C802.16-03-62)

EAP-Transfer Request/Reply message Format (referrer to C802.16-03-63)

[in 6.2.2.3.9]

[Add to Table 25]

Code	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	

6.2.2.3.9.2.2 EAP Transfer Request message

A client SS sends several different EAP Transfer Request messages to BS for the SS authorization, until SS is obtained AK. The EAP Transfer Request may contain the Security-Capabilities, the SAID, the SS's public key, and the EAP Payload. The Security-Capabilities, SAID, and SS's public key parameters should be included only in the 1'st EAP Transfer Request message among several EAP Transfer Request messages.

Code : 13

Attributes are shown in Table 27-b.

Table 27-b EAP Transfer Request attributes

Attribute	**Contents**
Security-Capabilities	Describes requesting SS's security capabilities
SAID	Security Association ID, being equal to the Basic CID
EAP Payload	Contains the authorization data, not interpreted in the MAC.

Security-Capabilities attribute is a compound attribute describing the requesting SS's security capabilities. This includes the data encryption and data authentication algorithms the SS supports.

An SAID attribute contains a Privacy SAID. In this case, the provided SAID is the SS's Basic CID, which is equal to the basic CID assigned to the SS during initial ranging.

SS's Public Key attribute is used only when AK is generated by BS.

EAP Payload attribute indicates authorization data payload for user authentication and is forwarded to upper authorization protocol without interpreting in the MAC sublayer. Payload format is according to RFC2254bis section 4.

6.2.2.3.9.3.2 EAP Transfer Reply message

Sent by BS to a client SS in response to an EAP Transfer Request message, the EAP Transfer Reply message may contain an EAP Result Code, Error Code, the Key-Lifetime, the Key-Sequence-Number, and a list of SA-Descriptors identifying the Primary and Static SAs. The requesting SS is authorized to access and one's particular properties (e.g., type, cryptographic suite). The SA Descriptor list shall include a descriptor for the Basic CID reported to the BS in the corresponding EAP Transfer Request. The SA-Descriptor list may include descriptors of Static SAIDs which are used for the SS authorization.

In addition, the EAP Result Code, Key-Sequence-Number, Key-Lifetime, SA-Descriptor parameters should be included only in the last EAP Transfer Reply (Success) message among several EAP Transfer Reply messages. When any errors occur in SS's authorization procedure, BS sends EAP Transfer Reply (Failure) message. The EAP Result code and Error Code should be included in the EAP Transfer Reply (Failure) message.

Code :14

Attributes are shown in Table 28-b

Table 28-b EAP Transfer Reply attributes

Attribute	Contents
EAP Result Code	Describes success or failure
Error Code	Error code identifying reason for rejection or failure of authorization request.
Key-Sequence-Number	Authorization key sequence number
Key-Lifetime	Authorization key life time
SA Descriptor	Specifies an SA ID and additional properties of the SA
EAP Payload	Contains the EAP-TLS Data, not interpreted in the MAC

The EAP Result Code describes success or failure of the SS's user authentication result.

The Error Code is used only if EAP Result Code describes failure and identifies reason for failure of authorization request.

The AUTH-Key is used only if BS assigns this Key and encrypted with the target client SS's public key.

The Key-Sequence-Number is authorization key sequence number and the Key-Lifetime is authorization key lifetime.

The SA Descriptor specifies an SAID and additional properties of the SA.

EAP Payload attribute indicates authorization data payload for user authentication and is forwarded to upper authorization protocol without interpreting in the MAC sublayer.

[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
29	EAP Result Code
30	SS' Public Key

[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

[in 11.2.21] EAP Result Code

Description : The EAP Result Code attribute indicates the error status, is included in an EAP Transfer Reply.

Type	Length	Value (string)
29	1	0 : Success 1 : Failure

[in 11.2.22] SS's Public Key

Description : The SS's public key attribute indicates public key of SS. AUTH-Key is encrypted with this SS's public key parameter.

Type	Length	Value (string)
30	Variable	

4.5 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 rely exclusively on the cryptographic protection provided by the EAP inner method.

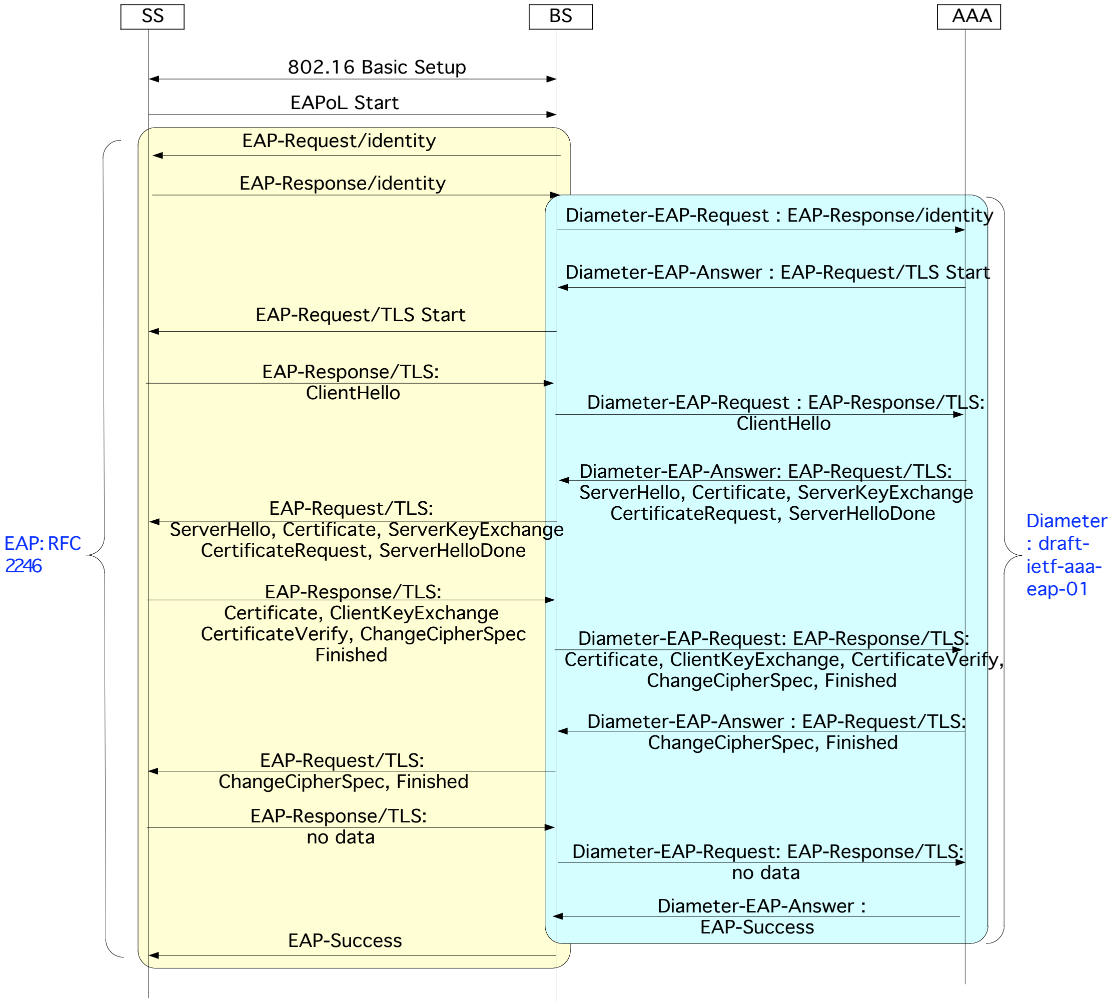
5 Specific 802.16e text changes

<tbd>

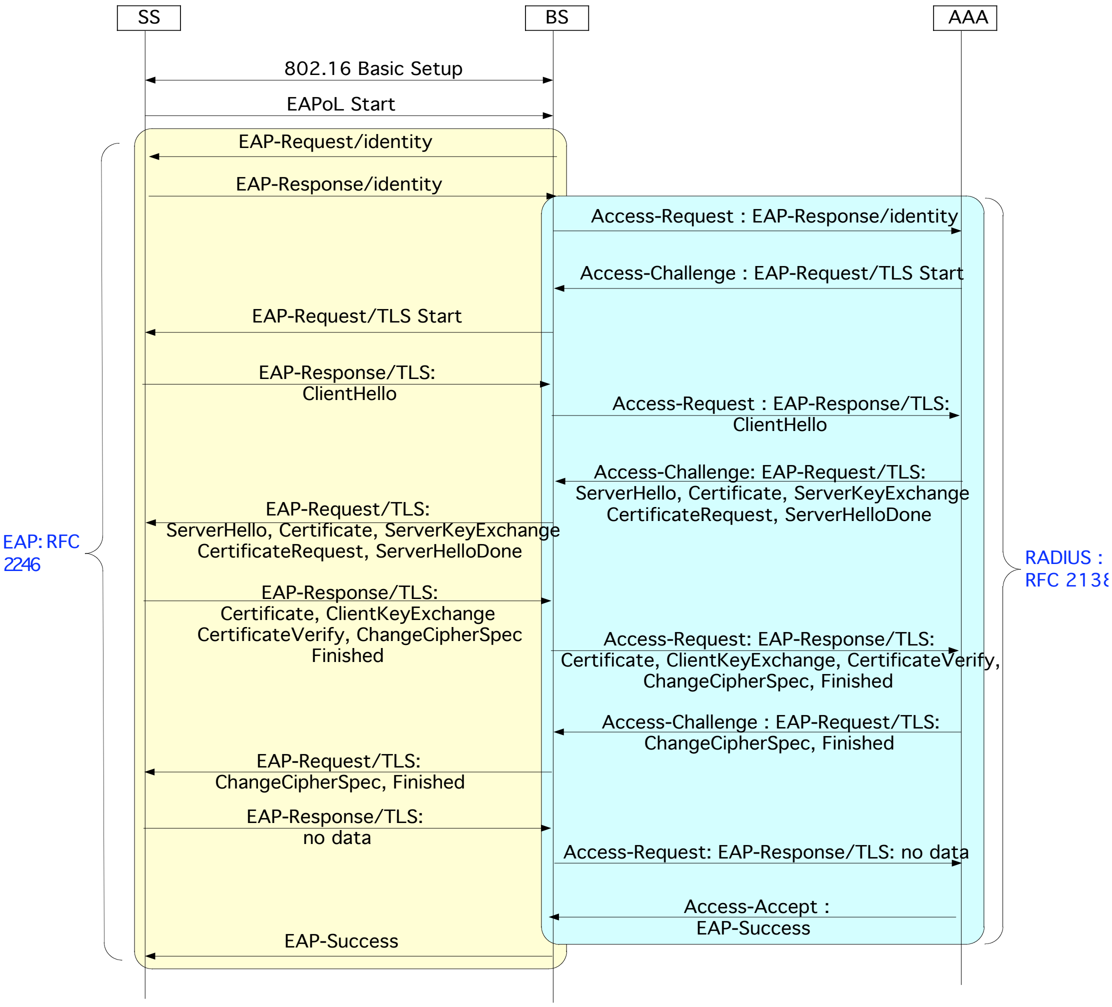
6 Appendix A – Call flow (for EAP-TLS with DIAMETER)

< diagram>

[Reference] Call flow


SS
BS
AAA
802.16 Basic Setup
EAPoL Start
EAP-Request/identity
EAP-Response/identity
Diameter-EAP-Request : EAP-Response/identity
Diameter-EAP-Answer : EAP-Request/TLS Start
EAP-Request/TLS Start
EAP-Response/TLS: ClientHello
Diameter-EAP-Request : EAP-Response/TLS: ClientHello
Diameter-EAP-Answer: EAP-Request/TLS: ServerHello, Certificate, ServerKeyExchange CertificateRequest, ServerHelloDone
EAP-Request/TLS: ServerHello, Certificate, ServerKeyExchange CertificateRequest, ServerHelloDone
EAP: RFC 2246
Diameter : draft-ietf-aaa-eap-01
EAP-Response/TLS: Certificate, ClientKeyExchange CertificateVerify, ChangeCipherSpec Finished
Diameter-EAP-Request: EAP-Response/TLS: Certificate, ClientKeyExchange, CertificateVerify, ChangeCipherSpec, Finished
Diameter-EAP-Answer : EAP-Request/TLS: ChangeCipherSpec, Finished
EAP-Request/TLS: ChangeCipherSpec, Finished
EAP-Response/TLS: no data
Diameter-EAP-Request: EAP-Response/TLS: no data
Diameter-EAP-Answer : EAP-Success
EAP-Success

AAA is Diameter Server


SS
BS
AAA
802.16 Basic Setup
EAPoL Start
EAP-Request/identity
EAP-Response/identity
Access-Request : EAP-Response/identity
Access-Challenge : EAP-Request/TLS Start
EAP-Request/TLS Start
EAP-Response/TLS: ClientHello
Access-Request : EAP-Response/TLS: ClientHello
Access-Challenge: EAP-Request/TLS: ServerHello, Certificate, ServerKeyExchange CertificateRequest, ServerHelloDone
EAP-Request/TLS: ServerHello, Certificate, ServerKeyExchange CertificateRequest, ServerHelloDone
EAP: RFC 2246
RADIUS : RFC 2138
EAP-Response/TLS: Certificate, ClientKeyExchange CertificateVerify, ChangeCipherSpec Finished
Access-Request: EAP-Response/TLS: Certificate, ClientKeyExchange, CertificateVerify, ChangeCipherSpec, Finished
Access-Challenge : EAP-Request/TLS: ChangeCipherSpec, Finished
EAP-Request/TLS: ChangeCipherSpec, Finished
EAP-Response/TLS: no data
Access-Request: EAP-Response/TLS: no data
Access-Accept : EAP-Success
EAP-Success

AAA is RADIUS Server

< diagram>

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

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

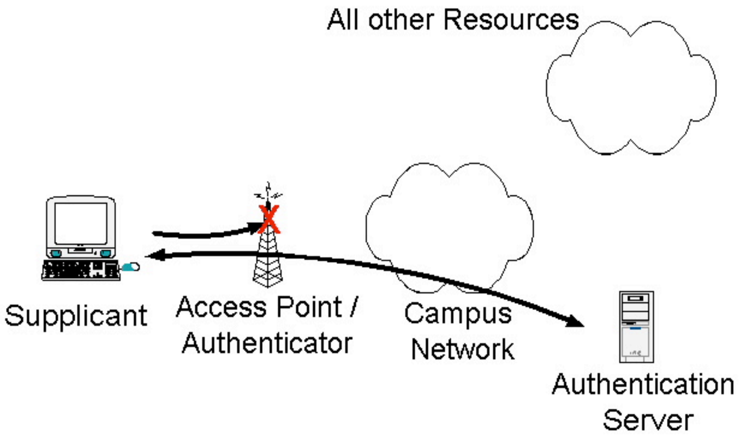
7.1.1 Model in Standard EAP/802.1x and 802.11i

7.1.1.1 Authentication flows in standard EAP/802.1x

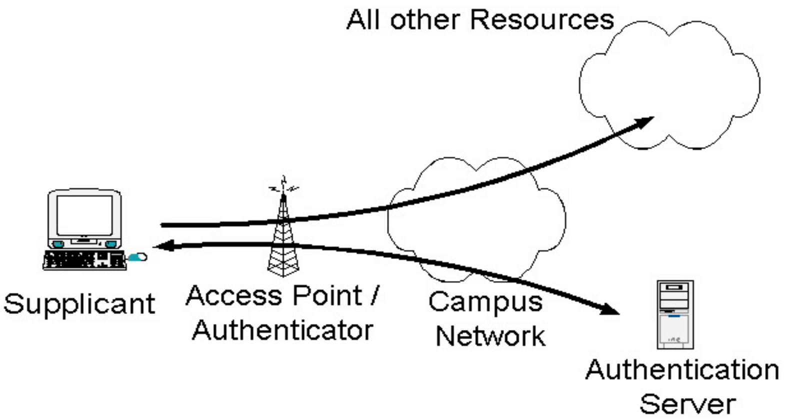
In EAP [2]/802.1x [1], authentication exchanges take place in the data plane, and *above the MAC layer*. Accordingly, the MAC layer itself must support the logical port model described in section xx of the 802.1x specification: the MAC link between the station and access point consists of two distinct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



7.1.1.2 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

7.1.1.3 Differences with 802.16

Note that in 802.16:

- EAP-based exchanges perform the same functions as the current PKM (ie. authentication and key/SA management).

- EAP-based functions must be separate from whatever convergence layer runs on top of the MAC layer.

Consequently:

- EAP operation and state machines will be completely contained **within** the 802.16 privacy layer.
- EAP messages will be encapsulated inside the payload of PKM MAC messages

8 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