Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**Secure Association Establishment for PKM-EAP**
Date Submitted	**2004-03-17**
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	
Purpose	
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>.

Secure Association Establishment for PKM-EAP

Jeff Mandin
Streetwaves Networking

1 Background

PKM-EAP provides a mechanism by which the BS and MSS can mutually authenticate and establish a shared secret (called the *AAA-key*).

To complete the integration of EAP-based authentication into 802.16e we must define the following:

- establishment and installation of the PKM *Authorization Key (AK)*
- ciphersuite signalling
- provisioning of the static Security Associations to the MSS

2 Summary of Solution

For compatibility with 802.11, we use only 32 bytes of the AAA-Key as our Master Key.

Accepted cryptographic practice strongly discourages direct use of the Master Key as it was previously known by another entity (ie. the Authentication Server). Moreover, it is now necessary for the BS and SS to prove to each other that they possess the shared-secret that the EAP peers negotiated.

Accordingly, we use nonces supplied by the BS and SS - together with their MAC addresses and the original Master Key – to derive an AK by way of a pseudo-random function. The security capabilities of the SS and the BS-provisioned Security Association descriptors, are piggybacked onto the exchange.

The last 2 of the of the 3-messages in the exchange are protected by an HMAC-digest.

3 Specific text changes

[7.2.1.2 – replace the final 2 paragraphs ie."The final steps of the authorization flow …" with the following]

The final steps of the authorization flow:

1) The BS and SS each derive the *EAP Master Key* from the AAA-Key. The EAP Master Key is derived simply the taking the 32 lowest order octets of the AAA-Key.

2) BS sends the EAP-Establish-Key-Request PKM message (including a 32-byte nonce) to the SS. The SS then generates its own 32-byte nonce, and derives a *Transient Key (TK)* as follows:

TK = PRF-384(*EAP Master Key*, "Pairwise key expansion",
Min(*BSId*, *SSId*) |
Max(*BSId*, *SSId*) |

Min(*BS-Generated-Nonce*, *SS-Generated-Nonce*) |
Max(*BS-Generated-Nonce*, *SS-Generated-Nonce*))

where

```
PRF-384 ($K$, $A$, $B$) :=

    for $i$ = 0 to 8 do
        $R = R$ | HMAC-SHA-1($K$, $A$ |0 |$B$ | $I$ )

    return LeastSignificant-384-bits($R$).
```

and "|" denotes bitstring concatenation.

The SS then derives *Key Confirmation Key (KCK)* and *Authorization Key (AK)* as follows:

KCK = bits 0-127 (ie. lowest order) of the TK

AK = bits 224-383 of the TK

The SS can attempt to use a cached or handover-transferred Master Key and avoid a full reauthentication. To do this, it sends EAP-Establish-Key-Request specifying the PMKID attribute, which identifies by name the Master Key that the SS should use for AK establishment if it also has the MK cached.

3) SS sends the EAP-Establish-Key-Reply PKM message (including the 32-byte nonce that it used to derive TK) to the BS. EAP-Establish-Key-Reply includes an HMAC Tuple TLV, which must be calculated using the KCK derived above.

   Upon receipt of the EAP-Establish-Key-Reply, the BS computes the TK, KCK, and AK as above. BS then validates the HMAC Tuple. If the HMAC tuple is incorrect, BS discards the message without responding.

   If the SS elects not to proceed with key establishment (eg. the EAP-Establish-key-request specified an unknown PMKID), the SS sends EAP-Establish-Key-Reject instead.

4) BS sends the EAP-Establish-Key-Confirm PKM message to supply the SS with its SA information and activate the AK.

5)

[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	EAP Establish-Key Request	PKM-RSP
16	EAP Establish-Key Reply	PKM-REQ
17	EAP Establish-Key Reject	PKM-REQ
18	EAP Establish-Key Confirm	PKM-RSP
19 ~ 255	reserved	

[Add section 6.4.2.3.9.12 EAP Establish-Key Request message]

The BS transmits the EAP Establish-Key Request message as the first step in the 4-step sequence of establishing an AK after EAP-based authentication.

.

Code : 15

Its attributes are shown in Table xx.

Table xx EAP Establish-Key Request attributes

Attribute	Contents
EAP-Master-Key-Id (optional)	A unique handle for the Master Key supplied by the EAP exchange. For use after handover or drop/reentry situations when a BS believes that it has a PMK for the SS and can proceed immediately to the Establish/Install Key phase. Derivation of the Master Key Id is described in x.x
Nonce	A fresh, randomly generated bit string

[Add section 6.4.2.3.9.13 EAP Establish-Key Reply message]

The SS transmits the EAP Establish-Key Request message as the second step in the 4-step sequence of establishing an AK after EAP-based authentication.

Code : 16

Its attributes are shown in Table xx.

Table xx EAP Establish-Key Reply attributes

Attribute	Contents
Nonce	A fresh, randomly generated bit string
Security-Capabilities	Describes SS's security and ciphersuite capabilities
Primary SAID	SS's primary SAID (equal to the Basic CID)
HMAC-Tuple	The cryptographic hash for the message. The key used to generate the hash is the KCK (key confirmation key) as described in xx

[Add section 6.4.2.3.9.13 EAP Establish-Key Reject message]

The SS transmits the EAP Establish-Key Reject message in order to reject a received Establish-Key Request.

Code : 17

Its attributes are shown in Table xx.

Attribute	Contents
Reject Reason	1 – Unrecognized PKMID

[Add section 6.4.2.3.9.12 EAP Establish-Key-Confirm message]

The BS transmits the EAP Establish-Key-Confirm message as the third step in the 4-step sequence of establishing an AK after EAP-based authentication.

.

Code : 18

Its attributes are shown in Table xx.

Table xx EAP Establish-Key-Confirm attributes

Attribute	Contents
Nonce	Same value as in the Establish–Key Request
Key-Sequence-Number	Sequence Number for established AK
(one or more) SA-descriptors	Each Compound SA-Descriptor attribute specifies an SAID and additional properties of the SA
HMAC-Tuple	The cryptographic hash for the message. The key used to generate the hash is the KCK (key confirmation key) as described in xx

[Section 11.2 Add to Table 282] PKM Attribute types]

Type	**PKM Attributes**

29	EAP-Master-Key-Id
30	Nonce

[Add section 11.2.19 and push down current section with that number] **EAP-Master-Key-Id**

Description : A quantity derived by the Base Station which identifies the 32-octet shared-secret Master Key that results from an EAP exchange. A BS computes the EAP-master-key-Id following EAP exchange success using the following formula:

EAP-Master-Key-Id = HMAC-SHA1-128 (MK, "MK Name" || BSId || SSId)

Where || denotes string concatenation

Type	Length	Value (string)
29	22	Master Key Id

[Add section 11.2.20] **Nonce**

Description : A quantity used to protect message exchanges from Replay Attack. As always, values for nonces should be generated using reliable random or pseudo-random generators.

Type	Length	Value (string)
30	32	Randomly generated value