Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**EAP Keying for PKMv2**
Date Submitted	**2004-08-31**
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:	Recirc #14c for review of IEEE P802.16e/D4-2004
Abstract	Revision of EAP keying protocol
Purpose	Adoption and inclusion in 802.16e specification
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>.

EAP Keying for PKMv2[1]

Jeff Mandin
Streetwaves Networking

1 Background

After the completion of the opaque EAP message exchanges between the Subscriber Station (SS) and AAA-server, and transport of the AAA-key/Pairwise Master Key (PMK) from the AAA-server to the Base Station (BS), the BS and SS must:

- Perform mutual authentication
- Establish an Authorization Key (AK) on the basis of the shared secret PMK. The AK is used:
    - to derive a hash key that's used for link management messages
    - to derive a Key Encryption Key for transporting the actual traffic encryption keys
- negotiate cryptographic capabilities and distribute an SAID list to the SS

We employ a 3-way handshake that is a profile of Authenticated Key Exchange (AKEP2). Authenticated Key Exchange has been proven secure under the Bellare-Rogaway model[2].

The solution described here is superior to the PKMv1 key exchange in the following respects:

- The BS and SS nonces are not used in derivation of the KCK, as this undermines the cryptographic assumptions needed for provable security
- Resistance to the DoS attack claimed in http://www.drizzle.com/~aboba/IEEE/11-04-0497-00-000i-1-message-attack-4-way-handshake.doc
- The keying statemachine is more thoroughly and more correctly described, and an error in the statemachine has been rectified

[1] Cryptographic review and other valuable advice was generously provided by Prof. Amir Herzberg of Bar-Ilan University.

[2] Bellare, M. , Rogaway, P. "Entity Authentication and Key Distribution" 1993 http://www-cse.ucsd.edu/users/mihir/papers/eakd.pdf

2 Summary of Solution

Notation:

Random$_{BS}$ - random value chosen by the BS (once per protocol run)

Random$_{SS}$ - random value chosen by the SS (once per protocol run)

SeqNum – Monotonically increasing Sequence Number (never reused during PMK lifetime)

PMKId – The handle for the shared secret PMK, computed after what is considered to be successful EAP authentication using the following formula:

PMKId = Dot16PRF-128(PMK, "PMK Name" | BsMacAddr | SsMacAddr)

KCK – Key Confirmation Key. Computed by the following formula:

KCK = Dot16PRF-128 (PMK, “802.16 Authenticated Key Exchange KCK” | SeqNum | BsId | SSId)

PreAK – Value used in derivation of Authorization Key. Computed by the following formula:

PreAK = Dot16PRF-128 (PMK, “802.16 PreAK” | SeqNum | BsId | SSId)

MAC$_{Kck}$() – Either AES-OMAC-128() or SHA1-HMAC-128(). Keyed by KCK

Protocol:

1. BS -> SS: *SsId, PMKId, SeqNum, Random$_{BS,}$ MAC$_{Kck}$(SsId | SeqNum | Random$_{BS}$)*

2. SS -> BS: *SsId, BsId, SeqNum, Random$_{SS,}$ Random$_{BS,}$ MAC$_{Kck}$(SsId | BsId | SeqNum | Random$_{SS}$ | Random$_{BS}$)*

3. BS-> SS: *BsId, SeqNum, Random$_{SS,}$ MAC$_{Kck}$(BsId | SeqNum | Random$_{SS}$)*

4. Assuming that BS and SS have decided to accept based on correct output of previous steps, BS and SS each derive:

AK = Dot16PRF (PreAK | RandomSS | RandomBS | “802.16 AK Derivation”)

3 Changes to 802.16e D4 text

[add entries to Tables 37c, d, e, and f:]

SequenceNumber | counter identifying a sequence of key establishment messages
BsMacAddr |
SsMacAddr |

[Table 37c change:]

Change Nonce to BS_RANDOM

[Table 37d, change:]

Change "Nonce" to SS_RANDOM

Add entry:
- BS_RANDOM | Random bit string previously supplied by the key establishment peer

[table 37fe]

- Change "unrecognized MKID" to "parameter error"

[Add section 7.8.1:]

7.8.1 PKMv2 EAP-Establish-Key sequence

Following the successful completion of the opaque EAP message exchanges between the Subscriber Station (SS) and AAA-server, and transport of the AAA-key/Pairwise Master Key (PMK) from the AAA-server to the Base Station (BS), the BS and SS still need to:

- Perform mutual authentication on the basis of the shared secret PMK
- Establish an Authorization Key (AK) on the basis of the shared secret PMK

Before the EAP-Establish-Key sequence begins, the BS and SS shall both derive a PMK Identifier (PMKId), Key Confirmation Key (KCK), and "Pre-AK" as follows:

PMKId = Dot16PRF-128(PMK, | BsMacAddr | SsMacAddr | "PMK Name Derivation")

KCK = Dot16PRF-128 (PMK, BsMacAddr | SsMacAddr | SeqNum |
"802.16 Authenticated Key Exchange KCK Derivation")

PreAK = Dot16PRF-128(PMK, BsMacAddr | SsMacAddr | SeqNum | "802.16 PreAK")

The BS and SS each maintain a SequenceNumber which is associated with the exchange and hence with the AK. After successful completion of the EAP exchange, BS and SS both set SequenceNumber to 0.

The EAP-Establish-Key sequence proceeds as follows:

1. The BS shall select a random number (BsNonce), and send EAP-Establish-Key-Request to the MSS (using the KCK as the key for the HMAC-digest).

If the BS does not receive EAP-Establish-Key-Reply from the MSS within *AKTimer1*, it shall resend the request. The BS may resend the EAP-Establish-Key-Request up to *EstablishKeyRequestMaxResends* times. If the BS reaches its maximum number of resends, it shall discard the PMK and reinitiate EAP authentication as described in xx.

2. Upon receipt of EAP-Establish-Key-Request, an MSS shall confirm that the supplied PMKId refers to a PMK that it has available. If the PMKId is unrecognized, the MSS shall ignor the message.

   The MSS shall confirm that the value of the SequenceNumber is equal to or greater than its own maintained SequenceNumber value. If the SequenceNumber is less than the expected SequeneNumber, the MSS shall ignor the message.

   The MSS shall verify the HMAC digest. If the HMAC digest is invalid, the MSS shall ignor the message.

   Upon validation (by the above criteria) of the received EAP-Establish-Key-Request, the SS shall confirm that the SsMacAddr and BsMacAddr are correct. If either is incorrect, the SS shall send EAP-Establish-Key-Reject to the BS. Otherwise the key establishment sequence may proceed.

3. To continue the key establishment sequence, an SS shall set its current SequenceNumber to the value received in EAP-Establish-Key-Request. The SS shall then select and store its own random number (SsNonce) and send EAP-Establish-Key-Reply to the BS.

   If the SS does not receive EAP-Establish-Key-Confirm from the BS within *AKTimer2*, it shall resend the reply. The BS may resend the EAP-Establish-Key-Request up to *EstablishKeyReplyMaxResends* times. If the SS reaches its maximum number of resends, it shall discard the PMK and drop the connection to the BS.

.

4. Upon receipt of EAP-Establish-Key-Reply, a BS shall confirm that the value of the SequenceNumber is equal to or greater than its own maintained SequenceNumber value. If the SequenceNumber is less than the current SequenceNumber, the MSS shall ignor the message.

   The BS shall verify the HMAC digest. If the HMAC digest is invalid, the BS shall ignor the message.

   If the value of the SequenceNumber in a validated (by the above criteria) EAP-Establish-Key-Reply is greater than the current expected SequenceNumber, the BS shall discard the PMK and reinitiate EAP authentication as described in xx.

   The BS shall confirm that the BsNonce value in a validated EAP-Establish-Key-Reply is the same as the one that it sent in EAP-Establish-Key-Request and that the BsMacAddr and SsMacAddr fields are correct. If any of these conditions are false, the BS shall increment its SequenceNumber by one, and increment its EstablishKeyRetryCounter by one. If the EstablishKeyRetryCounter reaches *EstablishKeyMaxRetries*, the BS shall discard the PMK and reinitiate EAP authentication as described in xx. Otherwise, the BS restarts the key establishment sequence by sending a new EAP-Establish-Key-Request.

5. To continue the key establishment sequence, the BS sends EAP-Establish-Key-Confirm to the MSS.

   The BS concludes that the key establishment has been successful, and sets:

AK = Dot16PRF (PreAK, RandomSS | RandomBS | “802.16 AK Derivation”)

If the BS receives another validated EAP-Establish-Key-Reply containing the same sequence number, it shall reply earnestly until *AKConfirmTimer* expires. After the timer expires, the BS shall ignor such messages.

6. Upon receipt of EAP-Establish-Key-Confirm, an SS shall confirm that the value of the SequenceNumber is equal to or greater than its own maintained SequenceNumber value. If the SequenceNumber is less than the current SequenceNumber, the SS shall ignor the message.

   The MSS shall verify the HMAC digest. If the HMAC digest is invalid, the SS shall ignor the message.

   If the value of the SequenceNumber in a validated (by the above criteria) EAP-Establish-Key-Confirm is greater than the current expected SequenceNumber, the MSS shall transmit EAP-Establish-Key-Reject to the BS carrying the invalid SequenceNumber.

   Otherwise, the MSS concludes that the key establishment has been successful, and sets the AK according to the same formula as the BS above.

7.8.1.1 Handling of EAP-Establish Key-Reject

Upon receipt of EAP-Establish-Key-Reject, a BS shall verify the HMAC digest. If the HMAC digest is invalid, the BS shall ignor the message. Otherwise, the BS shall accept the EAP-Establish-Key-Reject, increment its SequenceNumber, and restart the key establishment sequence.