Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	PKM version 2
Date Submitted	**2004-06-26**
Source(s)	Jeff Mandin jeff@streetwaves-networks.com
Re:	Security Adhoc
Abstract	Specification of Privacy Key Management version 2
Purpose	Specification of Privacy Key Management version 2
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>.

Privacy Key Management version 2

Jeff Mandin

1 Overview

This document is the strawman for the security adhoc contribution on PKMv2.

2 General Guidelines

PKMv2 and associated mechanisms:

- must not impose unreasonable compute requirements
- must work with existing hardware
- must include no frame format or GMH changes
- must work on existing hardware
-

PKMv2 should reflect "the way it's done in 802.16":

* It should fit into existing capability negotiation mechanisms
* It should be backwards compatible with the existing PKMv1
* It should map to approximately the same functions as PKMv1. I.E. solve the same problem, only better.

3 PKMv2 Functionality

3.1 MAC Management message protection

1. Forgery Protection

   In the current 802.16d/e drafts, not all MAC management messages include a Message Authentication Code TLV for forgery protection.

   We should include forgery protection for MAC management messages whenever possible.

2. Replay Attack Protection

   Currently an attacker can capture and replay a packet such as HO-Ind or Location-Update, inducing undesirable consequences. Replaying PKM-Key-Rsp could enable an attacker to force an SS to use an old (and possibly cracked) TEK.

   We must include replay prevention for all authenticated MAC management messages.

3. Message Authentication Code

   Support AES-MAC so as to enable use of implementation or hardware accelerators already necessary for AES-CCM data msg security.

.

3.2 Authentication

1. Full state machine definitions for the EAP-based authentication and the AK establishment handshake.

   Full state machine definitions, with description of timers etc. is a lacuna of the current document.

2. Preauthentication

   We must facilitate authentication with a potential target BS via the backbone.

   Some members note that the Key Caching mechanism to appear in D3 of 802.16e facilitates preauthentication.

3.3 Key Management

1. Broadcast Channel support

   When data is sent over the Broadcast Channel, precisely the same data must be transmitted by each BS using PHY timing info as supplied by the Multimedia Broadcast Server.

   Members of the adhoc have suggested 2 broad approaches to security requirements for the broadcast channel.

   - "**Approach A**": Since the data must be identically transmitted by all neighboring BSes, there should be no 802.16-MAC-layer confidentiality/authenticity employed for the broadcast service. Encryption/message authentication.should be performed at the data-originating MBS (though of course it can be transparently offloaded to another entity)

   -"**Approach B**": The requirement for identical transmissions from all BSes creates new requirements for MAC-layer security which must be built into PKMv2. Specifically there is now a requirement that the BSes synchronize their encryption states, so that identical keys, IVs etc. are used across all BSes at all times. Within Approach B, different mechanisms have been suggested as to how to accomplish the synchronization of keys etc.

2. Unicast TEK establishment

The unicast TEK establishment procedure should include liveness/freshness guarantees and otherwise pass muster.

3. Distribution of Multicast TEKs

   A more-efficient-than-currently method of distributing multicast TEKs is desirable ie. one which enables an BS to “push” the TEK to the SS. Ideally the “push” would be done only once for the entire multicast group.

3.4 Protection of the EAP authentication exchange

1. Protected channel for EAP messages

   The EAP messages exchange enables the BS and SS to authenticate (or re-authenticate) each other. However, even when the specific EAP methods are considered secure, there exists the possibility of Denial-of-service attack via forged EAP messages.

   As partial protection, we should use available techniques for protection against forged EAP messages. Available techniques include:

   a) using an unauthenticated tunnel for the EAP authentication exchange to ensure that only one peer is doing the talking

   b) using an existing security association during reauthentication

2. Encryption of the EAP-Transfer MAC message payload

   Some members of the adhoc have proposed encryption of the EAP-Transfer MAC messages payload using the available security associations described in item 1) directly above.

   According to the proposal, encryption of the payload is desirable as an alternative to establishing an encrypted tunnel as part of using a tunnelled EAP method such as PEAP.

   Other members believe that the encryption is unnecessary and could encourage insecure use of EAP.

4 Changes to 802.16e D3 text

[Modify section 7.1.2 as follows:]

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.~~ PKM version 2 is for mobility-supporting environments, and uses EAP [IETF RFC 3579] to perform authentication. The legacy PKM version 1 uses X.509 digital certificates [IETF RFC 3280] together with RSA public-key encryption algorithm [PKCS #1]

~~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.~~

~~1In security parlance, confidentiality = privacy + authenticity~~

~~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.

[Delete section 7.1.3]

[Modify section currently titled 7.1.4 as follows:]

7.1.4 Security associations

A *Security Association* (SA) is the set of security information a BS and one or more of its client SSs share in order to support secure communications across the IEEE Std 802.16 network. Three types of SAs are defined: *Primary, Static, and Dynamic*. Each SS establishes a Primary Security association during the SS initialization process. Static SAs are provisioned within the BS. Dynamic SAs are established and eliminated, on the fly, in response to the initiation and termination of specific service flows. Both Static and Dynamic SAs can be shared by multiple SSs.

An SA's shared information shall include the Cryptographic Suite employed within the SA. The shared information may include TEKs and Initialization Vectors. The exact content of the SA is dependent on the SA's Cryptographic Suite.

SAs are identified using SAIDs.

Each SS shall establish an exclusive Primary SA with its BS. The SAID of any SS's Primary SA shall be equal to the Basic CID of that SS.

Using the PKM protocol, an SS ~~requests from its BS an~~ receives or establishes an SA's keying material. The BS shall ensure that each
client SS only has access to the SAs it is authorized to access.

~~An SA's keying material [e.g., Data Encryption Standard (DES) key and CBC Initialization Vector] has a limited lifetime. When the BS delivers SA keying material to an SS, it also provides the SS with that material's remaining lifetime. It is the responsibility of the SS to request new keying material from the BS before the set of keying material that the SS currently holds expires at the BS. Should the current keying material expire before a new set of keying material is received, the SS shall perform network entry as described in 6.4.9.~~

~~In certain Cryptographic Suites, key lifetime may be limited by the exhaustion rate of a number space [e.g. the PN (Packet Number) in AES-CCM mode]. In this case, the key ends either at the expiry of the key lifetime or the exhaustion of the number space, which ever is earliest. Note that in this case, security is not determined by the key lifetime.~~

[Rename Section 7.2 to "PKM protocol version 1 "]

[delete section 7.1.3.2]

[Insert new section 7.3]

7.3 PKM protocol version 2

7.3.1 Overview of BS-MSS authentication and Authorization Key establishment

7.3.1.1 EAP-based Authentication

PKMv2 uses Extensible Authentication Protocol [IETF RFC 3748] in conjunction with a vendor-selected standardized EAP Method (eg. EAP-TLS [IETF RFC 2716]) to perform authenticate the identities of the BS and MSS. 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).

This specification does not mandate the particular credentials and EAP method that are to be used, but these should be selected according to the criteria discussed in Annex F

In PKMv2, EAP messages are transported between the BS and MSS in PKM-REQ/PKM-RSP MAC management messages which carry the code types EAP-Transfer-Request and EAP-Transfer-Reply.

7.3.1.1.2 Network model for EAP-based Authentication

As described in [IETF RFC 3748], an EAP-based authentication involves three entities:

a) Peer (ie. the MSS in the case of 802.16e)

b) Authenticator (the BS)

c) Backend Authentication Server (though the Authentication Server function can be colocated in the BS, more typically there is an external AAA server that communicates with the BS via an EAP-enabled protocol such as RADIUS or DIAMETER). Operations between the BS and Backend Authentication Server are out of the scope of this specification.

The PKMv2-enabled MAC layer supplies a control plane logical interface to an EAP logical entity. The EAP logical entity performs the following functions:

a) sends and receives EAP messages with its peer EAP entity (ie. at the BS or SS) on the other end of the 802.16e air link during authentication

b) Notifies the MAC layer of the success or failure of authentication

c) Supplies keying material to the MAC layer

Figure 130 compares the PKMv1 and PKMv2 control plane network models.

[Insert figure 130o –
change “Basic Privacy Layer” to “PKMv1 Privacy sublayer”]
change “Extended Privacy Layer” to “PKMv2-enabled Privacy sublayer”
correct spelling of “authentication”]

7.3.1.2 Authentication Exchanges on Initial Network Entry

During initial network entry, the BS and MSS determine capabilities using SBC-REQ/SBC-RSP as described in xx.

Upon successful negotiation of PKMv2, the EAP entity on the BS begins sending EAP messages to the MSS in order to authenticate its identity (except in the case where the BS has a PMK cached for the MSS, as described in 7.3.1.4).

The Privacy sublayer at the BS encapsulates the messages it receives from its EAP entity into EAP-Transfer-Reply PKM-RSP PDUs and transmits them to the MSS on the Basic CID. Similarly, the privacy sublayer on the MSS unpacks the EAP messages that it receives and passes them to its EAP entity. Finally, the MSS encapsulates EAP messages from its EAP entity into EAP-Transfer-Request PKM-REQ PDUs for transmission on the Basic CID.

EAP messages are passively carried by the Privacy Sublayer and do not affect MAC state.

The exchange of encapsulated EAP messages continues until either:

- EAP-based authentication is successful

- EAP-based authentication fails due to inadequate credentials or other errors

7.3.1.2.1 Successful Authentication

Upon successful authentication, bothe EAP entities (ie. at BS and MSS) supply the Privacy Sublayer with a 256-bit Shared Secret called the **Pairwise Master Key** (PMK). The PMK is computed or established by the two EAP entities during the execution of the EAP method.

Upon reception of the PMK from the EAP entity, the Privacy Layer must compute a “handle” for the PMK.

The handle is called the **PMKId**, and is derived using the following formula:

PMKId = HMAC-SHA1-128 (PMK, "MK Name" || BSId || SSId)

where || denotes string concatenation

7.3.1.2.2 Authentication Failure

When the EAP entity on the BS determines that authentication has failed (whether due to an unacceptable credential, message timeouts, or what have you), it instructs the MAC layer to deactivate the Basic CIDs of the MSS. Hence the MSS must reinitiate Ranging if it wishes to try again to enter the network.

7.3.1.3 MSS Authorization

After the BS and attached MSS have established a PMK (whether the PMK is newly established or cached), the BS seeks to *authorize* the MSS, to allow it to complete network entry. *Authorization* is the process of the BS and MSS reconfirming each other’s identity, and establishment of a 160-bit **Authorization Key** (AK).

The Authorization exchange is as follows:

1) 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*))

1)

where

```
PRF-384 (K, A, B) :=
    for i = 0 to 3 do
        R = R | HMAC-SHA-1(K, A | 0 | B | I )
    return LeastSignificant-384-bits(R).
```

and “|” denotes bitstring concatenation.

2) 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

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.

4) 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.

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

7.3.1.4 PMK Caching

A BS may maintain a cache of PMKs and their associated SS_IDs. The BS should cache PMKs associated with MSSes that are expected to arrive due to handover (see 7.3.1.4), and may also cache the PMKs belonging to MSSes that have recently disconnected.

Upon successful ranging by an MSS, a BS may determine that it has a PMK associated with the claimed MAC address in its PMK cache. It may then attempt to use the cached PMK to establish an AK, and avoid a full authentication.

To do this, the BS initiates *Fast MSS authorization* by sending **EAP-Establish-Key-Request** including the PMKID attribute, which identifies by name the Pairwise Master Key that the SS should use for authorization if it also has the PMK cached: If the MSS does not have the PMK cached, it responds with **EAP-Establish-Key-Reject**, and the BS should initiate full authentication.

7.3.1.5 Authentication following Handover

Following handover, the new BS will attempt to invoke Fast MSS authorization. If Fast MSS authorization fails, it should initiate full authentication.

7.3.2 Authentication and Authorization Pairwise State Machine

The State Machine is pairwise – ie. a BS or MSS can support multiple state machines, each of which pertains to its authentication/authorization state to a particular communication partner.

7.3.2.1 States

1. Initializing
2. unauthenticated
3. Authenticated
4. AuthenticationRejected
5. AuthorizationInProgress
6. Authorized
7. Departed

7.3.2.2 Messages

These are the PKM msgs.

7.3.2.3 Events

- AuthenticationSuccess signal from EAP entity
- AuthenticationFailure signal from EAP entity

7.3.2.4 Parameters

- PartnerId (ie. BS_Id or MSS_Id)
- PMK
- Authorization exchange timers

7.3.2.5 Actions

These accompany state transitions:

1) Initializing --> unauthenticated
   a. Send “connection up” indication to EAP entity

2) Unauthenticated --> departed
   a. Send “connection down” indication to EAP entity

3) Unauthenticated --> AuthenticationRejected
   a. Take down the connection/CID etc.

[Add Annex F]

Annex F – EAP method requirements

EAP methods used for 802.16e should be selected with awareness of the security issues described in [IETF RFC 3748] section 7. The considerations described in [IETF Draft “EAP Method Requirements for Wireless LANs”] apply for an 802.16e environment also.