Project	IEEE 802.16 Broadband Wireless Access Working Group <**http://ieee802.org/16**>
Title	PKMv2 considerations
Date Submitted	2006-03-07
Source(s)	Phillip Barber Chair, NetMan TG Huawei [mailto:pbarber@futurewei.com]
Re:	PKMv2 considerations
Abstract	PKMv2 considerations
Purpose	PKMv2 considerations
Notice	
Release	
Patent Policy and Procedures	

PKMv2 considerations

Phillip Barber

Behavior MUST BE KNOWN!

Security vulnerability if BS forces re-auth on AK invalid detection. Attacking MS could force a legit MS to re-auth by simply sending any message with an invalid CMAC. Solution is 'silent discard'. Silent Discard has problems:

- MS unaware that MAC mgmt messages are being discarded due to invalid CMAC; MS will assume the BS just is not receiving the messages; MS re-transmit of messages with continued bad CMACs will result; latency problem.
- MS secure re-auth—inclusion of invalid CMAC in re-auth request. What happens? How do we keep attacking MS from forcing unsecure reauth that messes with MS-to-BS State synchronization?
- Need different rules during MS-to-Target BS handover, re-entry than during MS-to-Serving BS communications. During handover communications, invalid AK/CMAC forces full authentication but MUST NOT affect existing, stored and/or active security associations for MS-to-Serving BS and stored SA on Target BS; at least not until handover, network re-entry completed and MS achieves Normal Operation at new location. Presents opportunity for MS-to- BS on the network Authentication State and context disjunction.

Some relevant portions of the PKMv2 architecture section with some notes:

From 802.16e-2005, page 285
7.2.2.2.11 Maintenance of PMK and AK
The BS and SS maintain cached PMK and AK as follows:

a) PMK caching
An SS caches a PMK upon successful EAP authentication. An Authenticator caches a PMK upon its receipt via the AAA protocol. Upon caching a new PMK for a particular SS, an Authenticator shall delete any PMK for that SS (as well as all associated AKs).
For the case of reauthentication, deletion of old PMKs at Authenticator and SS is accomplished via the switchover mechanism described in this subclause using the messages in 6.3.2.3.9.20.
The Authenticator and SS will additionally delete PMKs and/or associated AKs in various situations—including lifetime expiration, reauthentication, and reclamation of memory resources, or as the result of other mechanisms beyond the scope of this specification.
In the case of re-authentication, the older PMK and its AKs shall be deleted by the SS after verifying the HMAC or CMAC of the PKMv2 SA-TEK challenge message and the BS after verifying the HMAC/CMAC of the PKMv2 SA-TEK request message.

b) AK activation and deactivation
Successful completion of the 3-way SA-TEK handshake causes the activation of all the AKs associated with the new PMK (i.e., all AKs on BSs associated with the current authenticator will be active).

If the packet counter belonging to a short HMAC or a CMAC key reaches its maximum value, the associated AK becomes permanently deactivated.
The BS and SS must maintain the AK context (i.e., replay counters etc.) as long as they retain the AK. **[Note this requirement on PN context requirements; tied to keeping AK]**

7.2.2.2.12 PKMv2 PMK and AK switching methods
Once the PKMv2 SA-TEK 3-way handshake begins, the BS and SS shall use the new AK matching the new PMK context for the 3-way handshake messages. Other messages shall continue to use the old AK until the 3-way handshake completes successfully. Upon successful completion of the 3-way handshake, all messages shall use the new AK. **[What happens if the TEK 3-way handshake never completes?]**

The old AK matching the old PMK context may be used for receiving packets before the “frame number” attribute specified in PKMv2 SA-TEK-response message.

7.2.2.4.1 AK context
The PMK key has two phases of lifetime: the first begins at PMK creation and the second begins after validation by the 3-way handshake.

The phases ensure that when the PMK is created it will be defined with the PMK or PAK pre-handshake lifetime and after successful 3-way handshake, this lifetime may be enlarged using the PMK lifetime TLV within the 3-way handshake.

If the cached AK and associated context is lost by either BS or SS **[State problem; how does the other side know that AK context has been lost on the other side?]**, no new AKs can be derived from this PMK on handover. **[Note that this means MS MUST CACHE any AK that it creates; it cannot re-create it later][Note that CMAC_PN_* is in Table 133a, so loss of CMAC_PN_* requires flushing the AK and other context, which in turn re-quires re-authentication for a new PMK and AK]**

Cached AKs that were derived from the PMK can continue to be used in HO.

Reauthentication is required to obtain a new PMK so as to derive new AKs.

The AK context is described in Table 133a.
.
.
.
7.5.4.4.1 Calculation of CMAC Value
.
.
.
The CMAC Packet Number Counter (CMAC_PN_*) is a 4-byte sequential counter that is incremented in the context of UL messages by the SS, and in the context of DL messages by the BS,. The BS will also maintain a separate CMAC_PN_* for multicast packets per each GSA and increment that counter in the context of each multicast packet from the group. For MAC messages that have no CID e.g., RNG-REQ message, the CMAC_PN_* context will be the same as used on the basic CID. If basic CID is unknown (e.g., in network reentry situation) then CID 0 should be used.

The CMAC Packet Number Counter, CMAC_PN_*, is part of the CMAC security context and must be unique for each MAC management message with the CMAC tuple or digest. Any tuple value of {CMAC_PN_*, AK} shall not be used more than once. The reauthentication process should be initiated (by BS or SS) to establish a new AK before the CMAC_PN_* reaches the end of its number space.

.

.

.

7.7 Pre-Authentication

In anticipation of a handover, an MS may seek to use pre-authentication to facilitate an accelerated reentry at a particular target BS.

Pre-authentication results in establishment of an Authorization Key (with a unique AK Name) in the MS and target BS. The specific mechanism for Pre-authentication is out of the scope of this specification. **[Note that this can result in State synchronization problem between MS and BS, depending on the implementation]**

.

.

.

7.8.2 BS and SS RSA mutual authentication and AK exchange overview

.

.

.

After achieving initial authorization, an SS periodically seeks reauthorization with the BS; reauthorization is also managed by the SS’s PKMv2 Authorization state machine **[Note that PKMv2 Authorization State Machine is not defined]**. An SS must maintain its authorization status with the BS in order to be able to refresh aging TEKs and GTEKs. **[So loss of AK context means cannot update TEK; what happens when SS loses AK for a different BS]**