Project	IEEE 802.16 Broadband Wireless Access Working Group <*http://ieee802.org/16*>	
Title	Auth FSM	
Date Submitted	**2006-09-22**	
Source(s)	Avishay Shraga_ _____Intel corporation Phillip Barber Huawei Joseph Schumacher Motorola Tricci So Nortel	voice: +972-3-920-5763 avishay.shraga@intel.com
Re:	Missing Auth FSM in 802.16e	
Abstract	The contribution defines the Authentication FSM for PKMv2 EAP only authentication. Including States, Events and Transitions	
Purpose	Resolve ambiguity in definition of Auth FSM	
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>.	

Single EAP Authentication FSM

Intel Corporation

1. Background

The standard defines the FSM for TEK exchange but does not define the FSM for Authentication.
The absence of this FSM makes it very unclear how the authentication and re-authentication are done during initial entry and re-entry from HA HO and from idle.

2. Suggested remedy

Define the Auth FSM for EAP only authentication as negotiated in SBC.
The FSM includes states events and transitions to support initial authentication, re-authentication and re-entry (optimized) for HO and idle.
Fix the action to be done in case of 3-way hand-shake fail for re-authentication.

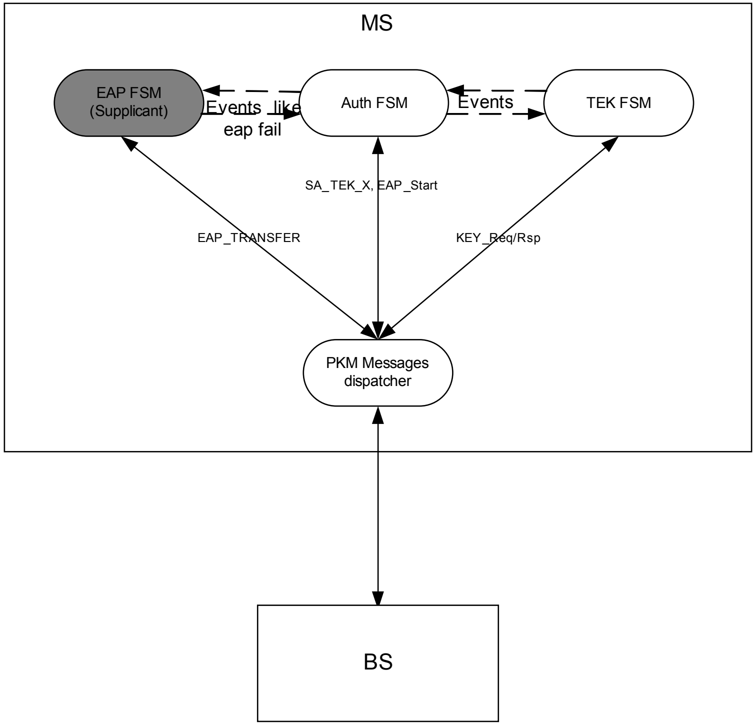
3. Proposed Text Changes

[Please add the following text as new clause number 7.2.2.X in 802.16e-2005]

7.2.2.X **Authentication state machine**

The Authentication state machine for single EAP authentication consists of six states and sixteen events (including receipt of messages and events from other FSMs) that may trigger state transitions. The Authentication state machine is presented in both a state flow diagram (Figure XXX) and a state transition matrix (Table XXX).The state transition matrix shall be used as the definitive specification of protocol actions associated with each state transition.

The Authentication process has 2 phases: EAP phase and 3-way handshake phase (also known as SA_TEK exchange).
The EAP phase is controlled by the EAP_FSM as defined in RFC3748 and RFC4173 and it is out of scope in this standard.
The Auth_FSM is responsible for all PKM phase but the actual EAP exchange and communicates with other FSMs in the system using events.
The relationships between the security related FSMs in the system is as described in the diagram:



Through operation of a Authentication state machine, the MS attempts to get authenticated with the NW, maintain this authentication and support Authentication context switching for HO and Idle situations.
The state machine takes care of getting authenticated with the NW, ensuring re-authentication will occur before authentication expires and support key derivations according to support optimized re-entry for HO and idle.
The optimized re-entry support is done in a special state in which the NW connection is suspended and therefore re-authentication can’t occur, the triggers for re-authentication continue to work in this state but the initiation is done only after returning to an authenticated state.

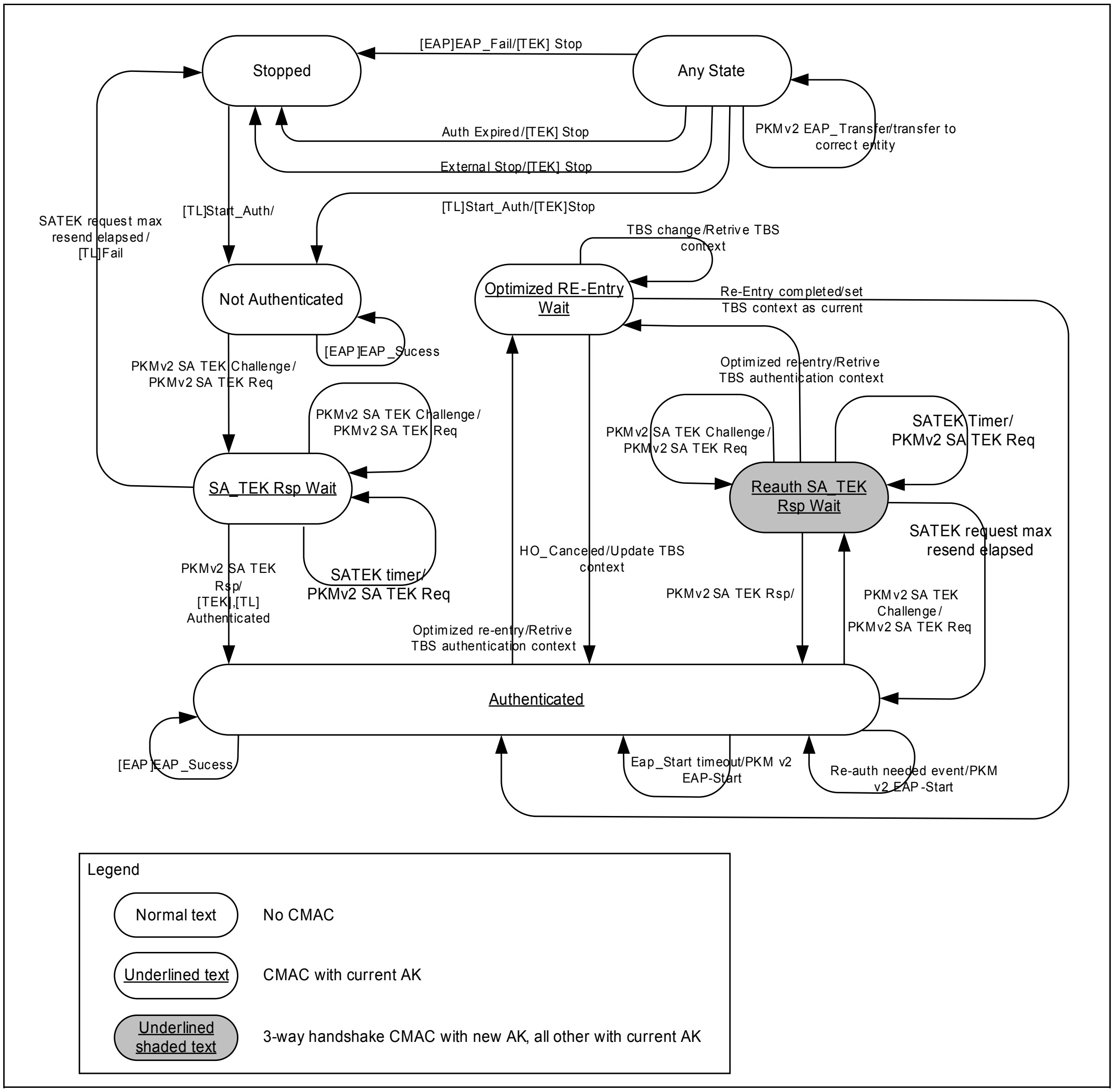

Stopped
[EAP]EAP_Fail/[TEK] Stop
Any State
Auth Expired/[TEK] Stop
External Stop/[TEK] Stop
PKMv2 EAP_Transfer/transfer to correct entity
[TL]Start_Auth/[TEK]Stop
[TL]Start_Auth/
SATEK request max resend elapsed/ [TL]Fail
TBS change/Retrive TBS context
Not Authenticated
Optimized RE-Entry Wait
Re-Entry completed/set TBS context as current
[EAP]EAP_Sucess
PKMv2 SA TEK Challenge/ PKMv2 SA TEK Req
Optimized re-entry/Retrive TBS authentication context
PKMv2 SA TEK Challenge/ PKMv2 SA TEK Req
PKMv2 SA TEK Challenge/ PKMv2 SA TEK Req
SATEK Timer/ PKMv2 SA TEK Req
SA_TEK Rsp Wait
Reauth SA_TEK Rsp Wait
SATEK request max resend elapsed
PKMv2 SA TEK Rsp/ [TEK],[TL] Authenticated
SATEK timer/ PKMv2 SA TEK Req
HO_Canceled/Update TBS context
PKMv2 SA TEK Rsp/
PKMv2 SA TEK Challenge/ PKMv2 SA TEK Req
Optimized re-entry/Retrive TBS authentication context
Authenticated
[EAP]EAP_Sucess
Eap_Start timeout/PKM v2 EAP-Start
Re-auth needed event/PKM v2 EAP-Start
Legend
Normal text
No CMAC
Underlined text
CMAC with current AK
Underlined shaded text
3-way handshake CMAC with new AK, all other with current AK

Figure XXX – Authentication State Machine for single EAP

Table YYY – Authentication FSM state transition matrix

State *Event or receive message*	(A) Stopped	(B) Not authenticated	(C) SA_TEK Rsp Wait	(D) Authenticated	(E) Reauth SA_TEK Rsp Wait	(F) Optimized re-entry wait
(1) *Start Auth*	Not authenticated	Not authenticated	Not authenticated	Not authenticated	Not authenticated	Not authenticated
(2) *PKMv2 SA_TEK Challenge*		SA_TEK Rsp Wait	SA_TEK Rsp Wait	Reauth SA_TEK Rsp Wait	Reauth SA_TEK Rsp Wait	
(3) *PKMv2 SA_TEK RSP*			Authenticated		Authenticated	
(4) *EAP Success*		Not authenticated		Authenticated		
(5) *SATEK timer*			SA_TEK Rsp Wait		Reauth SA_TEK Rsp Wait	
(6) *SATEK req max resend elapsed*			Stopped		Authenticated	
(7) *ReAuth needed*				Authenticated		
(8) *optimized re-entry*				optimized re-entry wait	optimized re-entry wait	
(9) *EAP_Start timeout*				Authenticated		
(10) *Re-entry completed*						Authenticated
(11) *HO canceled*						Authenticated
(12) *TBS change*						optimized re-entry wait
(13) *Auth expired*				Stopped	Stopped	Stopped
(14) *EAP fail*	Stopped	Stopped	Stopped	Stopped	Stopped	Stopped
(15) *External*	Stopped	Stopped	Stopped	Stopped	Stopped	Stopped

stop						
(16) *EAP Transfer*		Not authenticated	SA_TEK Rsp Wait	Authenticated	Reauth SA_TEK Rsp Wait	optimized re-entry wait

7.2.2.X.1 States

Stopped: This is the initial state of the FSM, nothing is done in this state.

Not Authenticated: The Auth FSM is not authenticated and waiting for MSK to be received from EAP FSM in order for MAC level authentication to start from BS side by SA_TEK_Challange.
Upon receiving SA_TEK Challenge, it is validated by checking H/CMAC using current AK and silently discarded if not valid.

SA TEK Rsp Wait: The Auth FSM has send SA_TEK_Req and wait for SA_TEK_Rsp, It retry the message if needed as long as SATEK counter not elapsed.
Upon receiving SA_TEK Response, it is validated by checking H/CMAC using current AK and silently discarded if not valid.

Authenticated: Successful authentication finished, the MS derived all keys from MSK, set the valid time of authentication keys and started timers to monitor authentication expiration. In this state, the FSM is ready to receive MSK from EAP FSM and derive keys of re-authentication and to receive MAC level re-auth from BS side (SA_TEK_Challange).
In this state all management messages that defined to be sent with H/CMAC are checked using current AK and silently discarded if MAC is not valid.
In this state the MS may hold two authentication contexts: current and next context created during re-authentication, the current context is deleted in the frame number defined in SA_TEK_RSP and the new one becomes current.

Reauth SA TEK Rsp Wait: The Auth FSM has sent SA_TEK_Req for reentry and wait for SA_TEK_Rsp, It retry the message if needed as long as SATEK counter not elapsed.
In this state there are two authentication contexts: the current which is used for CMAC with all management messages (that needs MAC tuple) and temporary context created after EAP phase finished, the temporary context is used for the re-authentication 3-way handshake exchange and upon successful reception of SA_TEK_RSP it changes from temporary to next context.

Optimized re-entry Wait: In this state the Auth FSM is supporting Auth key derivation for Optimized reentry (for HO or Idle), it stopped all other actions and just give services to the HO FSM for optimized re-entry.
All authentication related timers continue to tick in this state but actions are taken only upon returning to regular “authenticated” state.
In this state all derived AK and context are maintained and cached for as long as they

may be valid in order to protect from replay attack.
Messages in this state are protected with CMAC using AK of current BS re-entry is done with.
Only in this state, in case of CMAC validation fail the message is discarded but an event is sent to top level FSM to notify this event.

7.2.2.X.2 Messages

PKMv2 SA TEK Challenge: first message of MAC level 3-way handshake, send from the BS to MS after EAP authentication has finished, protected by CMAC digest using the key of the last EAP authentication.

PKMv2 SA TEK Request: Second message of MAC level 3-way handshake, send from MS to BS as response to SA TEK Challenge, protected by CMAC digest using the key of the last EAP authentication.

PKMv2 SA TEK Response: Last message of MAC level 3-way handshake, send from BS to MS as response to SA TEK request, protected by CMAC digest using the key of the last EAP authentication.
Receiving this message validate the authentication and order the MS to start using the key in the frame number as included in the message (if frame number TLV is missing, usage immediately)

PKMv2 EAP Start: Message used by MS to ask BS to initiate EAP Authentication, the message should be used by MS if it realize re-authentication is needed, the auth FSM do not receive any acknowledge for this message, the only acknowledge is finish of successful re-authentication (receiving SA_TEK_RSP).

EAP Transfer: This message is bidirectional and used as transport to EAP packets send and received to EAP FSM. Auth FSM does not use this message and just transfer them to and from EAP_FSM.
These messages are send/received unprotected in not authenticated state and protected with H/CMAC in any other states.
The messages are not really part of Auth FSM but since they part of security protocol they are here to emphasize they can be processed in any state but “STOP” and not part of Auth FSM itself.

7.2.2.X.3 Events

Start Auth: send from higher level FSM (NW entry for example) to start the authentication state machine and have it waiting for authentication.
This event can be send anytime from any reason some other FSM decides full NW-entry is needed and therefore Auth FSM should start from initial authentication state.

EAP Success: Send from EAP FSM to notify Auth FSM that EAP success was finished successfully, this event may not arrive it the EAP level message lost in the air link.

SATEK timer: timer event that signal FSM to retry sending the SA_TEK_Request message because the Rsp was not received.

SATEK req max resend elapsed: counter event that signal the FSM that max retry of SA_TEK_Request resend finished and another action should be taken.

ReAuth needed: An internal event of state that can be derived from several sources such as ~~authentication~~ authorization grace time, PN space grace value or other reason that makes authentication close to expiration.

HO re-entry: Send from HO FSM to inform Auth FSM that MS is in HO re-entry phase and FSM should support AK derivation for re-entry and not normal usage of AK.

EAP_Start timeout: Timer event that courses the MS to re-send EAP start in order to ask the BS to start EAP Authentication. This event is used for EAP_Start retry in case re-authentication did not finished successfully after last EAP_Start. This timer is active only after ReAuth needed event occurred.

Re-entry completed: Event from HO FSM to signal Auth FSM that re-entry was finished successfully and Auth can continue it's normal course with AK of new BS

HO canceled: Event from HO FSM to signal Auth FSM that HO was canceled and should continue normal course with original AK.

TBS change: Event from HO FSM to signal that need to switch context to new TBS.

Auth expired: Event that signal auth expired due to timer and need to terminate authentication and connection to BS.

External stop: Event from other FSM to stop Auth FSM and terminate connection with BS.

7.2.2.X.4 Parameters

SATEK timer: timeout value between sending SA_TEK_Request messages

SATEK max resend counter: Counter of number of SA_TEK_Request retries allowed.

EAP_Start timout: Timer between sending EAP_Start messages, since the only meaningful result of EAP_Start is successful re-authentication complete by receiving SA_TEK_RSP, the value of the timer should be larger than the max re-auth time including EAP phase and SA_TEK exchange.

~~Authentication~~ Authorization grace time: period before authentication will be expired when re-authentication attempts should begin to ensure successful re-auth before actual authentication expiration.
This value should be larger that large number of EAP_Start timeout.

PN grace value: A value of PN space that when reached re-auth is needed, this value should be large enough to support message exchange during several re-auth attempts

7.2.2.X.5 Actions

1- A..F: Any State (*Start Auth*)→ not authenticated
a) Enable EAP_Transfer message transfer.
b) Stop all TEK FSMs

2-B: Not Authenticated (*SA_TEK_Challange*)→SA TEK Rsp wait
a) Send SA TEK Req.
b) Start SATEK Timer.
c) Initiate SATEK counter

2-C: SA TEK Rsp wait (*SA_TEK_Challange*)→SA TEK Rsp wait
a) Send SA TEK Req.
b) Start SATEK Timer.
c) Initiate SATEK counter

2-D: SA Authenticated (*SA_TEK_Challange*)→reauth SA TEK Rsp wait
a) Send SA TEK Req.
b) Start SATEK Timer.
c) Initiate SATEK counter

2-E: reauth SA TEK Rsp wait (*SA_TEK_Challange*)→reauth SA TEK Rsp wait
a) Send SA TEK Req.
b) Start SATEK Timer.
c) Initiate SATEK counter

3-C: SA TEK Rsp wait (PKMv2 *SA_TEK_Rsp*)→Authenticated
a) Start TEK FSM
b) Start ~~Authentication~~ Authorization grace timer
c) Set Authentication keys as valid

3-E: reauth SA TEK Rsp wait (PKMv2 *SA_TEK_Rsp*)→Authenticated
a) start ~~Authentication~~ Authorization grace timer
c) Set frame number for Authentication keys to become valid.
d) Clear old authentication keys upon frame number arrive.

4-B: not Authenticated (*EAP success*)→not Authenticated
a) Obtain MSK
b) Derive Authentication keys (AK, KEK, CMAC/HMAC etc..)

4-D: Authenticated (*EAP success*)→Authenticated
a) Obtain MSK
b) Derive Authentication keys (AK, KEK, CMAC/HMAC etc..)

5-C: SA TEK Rsp wait (*SATEK timer*)→ SA TEK Rsp wait
a) Send SA TEK Req.

b) Start SATEK Timer
c) SATEK counter--

5-E: reauth SA TEK Rsp wait (*SATEK timer*)→ reauth SA TEK Rsp wait
a) Send SA TEK Req.
b) Start SATEK Timer
c) SATEK counter--

6-C: SA TEK Rsp wait (*SATEK counter elapsed*)→ Stopped
a) Stop auth FSM
b) Signal fail to top level FSM

6-E: reauth SA TEK Rsp wait (*SATEK counter elapsed*)→ Authenticated
a) do nothing

7-D: Authenticated (*reauth needed*)→Authneticated
a) Send EAP_Start
b) Start EAP_Start timer

8-D: Authenticated (*optimized re-entry*)→optimized reentry wait
a) Retrieve AK and context for TBS
b) Delete re-authentication 3-way handshake temporal context is exists (MS must maintain the PMK and AK context because 3-way HS may re-start after re-entry finished).

8-E: reauth SA TEK Rsp wait (*optimized re-entry*)→ optimized reentry wait
a) Terminate re-auth context
b) Retrieve AK and context for TBS

9-D: Authenticated (*EAP Start Timer*)→Authneticated
a) Send EAP_Start
b) Start EAP_Start timer

10-F: optimized reentry wait (*Re-entry completed*)→ Authenticated
a) Set chosen BS context as current

11-F: optimized reentry wait (*HO canceled*)→ Authenticated
a) Continue using current BS context

12-F: optimized reentry wait (*TBS changed*)→ optimized reentry wait
a) Retrieve context of new TBS
b) Cache context of former TBS for as long as needed.

13-D,E,F: Any state (*Auth Expired*)→Stopped
a) Stop TEK FSMs
b) Stop Auth FSM
c) Disconnect from NW

14-A..F: Any state (*EAP FAIL*)→Stopped
a) Stop TEK FSMs
b) Stop Auth FSM
c) Disconnect from NW

15-A..F: Any state (*External Stop*)→Stopped
a) Stop TEK FSMs
b) Stop Auth FSM
c) Disconnect from NW

[Please update clause “7.8.1 PKMv2 SA-TEK 3-way handshake” in page 309 of 802.16e-2005 as follows]

The PKMv2 SA-TEK 3-way handshake sequence proceeds as follows:
1) During initial network entry or reauthorization, the BS shall send PKMv2 SA-TEK-Challenge (including a random number BS_Random) to the SS after protecting it with the HMAC/CMAC Tuple. If the BS does not receive PKMv2 SA-TEK-Request from the SS within SAChallenge-Timer, it shall resend the previous PKMv2 SA-TEK-Challenge up to SAChallengeMaxResends times. If thBit #e BS reaches its maximum number of resends, it shall:
In case of initial entry - initiate another full authentication or drop the SS.
In case of BS initiated re-authentication – initiate another full re-authentication.
In case of MS initiated re-authentication – ignore and continue using current authentication.
2) If HO Process Optimization Bit #1 is set indicating that PKM Authentication phase is omitted during network re-entry or handover, the BS begins the 3-way-handshake by appending the SA Challenge Tuple TLV to the RNG-RSP. If the BS does not receive PKMv2 SA-TEK-Request from the MS within SaChallengeTimer (suggested to be several times greater than the length of SaChallengeTimer), it may initiate full re-authentication or drop the MS. If the BS receives an initial RNG-REQ during the period that PKMv2 SA-TEK-Request is expected, it shall send a new RNG-RSP with another SaChallenge TLV.
3) The SS shall send PKMv2 SA-TEK-Request to the BS after protecting it with the HMAC/ CMAC. If the SS does not receive PKMv2 SA-TEK-Response from the BS within SATEKTimer, it shall resend the request. The SS may resend the PKMv2 SA-TEK-Request up to SATEKRequestMaxResends times. If the SS reaches its maximum number of resends, it shall:
In case of initial entry - initiate another full authentication or attempt to connect to another BS
In case of BS initiated re-authentication – ignore and continue using current authentication.
In case of MS initiated re-authentication – initiate another full re-authentication.
The SS shall include,through the Security Negotiation Parameters attribute, the security capabilities that it includedin the SBC-REQ message during the basic capabilities negotiation phase.

[Please insert the following raw to “table 343 – Operational ranges for privacy configuration settings”” in section 10.2 in page 658 of 802.16e-2005]

Table 343 – Operational ranges for privacy configuration settings

System	Name	Description	Min	Default	max
MS	PN grace value	The value of CMAC PN counter that triggers re-authentication	0x7FFFFFFF	0xFF000000	0xFF000000
MS	~~Authentication~~	Time before	~~60sec~~300sec	600sec	3600sec

	Authorization grace time	authentication expires that triggers re-authentication			
MS	Eap start timeout	Timer between resend of EAP start if re authentication was not completed	10sec	10sec	60sec