Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**	
Title	**Efficient and Secure Security Framework in PKMv2**	
Data Submitted	**2005-03-09**	
Source(s)	Seokheon Cho Sungcheol Chang Chulsik Yoon, ETRI 161, Gajeong-dong, Yuseong-Gu, Daejeon, 305-350, Korea	Voice: +82-42-860-5524 Fax: +82-42-861-1966 chosh@etri.re.kr
Re:	IEEE P802.16e/D6	
Abstract	The existing PKMv2 is somewhat unorganized and insecure security framework. This contribution provides a resolution for unorganized and insecure issues in the PKMv2.	
Purpose	Adoption of proposed changes into P802.16e/D6	
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:chiar@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>.	

Efficient and Secure Security Framework in PKMv2

Seokheon Cho, Sungcheol Chang, and Chulsik Yoon
ETRI

Introduction

The existing PKMv2 is somewhat in disorder and provides unorganized and insecure security framework.
This contribution supports the backward compatibility with the PKMv1 and security framework of the PKMv2.

This contribution provides a resolution for those problems in the PKMv2.

Remedy 1. Corrections for Flow and Message Confusion between PKMv1 and PKMv2

1.1 IEEE P802.16e/D6 status

There are many sub-messages in the PKM-REQ/RSP messages. Some of them are for the PKMv1, the other are for the PKMv2.

1.2 Problems

_ Messages for the PKMv2 were obviously proposed for assuring more secure message transfer, safe key share, and so on. But, it is difficult to distinguish which messages are for the PKMv1 or PKMv2, e.g. Key-Request message, Key-Reply message, EAP-Transfer message.

_ Some messages included in the PKMv1 are needed for full operation of the PKMv2. Those messages need to be changed to satisfy the aim of PKMv2 and backward compatibility with PKMv1

1.3 Solutions

We propose PKM-related flow and messages as follows.

a) The messages included in the PKMv1 are remained for backward compatibility with the PKMv1. In other words, the name and the attributes of messages are maintained.
   - SA Add, Auth Request, Auth Reply, Auth Reject, Key Request, Key Reply, Key Reject, Auth Invalid, TEK Invalid, and Auth Info messages

b) The messages included in the PKMv2 are changed under the PKMv2 procedure features. In other words, the names of messages are changed by procedure features. Their attributes are changed in case that some problems in the attribute occur. Moreover, code values for PKMv2 message type are re-numbered.
   - For the RSA-based Authorization procedure: The name of messages for the RSA-based Authorization procedure and a few attributes included in those messages are changed as follows:
     i. PKMv2 Auth-Request message → PKMv2 RSA-Request message (Name and attributes are changed: code # 13)
     ii. PKMv2 Auth-Reply message → PKMv2 RSA-Reply message (Name and attributes are changed: code # 14)
     iii. PKMv2 RSA-Reject message (New message is added: code # 15)
     iv. PKMv2 RSA-Acknowledgement message (New message is added: code # 16)
   - For the EAP-based Authorization procedure: The name of messages for the EAP-based Authorization procedure and a few attributes included in those messages are changed as follows:
     i. EAP Transfer message → PKMv2 EAP-Transfer message (Name is changed: code # 17)
     ii. Protected EAP message → PKMv2 Protected-EAP-Transfer message (Name and attributes are changed: code # 18)
     iii. PKMv2 EAP-Transfer-Complete message (New message is added: code # 19)
   - For MS's Authorization Key Generation procedure: This procedure generates the AK with seeds (such as Nonce) transferred from MS and BS. New messages for MS's Authorization Key Generation procedure are as follows:
     i. PKMv2 Authorization-Challenge message (New message is added: code # 20)
     ii. PKMv2 Authorization-Request message (New message is added: code # 21)
     iii. PKMv2 Authorization-Reply message (New message is added: code # 22)
     iv. PKMv2 Authorization-Reject message (New message is added: code # 23)
   - For the TEK exchange procedure: This procedure is for distributing TEK (or GTEK) in protecting replay-attack. The protecting function from replay-attack is added into the messages used for PKMv1 TEK exchange procedure. New messages for TEK exchange procedure are as follows:
     i. PKMv2 Key-Request message (New message is added: code # 24)
     ii. PKMv2 Key-Reply message (New message is added: code # 25)
     iii. PKMv2 Key-Reject message (New message is added: code # 26)
   - For the Dynamic SA addition procedure: This procedure is for adding new dynamic SA in protecting replay-attack

The protecting function from replay-attack is added into the messages used for PKMv1 Dynamic SA addition procedure. New messages for Dynamic SA addition procedure are as follows:

i. PKMv2 SA-Addition message (New message is added: code # 27)

- For the TEK Invalid procedure: This procedure is for informing MS of using the invalid TEK in protecting replay-attack. The protecting function from replay-attack is added into the messages used for PKMv1 TEK Invalid procedure. New messages for TEK Invalid procedure are as follows:

i. PKMv2 TEK-Invalid message (New message is added: code # 28)

- For Group Key Update procedure: This procedure is for pushing Group keying material to MSs. The name of messages for Group Key Update procedure are changed as follows:

i. Group Key Update Command message → PKMv2 Group-Key-Update-Command message (Name and attributes are changed: code # 29)

- For Pre-Authentication procedure: This procedure is for pre authentication for MS trying to HO. The name and the attributes of messages for Group Key Update procedure are changed as follows:

i. Pre-Auth-Request message → PKMv2 Pre-Authentication-Request message (Name and attributes are changed: code # 30)

ii. Pre-Auth-Reply message → PKMv2 Pre-Authentication-Reply message (Name and attributes are changed: code #31)

iii. Pre-Auth-Reject message → PKMv2 Pre-Authentication-Reject message (Name and attributes are changed: code #32)

Code	PKM message type	MAC Management message name
~~13~~	~~EAP Transfer~~	~~PKM-REQ/PKM-RSP~~
~~14~~	~~Pre-Auth Request~~	~~PKM-REQ~~
~~15~~	~~Pre-Auth Reply~~	~~PKM-RSP~~
~~16~~	~~Pre-Auth Reject~~	~~PKM-RSP~~
~~17~~	~~PKMv2 Auth-Request~~	~~PKM-REQ~~
~~18~~	~~PKMv2 Auth-Reply~~	~~PKM-RSP~~
~~19~~	~~Key Update Command~~	~~PKM-RSP~~
~~20~~	~~Protected EAP~~	~~PKM-REQ/PKM-RSP~~
~~21~~	~~SA-TEK-Challenge~~	~~PKM-RSP~~
~~22~~	~~SA-TEK-Request~~	~~PKM-REQ~~
~~23~~	~~SA-TEK-Response~~	~~PKM-RSP~~
~~24-255~~	~~reserved~~	
13	PKMv2 RSA-Request	PKM-REQ
14	PKMv2 RSA-Reply	PKM-RSP
15	PKMv2 RSA-Reject	PKM-RSP
16	PKMv2 RSA-Acknowledgement	PKM-REQ
17	PKMv2 EAP-Transfer	PKM-REQ/PKM-RSP
18	PKMv2 Protected EAP-Transfer	PKM-REQ/PKM-RSP
19	PKMv2 EAP-Transfer-Complete	PKM-REQ
20	PKMv2 Authorization-Challenge	PKM-RSP
21	PKMv2 Authorization-Request	PKM-REQ
22	PKMv2 Authorization-Reply	PKM-RSP
23	PKMv2 Authorization-Reject	PKM-RSP
24	PKMv2 Key-Request	PKM-REQ
25	PKMv2 Key-Reply	PKM-RSP
26	PKMv2 Key-Reject	PKM-RSP
27	PKMv2 SA-Addition	PKM-RSP
28	PKMv2 TEK-Invalid	PKM-RSP
29	PKMv2 Group-Key-Update-Command	PKM-RSP
30	PKMv2 Pre-Authentication-Request	PKM-REQ
31	PKMv2 Pre-Authentication-Reply	PKM-RSP
32	PKMv2 Pre-Authentication-Reject	PKM-RSP
33-255	reserved	

Remedy 2. Corrections for MS's Authorization Flow

2.1 IEEE P802.16e/D6 Status

The AK is derived from the PAK or/and the PMK. The PAK and the PMK are obtained from the RSA-based Authorization procedure and the EAP-based Authorization procedure, respectively.

2.2 Problems

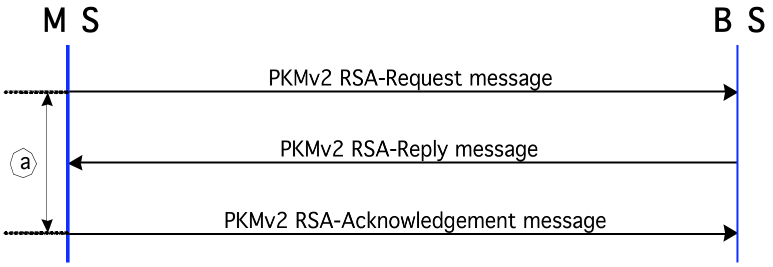
_ For the RSA-based Authorization procedure:
- The Authorization Request message and the Authorization Reply message in the RSA-based Authorization procedure are in the same rank as the EAP Transfer message in the EAP-based Authorization procedure. The Authorization Request/Reply messages contains Security-Capabilities, SAID, and SA-Descriptors, but the EAP Transfer message doesn't contain those parameters.
- The sequence number and lifetime included in the RSA-based Authorization procedure are not an AK sequence number and AK lifetime.
- When the RSA-based Authorization procedure is selected, there is no message for informing the MS's authentication failure from BS to MS and the BS's authentication result (such as success or reject) from MS.

_ For the EAP-based Authorization procedure:
- The Key Sequence Number used in the Protected EAP-Transfer message is not an AK Sequence Number.
- The BS doesn't know the completion time of the EAP-based Authorization procedure in case that EAP protocol doesn't yield AAA-key and whether an MS receives the last EAP Transfer message (such as "EAP Success" used in EAP-TLS) or not. Both the BS and the MS cannot simultaneously share the AK derived from PMK, when an MS doesn't receive the last EAP Transfer message.

_ AK generation process needs more secure mechanism.

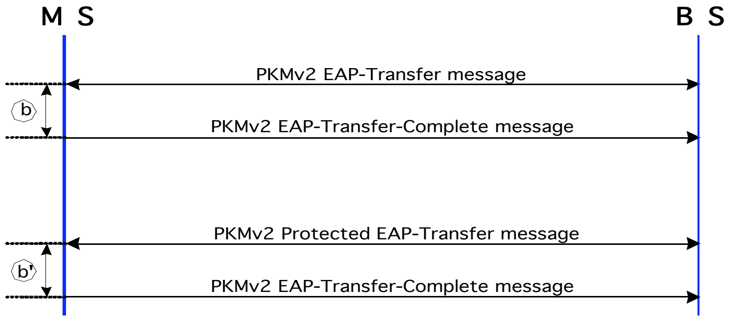
2.3 Solutions

a) For the RSA-based Authorization procedure:
- Since the RSA-based Authorization is the same rank as the EAP-based Authorization, several attributes (such as Security-Capabilities, SAID, and SA-Descriptors) which are present in the EAP-based Authorization procedure, are omitted in the RSA-based Authorization procedure. The RSA-based Authorization procedure still supports mutual authentication. The messages in the RSA-based Authorization procedure are as follows:
- The sequence number and lifetime included in the RSA-based Authorization procedure is not an AK sequence number and AK lifetime but the PAK sequence number and PAK lifetime.
- To inform the MS's authentication failure from BS to MS, the PKMv2 RSA-Reject message is added. Also, to inform the BS's authentication result (such as success or reject) from MS to BS, Auth Result Code and Error-Code attributes shall be included in the PKMv2 RSA-Acknowledgement message.



i. PKMv2 RSA-Request message: MS_Random, MS_Cerfificate
ii. PKMv2 RSA-Reply message: MS_Random, BS_Random, Encrypted pre-PAK, Key lifetime (PAK), Key Sequence Number (PAK), BS_Certificate, SigBS
iii. PKMv2 RSA-Reject message: MS_Random, BS_Random, Error-Code, Display-String, SigBS
iv. PKMv2 RSA-Acknowledgement message: BS_Random, Auth Result Code, Error-Code, Display-String, SigMS

b) For the EAP-based Authorization procedure:
- The messages used in the EAP-based Authorization procedure are as follows
- The Key Sequence Number used in the PKMv2 Protected EAP-Transfer message is not an AK sequence number but PAK sequence number.
- In order that BS knows the completeness time of EAP-based Authorization procedure in case that EAP protocol doesn't yield AAA-key and whether a MS receives the last EAP Transfer message (such as "EAP Success" used in EAP-TLS) or not, an MS shall send the PKMv2 EAP-Transfer-Complete message to report EAP-based Authorization completeness. Therefore, both MS and BS can synchronize the AK.


MS
BS
PKMv2 EAP-Transfer message
b
PKMv2 EAP-Transfer-Complete message
PKMv2 Protected EAP-Transfer message
b'
PKMv2 EAP-Transfer-Complete message

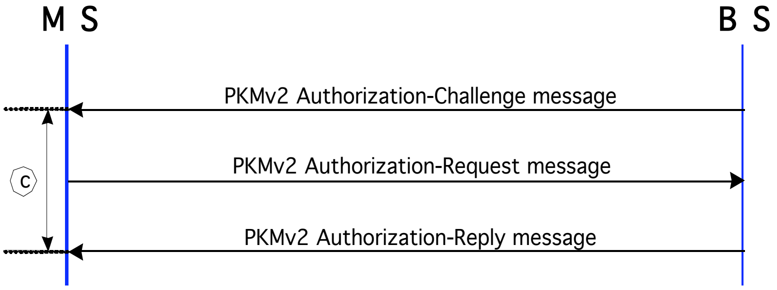
i. PKMv2 EAP-Transfer message: EAP Payload

ii. PKMv2 Protected EAP-Transfer message: Key Sequence Number (PAK), EAP Payload, OMAC Digest (from EIK)

iii. PKMv2 EAP-Transfer-Complete message:

c) For MS's Authorization Key Generation procedure:

- To assure more secure AK, seeds (such as MS_Nonce and BS_Nonce) should be used to derive AK.
- To protect from replay-attack, the messages needed in MS's AK Generation procedure should contains random number (such as MS_Nonce and BS_Nonce) and message authentication function (such as OMAC Digest).
- The messages with important parameters, e.g. Security-Capabilities, SAID, and SA-Descriptors, should be authenticated.
- This procedure supports secure 3 way handshake.


MS
BS
PKMv2 Authorization-Challenge message
c
PKMv2 Authorization-Request message
PKMv2 Authorization-Reply message

i. PKMv2 Authorization-Challenge message: BS_Nonce

ii. PKMv2 Authorization-Request message: Key Sequence Number (PAK), MS_Nonce, BS_Nonce, Security_Capabilities, SAID, OMAC Digest (from AK)

iii. PKMv2 Authorization-Reply message: Key Sequence Number (AK), Key Lifetime (AK), BS_Nonce, (one or more) SA-Descriptor(s), OMAC Digest (from AK)

iv. PKMv2 Authorization-Reject message: Error-Code, Display-String, BS_Nonce, OMAC Digest (from AK)

Remedy 3. Corrections for PKMv2 Key Hierarchy

3.1 IEEE P802.16e/D6 Status

The Key Hierarchy for the PKMv2 is defined. The AK is derived by PAK or/and PMK, SSID, BSID, and so on.

3.2 Problems

_ The AK is derived from PAK or/and PMK which are generated and distributed from the BS and Authenticator, respectively. A Nonce from MS as well as BS is necessary to generate AK so as to make more secure key generation mechanism.

_ The input key used for AK generation is the only PMK. The PAK should be also used to generate the AK as not input data but an input key.

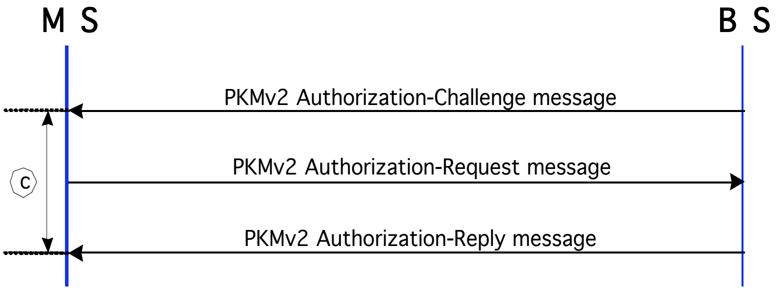
_ The value of EAP session-id is not changed, even though the new AAA-key is refreshed. That is, even if PMK is updated, the value of PMKID (=> hash64(EAP session-id)) and AKID (=> hash64(EAP sessionid|PAKID|BSID)) is not also changed. Therefore, AKID is unsuitable as the identifier or sequence number needed to distinguish new AK from old AK.

- The AK lifetime is computed from the value (=> MIN(PAK lifetime, PMK lifetime)). To maintain AK more secure, however, the AK should be frequently refreshed. Different definition of the AK lifetime is necessary.

3.3 Solutions

a) To derive AK, 3 way handshake procedure is newly provided as follows.

- The MS_Nonce and the BS_Nonce generated from the MS and the BS respectively. These MS_Nonce and BS_Nonce with PAK or/and PMK shall be used to generate AK.


M S
B S
PKMv2 Authorization-Challenge message
PKMv2 Authorization-Request message
PKMv2 Authorization-Reply message
c

i. PKMv2 Authorization-Challenge message: BS_Nonce
ii. PKMv2 Authorization-Request message: Key Sequence Number (PAK), MS_Nonce, BS_Nonce, Security_Capabilities, SAID, OMAC Digest (from AK)
iii. PKMv2 Authorization-Reply message: Key Sequence Number (AK), Key Lifetime (AK), BS_Nonce, (one or more) SA-Descriptor(s), OMAC Digest (from AK)
iv. PKMv2 Authorization-Reject message: Error-Code, Display-String, BS_Nonce, OMAC Digest (from AK)

- The input key for generating the AK should be both PAK and PMK. The exclusive-or (XOR: ⊕) value of PAK and PMK as input key is used to generated the AK. The generation method of the AK is as follows.

```
If (RSA-based authorization and EAP-based authorization)
        AK <= Dot16KDF (PAK⊕PMK, SS_NONCE|BS_ NONCE|SSID|BSID|“AK”, 160)
Else if (RSA-based authorization)
        AK <= Dot16KDF (PAK, SS_ NONCE|BS_ NONCE|SSID|BSID|“AK”, 160)
Else if (EAP-based authorization)
        AK <= Dot16KDF (PMK, SS_ NONCE|BS_ NONCE|SSID|BSID|“AK”, 160)
```

- The new key hierarchy is as follows.

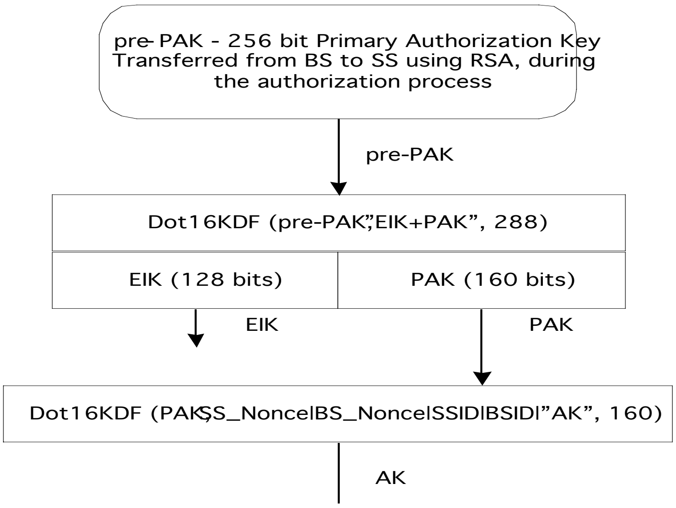

pre-PAK - 256 bit Primary Authorization Key Transferred from BS to SS using RSA, during the authorization process
pre-PAK
Dot16KDF (pre-PAK,"EIK+PAK”, 288)
EIK (128 bits)
PAK (160 bits)
EIK
PAK
Dot16KDF (PAK,SS_Nonce|BS_Nonce|SSID|BSID|”AK”, 160)
AK

Figure -AK with the only RSA-based authorization process

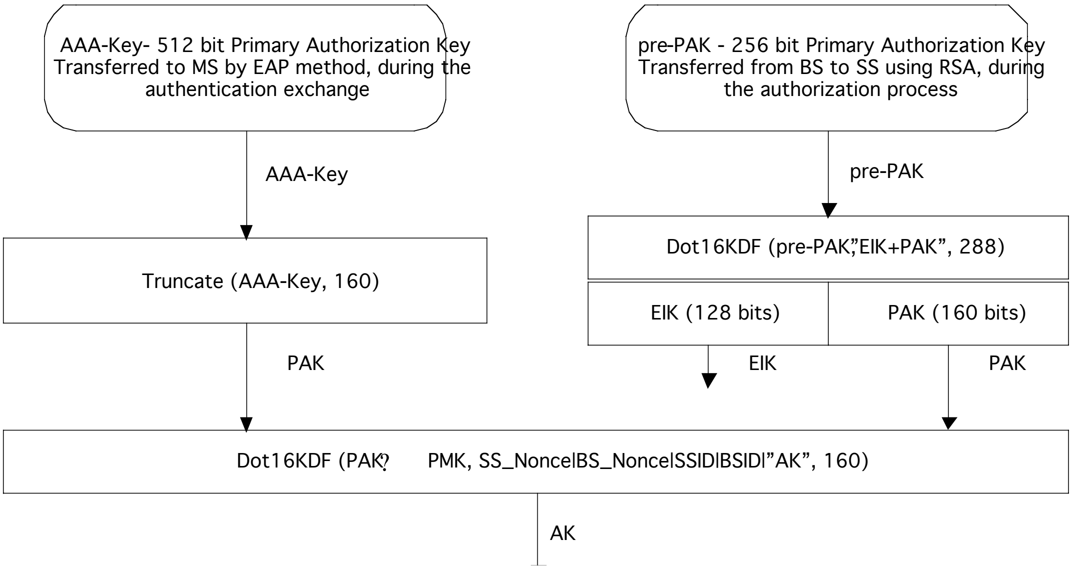

AAA-Key- 512 bit Primary Authorization Key Transferred to MS by EAP method, during the authentication exchange
pre-PAK - 256 bit Primary Authorization Key Transferred from BS to SS using RSA, during the authorization process
AAA-Key
pre-PAK
Truncate (AAA-Key, 160)
Dot16KDF (pre-PAK,"EIK+PAK", 288)
EIK (128 bits)
PAK (160 bits)
PAK
EIK
PAK
Dot16KDF (PAK? PMK, SS_Nonce|BS_Nonce|SSID|BSID|"AK", 160)
AK

Figure -AK with RSA and EAP authorization process

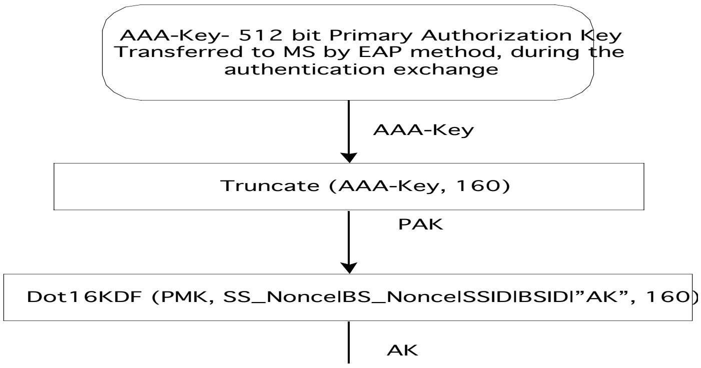

AAA-Key- 512 bit Primary Authorization Key Transferred to MS by EAP method, during the authentication exchange
AAA-Key
Truncate (AAA-Key, 160)
PAK
Dot16KDF (PMK, SS_Nonce|BS_Nonce|SSID|BSID|"AK", 160)
AK

Figure -AK with the only EAP-based authorization process

- To solve the AKID, the AK sequence number as AK identifier is newly defined. The BS generates the AK sequence number and informs it to MS, whenever the AK is updated.
- To maintain AK more secure, the AK has AK lifetime which is assigned from the BS. That is, the MS shall request the new AK, before the old AK expires and after the PAK or the PMK is updated.
- The AK context is as follows.

Table -AK Context in PKMv2

Parameter	Size	Usage
Primary AK (PAK)	160bits	A key yielded from the RSA-based authorization.
PAK Sequence Number	64bits	PAK sequence number, when the RSA-based authorization is achieved.
PAK lifetime		PAK sequence number, when the RSA-based authorization is achieved.
PMK	160bits	A key yielded from the EAP-based authentication (only if EAP protocol generates the AAA-key)..
PMK lifetime		PMK sequence number, when the EAP-based authorization is achieved and the AAA-key is obtained.
AK	160bits	The authorization key, calculated as defined in 7.2.2.2.3
AK Sequence Number	64bits	AK sequence number
AK lifetime		AK lifetime – when this expires, MS's Re-authorization Key process is needed.
H/OMAC_KEY_U	160 bits/128 bits	The key which is used for signing UL management messages.
H/OMAC_PN_U	32 bits	Used to avoid UL replay attack on management messages – when this expires re-authentication is needed.
H/OMAC_KEY_D	160 bits/128 bits	The key which is used for signing DL management messages.

H/OMAC_PN_D	32 bits	Used to avoid DL replay attack on management messages – when this expires re-authentication is needed.
KEK	160 bits	Used to encrypt TEK or GKEK from the BS to the SS.

Remedy 4. Corrections for Adaptation the PKMv1 Messages to PKMv2

4.1 IEEE P802.16e/D6 Status

Some messages defined in the PKMv1 are still used in the PKMv2.

4.2 Problems

_ Some messages included in the PKMv1 are needed for full operation of the PKMv2. Those messages need to be changed to satisfy the aim of PKMv2 and backward compatibility with PKMv1.

4.3 Solutions

b) For TEK exchange procedure:
- The messages used in the PKMv1 should be added some attributes to protect replay-attack.
  i. PKMv2 Key-Request message: Key Sequence Number (AK), SAID, MS_Nonce, OMAC Digest (from AK)
  ii. PKMv2 Key-Reply message: Key Sequence Number (AK), SAID, TEK-Parameters (for old), TEK-Parameters (for new), BS_Nonce, OMAC Digest (from AK)
  iii. PKMv2 Key-Reject message: Key Sequence Number (AK), SAID, Error-Code, Display-String, BS_Nonce, OMAC Digest (from AK)

c) For Dynamic SA addition procedure:
- The messages used in the PKMv1 should be added some attributes to protect replay-attack.
  i. PKMv2 SA-Addition message: Key Sequence Number (AK), (one or more) SA-Descriptor(s), BS_Nonce, OMAC Digest (from AK)

d) For TEK Invalid procedure:
- The messages used in the PKMv1 should be added some attributes to protect replay-attack.
  i. PKMv2 TEK Invalid message: Key Sequence Number (AK), SAID, Error-Code, Display-String, BS_Nonce, OMAC Digest (from AK)

Remedy 5. Corrections for 3 Way SA-TEK Exchange

5.1 IEEE P802.16e/D6 Status

There are messages related to 3 way handshake SA-TEK exchange, e.g. SA-Challenge, SA-TEK-Request, and SA-TEK-Response. These messages are used during initial network entry, reauthorization, HO.

5.2 Problems

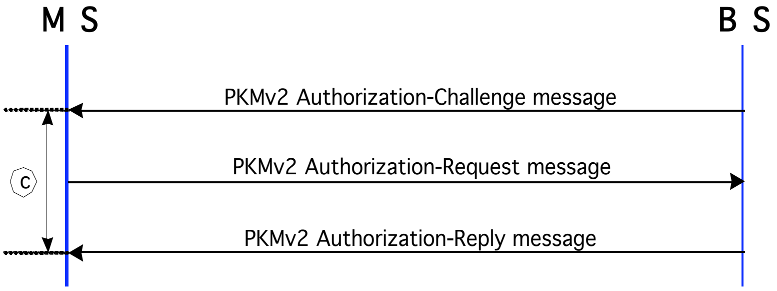
_ The Security_Capabilities, SAID, and SA-Descriptors attributes are included in SA-TEK exchange. However, negotiation of Security_Capabilities and SA-Descriptor should be done before the MS generates and distributes the TEK. It is reasonable that those attributes should be negotiated during the AK generation procedure.

_ The SA-Descriptors included in SA-TEK exchage identifies the Primary and Static SAs the requesting MS is authorized to access and their particular properties. In the case of the multicast service, it is so dangerous to distribute the information of all Static SAs (including static SAID and static TEK-parameters) without DSx-exchange procedure (= without user's use intention for the multicast service). In order to use this SA-TEK exchange procedure, all DSx-exchanges for Static SAs should be performed.

_ It is already defined that the TEK doesn't need to be updated during reauthorization in the IEEE P802.16d/D5. Thus, the TEK doesn't need to be refreshed during HO. The TEK-parameters transfer and share among BSs should be guaranteed. If not, no information shall be shared among BSs and even HO-optimization is impossible.

5.3 Solutions

a) It is appropriate that Security_Capabilities and SA-Descriptors should be transferred not during the TEK exchange procedure but during the MS's AK generation procedure. The following MS's Authorization Key Generation procedure shall support the above solution and transfer those attributes securely.


M S
B S
PKMv2 Authorization-Challenge message
PKMv2 Authorization-Request message
PKMv2 Authorization-Reply message
c

v. PKMv2 Authorization-Challenge message: BS_Nonce

vi. PKMv2 Authorization-Request message: Key Sequence Number (PAK), MS_Nonce, BS_Nonce, Security_Capabilities, SAID, OMAC Digest (from AK)

vii. PKMv2 Authorization-Reply message: Key Sequence Number (AK), Key Lifetime (AK), BS_Nonce, (one or more) SA-Descriptor(s), OMAC Digest (from AK)

viii. PKMv2 Authorization-Reject message: Error-Code, Display-String, BS_Nonce, OMAC Digest (from AK)

b) The DSx-exchange procedure (user's intention) should precede the TEK exchange procedure, especially the multicast service to use Static SA. It is appropriate to use the PKMv2 Key-Request and the PKMv2 Key-Reply message after performing DSx-exchange procedure.

Proposed Changes into IEEE P802.16e/D6

Remedy 1. Corrections for Flow and Message Confusion between PKMv1 and PKMv2

[Change the Table 26 in sub-clause 6.3.2.3.9:]

6.3.2.3.9 Privacy key management (PKM) message (PKM-REQ/PKM-RSP)

Code	PKM message type	MAC Management message name
~~13~~	~~EAP Transfer~~	~~PKM-REQ/PKM-RSP~~
~~14~~	~~Pre-Auth Request~~	~~PKM-REQ~~
~~15~~	~~Pre-Auth Reply~~	~~PKM-RSP~~
~~16~~	~~Pre-Auth Reject~~	~~PKM-RSP~~
~~17~~	~~PKMv2 Auth-Request~~	~~PKM-REQ~~
~~18~~	~~PKMv2 Auth-Reply~~	~~PKM-RSP~~
~~19~~	~~Key Update Command~~	~~PKM-RSP~~
~~20~~	~~Protected EAP~~	~~PKM-REQ/PKM-RSP~~
~~21~~	~~SA-TEK-Challenge~~	~~PKM-RSP~~
~~22~~	~~SA-TEK-Request~~	~~PKM-REQ~~
~~23~~	~~SA-TEK-Response~~	~~PKM-RSP~~
~~24-255~~	~~reserved~~	
13	PKMv2 RSA-Request	PKM-REQ
14	PKMv2 RSA-Reply	PKM-RSP
15	PKMv2 RSA-Reject	PKM-RSP
16	PKMv2 RSA-Acknowledgement	PKM-REQ
17	PKMv2 EAP-Transfer	PKM-REQ/PKM-RSP
18	PKMv2 Protected EAP-Transfer	PKM-REQ/PKM-RSP
19	PKMv2 EAP-Transfer-Complete	PKM-REQ
20	PKMv2 Authorization-Challenge	PKM-RSP
21	PKMv2 Authorization-Request	PKM-REQ
22	PKMv2 Authorization-Reply	PKM-RSP
23	PKMv2 Authorization-Reject	PKM-RSP
24	PKMv2 Key-Request	PKM-REQ
25	PKMv2 Key-Reply	PKM-RSP
26	PKMv2 Key-Reject	PKM-RSP
27	PKMv2 SA-Addition	PKM-RSP
28	PKMv2 TEK-Invalid	PKM-RSP
29	PKMv2 Group-Key-Update-Command	PKM-RSP
30	PKMv2 Pre-Authentication-Request	PKM-REQ
31	PKMv2 Pre-Authentication-Reply	PKM-RSP
32	PKMv2 Pre-Authentication-Reject	PKM-RSP
33-255	reserved	

Remedy 2. Corrections for MS's Authorization Flow

[Change sub-clauses 6.3.2.3.9.15 as follows]

~~6.3.2.3.9.15 Auth-Request message~~
6.3.2.3.9.11 PKMv2 RSA-Request message

A client MS sends a PKMv2 RSA-Request message to the BS in order to request mutual authentication in the RSA-based authorization.

Code: ~~21~~ 13

Attributes are shown in Table ~~37e~~ 37a.

Table ~~37e~~ 37a-~~Auth-Request attributes~~ PKMv2 RSA-Request attributes

Attribute	Contents
~~SS_Random~~ MS_Random	A 64 bit random number generated in the MS
~~SS_Certificate~~ MS_Certificate	Contains the MS's X.509 user certificate
~~Security_Capabilities~~	~~Describes requesting MS's security capabilities~~
~~AAID/SAID~~	~~Either the AAID or the Basic CID if in initial network entry~~

The MS-certificate attribute contains an X.509 MS certificate (see 7.6) issued by the MS's manufacturer.
The MS's X.509 certificate ~~and Security Capabilities attribute~~ is as defined in 6.3.2.3.9.2.

[Change sub-clauses 6.3.2.3.9.16 as follows]

~~6.3.2.3.9.16 Auth-Reply message~~
6.3.2.3.9.12 PKMv2 RSA-Reply message

Sent by the BS to a client MS in response to ~~an Authorization Request~~ a PKMv2 RSA-Request message, the ~~Authorization Reply~~ PKMv2 RSA-Reply message contains ~~an AK~~ an encrypted pre-PAK, the key's lifetime, and the key's sequence number~~, and a list of SA-Descriptors identifying the Primary and Static SAs the requesting MS is authorized to access and their particular properties (e.g., type, cryptographic suite).~~ The ~~AK~~ pre-PAK shall be encrypted with the MS's public key. ~~The SA-Descriptor list shall include a descriptor for the Basic CID reported to the BS in the corresponding Auth-Request.~~ The SS_Random number is returned from the ~~auth-req~~ PKMv2 RSA-Request message, along with a random number supplied by the BS, thus enabling assurance of key liveness.

Code: ~~22~~ 14

Attributes are shown in Table ~~37f~~ 37b.

Table ~~37f~~ 37b-~~Auth-Reply attributes~~ PKMv2 RSA-Reply attributes

Attribute	Contents
MS_Random	A 64 bit random number generated in the MS
BS_Random	A 64 bit random number generated in the BS
Encrypted pre-PAK	RSA-OAEP-Encrypt(PubKey(MS), pre-PAK|~~Id(MS)~~ MS ID)
Key Lifetime	~~AK~~ PAK Aging timer
Key Sequence Number	64 bit ~~AK~~ PAK sequence number
~~(one or more) SA-Descriptor(s)~~	~~The primary SA and zero or more static SAs.Each compound SA-Descriptor attribute specifies an SAID and additional properties of the SA (optional, only if there is no EAP phase afterwards)~~
~~CertBS~~ BS_Certificate	~~The BS Certificate~~ Contains the BS's X.509 certificate
SigBS	An RSA signature over all the other attributes in the message

[Insert the following sub-clause in 6.3.2.3.9:]

6.3.2.3.9.13 PKMv2 RSA-Reject message

The BS responds to an SS's authorization request with an Authorization Reject message if the BS rejects the SS's authorization request.

Code: 15

Attributes are shown in Table 37c.

Table 37c-PKMv2 RSA-Reject attributes

Attribute	Contents
MS_Random	A 64 bit random number generated in the MS
BS_Random	A 64 bit random number generated in the BS
Error-Code	Error code identifying reason for rejection of authorization request
Display-String (optional)	Display string providing reason for rejection of authorization request
SigBS	An RSA signature over all the other attributes in the message

The Error-Code and Display-String attributes describe to the requesting MS the reason for the RSA-based authorization failure.

[Insert the following sub-clause in 6.3.2.3.9:]

6.3.2.3.9.14 PKMv2 RSA-Acknowledgement message

The MS sends the PKMv2 RSA-Acknowledgement message to BS in response to a PKMv2 RSA-Reply message or a PKMv2 RSA-Reject message. Only if the value of Auth Result Code is failure, then the Error-Code and Display-String can be included in this message.

Code: 16

Attributes are shown in Table 37d.

Table 37d-PKMv2 RSA-Acknowledgement attributes

Attribute	Contents
BS_Random	A 64 bit random number generated in the BS
Auth Result Code	Indicates result (Success or Failure) of authorization procedure.
Error-Code	Error code identifying reason for rejection of authorization request
Display-String (optional)	Display string providing reason for rejection of authorization request
SigMS	An RSA signature over all the other attributes in the message

[Change sub-clauses 6.3.2.3.9.11 as follows]

~~6.3.2.3.9.11 EAP Transfer message~~

6.3.2.3.9.15 PKMv2 EAP-Transfer message

When an MS has an EAP message received from an EAP method for transmission to the BS or when a BS has an EAP message received from an EAP method for transmission to the MS, it encapsulates it ~~in an EAP Transfer~~ a PKMv2 EAP Transfer message.

Code: ~~13~~ 17

Attributes are shown in Table ~~37a~~ 37e.

Table ~~37a~~ 37e ~~–EAP Transfer attributes~~ PKMv2 RSA-Acknowledgement attributes

Attribute	Contents
EAP Payload	Contains the EAP authentication data, not interpreted in the MAC

The EAP Payload field carries data in the format described in section 4 of RFC 2284bis.

[Change sub-clauses 6.3.2.3.9.18 as follows]

~~6.3.2.3.9.18 Protected EAP message~~

6.3.2.3.9.16 PKMv2 Protected EAP-Transfer message

If EIK is available and an MS or BS has an EAP message received from an EAP method for transmission, it encapsulates EAP message in ~~a Protected EAP Transfer message~~ a PKMv2 Protected EAP Transfer message. In other words, this message may be used in case that both an MS and BS negotiate RSA-based authorization and Protected EAP-based authorization as authorization policy support.

Code: ~~24~~ 18

Attributes are shown in Table ~~37h~~ 37f.

Table ~~37h~~ 37f ~~–Protected EAP message attributes~~ PKMv2 Protected EAP-Transfer attributes

Attribute	Contents
Key Sequence Number	~~AK~~ PAK Sequence Number
EAP Payload	Contains the EAP authentication data, not interpreted in the MAC
OMAC Digest	Message Digest calculated using EIK

The EAP Payload field carries EAP data in the format described in RFC 3748.

The OMAC-Digest attribute shall be the final attribute in the message’s attribute list.

Inclusion of the OMAC digest allows the MS and BS to cryptographically bind previous authorization and following EAP authentication by authenticating the EAP message. The OMAC-Digest’s authentication key is derived from the ~~AK~~ EIK.

[Insert the following sub-clause in 6.3.2.3.9:]

6.3.2.3.9.17 PKMv2 EAP-Transfer-Complete message

A MS sends the PKMv2 EAP-Transfer-Complete message to the BS to report completeness of EAP-based authorization procedure (PKMv2 EAP-Transfer message and PKMv2 Protected EAP-Transfer message). This message doesn’t contain any attributes.

Code: 19

Attributes are shown in Table 37g

Table 37g-PKMv2 EAP-Transfer-Complete attributes

Attribute	Contents

[Insert the following sub-clause in 6.3.2.3.9:]

6.3.2.3.9.18 PKMv2 Authorization-Challenge message

The BS transmits the PKMv2 Authorization-Challenge message to an MS at initial authorization procedure or at reauthorization procedure. After achieving the RSA-based authorization procedure or/and the EAP-based authorization procedure, the BS sends the PKMv2 Authorization-Challeng message to BS to share the AK (Authorization Key).

Code: 20

Attributes are shown in Table 37h

Table 37h-PKMv2 Authorization-Challenge attributes

Attribute	Contents
BS_Nonce	A 64 bit pseudo-random or random number freshly generated by the BS

[Insert the following sub-clause in 6.3.2.3.9:]

6.3.2.3.9.19 PKMv2 Authorization-Request message

Sent by a client MS to the BS in response to an PKMv2 Authorization-Challenge message, the PKMv2 Authorization-Request message may contain the pre-PAK's Sequence Number, the PMK's sequence number, MS_Nonce, BS_Nonce, Security Capabilities and SAID.

Code: 21

Attributes are shown in Table 37i

Table 37i-PKMv2 Authorization-Request attributes

Attribute	Contents
Key Sequence Number	PAK Sequence Number
MS_Nonce	A 64 bit pseudo-random or random number freshly generated by the MS
BS_Nonce	BS_Nonce included in the PKMv2 Authorization-Challenge message
Security_Capabilities	Describes requesting MS's security capabilities
SAID	MS's primary SAID equal to the Basic CID
OMAC-Digest	Message Digest calculated using AK

The pre-PAK Sequence Number and PMK Sequence Number shall be included in case of supporting the RSA-based authorization and the EAP-based authorization, respectively. And both pre-PAK Sequence Number and PMK Sequence Number shall be included when the RSA-based authorization and the EAP-based authorization are negotiated as authorization policy.

The Security Capabilities attribute is a compound attribute describing the requesting MS's security capabilities. This includes the data encryption, data authentication, and TEK(GTEK)/GKEK encryption algorithms the MS supports. The Security Capabilities and SAID attributes are as defined in 6.3.2.3.9.2.

The OMAC-Digest attribute shall be the final attribute in the message's attribute list.

Inclusion of the OMAC digest allows the MS and BS to authenticate the PKMv2 Authorization-Request message. The OMAC-Digest's authentication key is derived from the AK.

[Insert the following sub-clause in 6.3.2.3.9:]

6.3.2.3.9.20 PKMv2 Authorization-Reply message

Sent by the BS to a client MS in response to an PKMv2 Authorization-Request message, the PKMv2 Authorization-Reply message contains the AK's lifetime, the AK's sequence number, and a list of SA-Descriptors identifying the Primary and Static SAs the requesting MS is authorized to access and their particular properties (e.g., type, cryptographic suite). The SA-Descriptor list shall include a descriptor for the Basic CID reported to the BS in the corresponding PKMv2 Authorization-Request message. The MS_Nonce received from the PKMv2 Authorization-Request message is returned, along with Nonce supplied by the BS, thus enabling assurance of key liveness.

Code: 22

Attributes are shown in Table 37j

Table 37j-PKMv2 Authorization-Reply attributes

Attribute	Contents
Key Sequence Number	AK Sequence Number
Key Lifetime	AK Lifetime
BS_Nonce	BS_Nonce included in the PKMv2 Authorization-Challenge message
(one or more) SA-Descriptor(s)	The primary SA and zero or more static SAs. Each compound SA-Descriptor attribute specifies an SAID and additional properties of the SA

OMAC Digest	Message Digest calculated using AK

The OMAC-Digest attribute shall be the final attribute in the message's attribute list.

Inclusion of the OMAC digest allows the MS and BS to authenticate the PKMv2 Authorization-Reply message. The OMAC-Digest's authentication key is derived from the AK.

[Insert the following sub-clause in 6.3.2.3.9:]

6.3.2.3.9.21 PKMv2 Authorization-Reject message

The BS responds to a MS's PKMv2 Authorization-Request message with a PKMv2 Authorization-Reject message if the BS rejects the MS's authorization request.

Code: 23

Attributes are shown in Table 37k.

Table 37k-PKMv2 Authorization-Reject attributes

Attribute	Contents
Error-Code	Error code identifying reason for rejection of authorization request.
Display-String (optional)	Display string providing reason for rejection of authorization request.
BS_Nonce	BS_Nonce included in the PKMv2 Authorization-Challenge message
OMAC Digest	Message Digest calculated using AK

The Error-Code and Display-String attributes describe to the requesting MS the reason for the authorization failure.

The OMAC-Digest attribute shall be the final attribute in the message's attribute list.

Inclusion of the OMAC digest allows the MS and BS to authenticate the PKMv2 Authorization-Reject message. The OMAC-Digest's authentication key is derived from the AK.

[Delete 7.2.2.1] and [Insert 7.2.2.1 as follows] ***(Informative: Add the deleted contents in 7.2.2.1 into 7.2.2.3)***

~~7.2.2.1 Security Associations~~

~~Upon achieving authorization, an MS starts a separate TEK state machine for each of the SAIDs identified in the Authorization Reply message. Each TEK state machine operating within the MS is responsible for managing the keying material associated with its respective SAID. TEK state machines periodically send Key Request messages to the BS, requesting a refresh of keying material for their respective SAIDs.~~

~~The BS responds to a Key Request with a Key Reply message, containing the BS's active keying material for a specific SAID.~~

~~The TEK is encrypted using appropriate KEK derived from the AK.~~

~~TEKs and KEKs may be either 64 bits or 128 bits long. SAs employing any ciphersuite with a basic block size of 128 bits shall use 128 bit TEKs and KEKs. Otherwise 64 bit TEKs and KEKs shall be used. The name TEK-64 is used to denote a 64 bit TEK and TEK-128 is used to denote a 128 bit TEK. Similarly, KEK-64 is used to denote a 64 bit KEK and KEK-128 is used to denote a 128 bit KEK.~~

~~For SAs using a ciphersuite employing DES-CBC, the TEK in the Key Reply is triple DES (3-DES) (encrypt-decrypt-encrypt or EDE mode) encrypted, using a two-key, 3-DES KEK derived from the AK.~~

~~For SAs using a ciphersuite employing 128 bits keys, such as AES-CCM mode, the TEK in the key Reply is AES encrypted using a 128 bit key derived from the AK and a 128 bit block size.~~

~~Note that at all times the BS maintains two active sets of keying material per SAID. The lifetimes of the two generations overlap such that each generation becomes active halfway through the life of it predecessor and expires halfway through the life of its successor. A BS includes in its Key Replies both of an SAID's active generations of keying material.~~

~~The Key Reply provides the requesting SS, in addition to the TEK and CBC initialization vector, the remaining lifetime of each of the two sets of keying material. The receiving SS uses these remaining lifetimes to estimate when the BS will invalidate a particular TEK, and therefore when to schedule future Key Requests such that the SS requests and receives new keying material before the BS expires the keying material the SS currently holds.~~

~~For SAs using a ciphersuite employing CBC mode encryption the Key Reply provides the requesting MS, in addition to the TEK and CBC initialization vector, the remaining lifetime of each of the two sets of keying material. For SAs using a ciphersuite employing AES-CCM mode, the Key Reply provides the requesting MS, in addition to the TEK, the remaining lifetime of each of the two sets of keying material. The receiving MS uses these remaining lifetimes to estimate when the BS will invalidate a particular TEK, and therefore when to schedule future Key Requests such that the MS requests and receives new keying material before the BS expires the keying material the MS currently holds. For AES-CCM mode, when more than half the available PN numbers in the 31 bit PN number space are exhausted, the MS shall schedule a future Key Request in the same fashion as if the key lifetime was approaching expiry.The operation of the TEK state machine's Key Request scheduling algorithm, combined with the BS's regimen for updating and using an SAID's keying material (see 7.4), ensures that the MS will be able to continually exchange encrypted traffic with the BS.~~

~~The operation of the TEK state machine's Key Request scheduling algorithm, combined with the BS's regimen for updating and using an SAID's keying material (see 7.4), ensures that the SS will be able to continually exchange encrypted traffic with the BS.~~

~~A TEK state machine remains active as long as~~
~~a) the MS is authorized to operate in the BS's security domain, i.e., it has a valid AK, and~~
~~b) the MS is authorized to participate in that particular SA, i.e., the BS continues to provide fresh keying material during rekey cycles.~~

7.2.2.1 MS authorization and AK exchange

MS authorization procedure in PKMv2 constitutes three authorization exchange ways as shown in Fig xxx. Three authorization exchange ways are the RSA-based authorization process (_), the EAP-based authorization process (_ or _´), and the MS's Authorization Key Generationprocess (_).

There are PKMv2 RSA-Request, PKMv2 RSA-Reply, PKMv2 RSA-Reject, and PKMv2 RSA-Acknowledgement messages needed in the RSA-based authorization process (_). Both MS and BS shall share the same PAK derived from the pre-PAK.

There are two kinds of procedures in the EAP-based authorization process; the EAP Transfer procedure (_) to transfer only EAP payload and the Protected EAP Transfer procedure (_´) to assure secure EAP payload transfer. Both MS and BS can share the same PMK derived from the AAA-key. Depending on the EAP protocol, gowever, the EAP method used can not yield an AAA-key and both of them shall not share the PMK.

There are PKMv2 Authorization-Challenge, PKMv2 Authorization-Request, and PKMv2 Authorization-Reply, and PKMv2 Authorization-Reject messages needed in the MS's Authorization Key Generationprocess (_). Both MS and BS shall share the same AK derived from the PAK or/and the PMK, obtained in the RSA-based authorization process and the EAP-based authorization process, respectively.

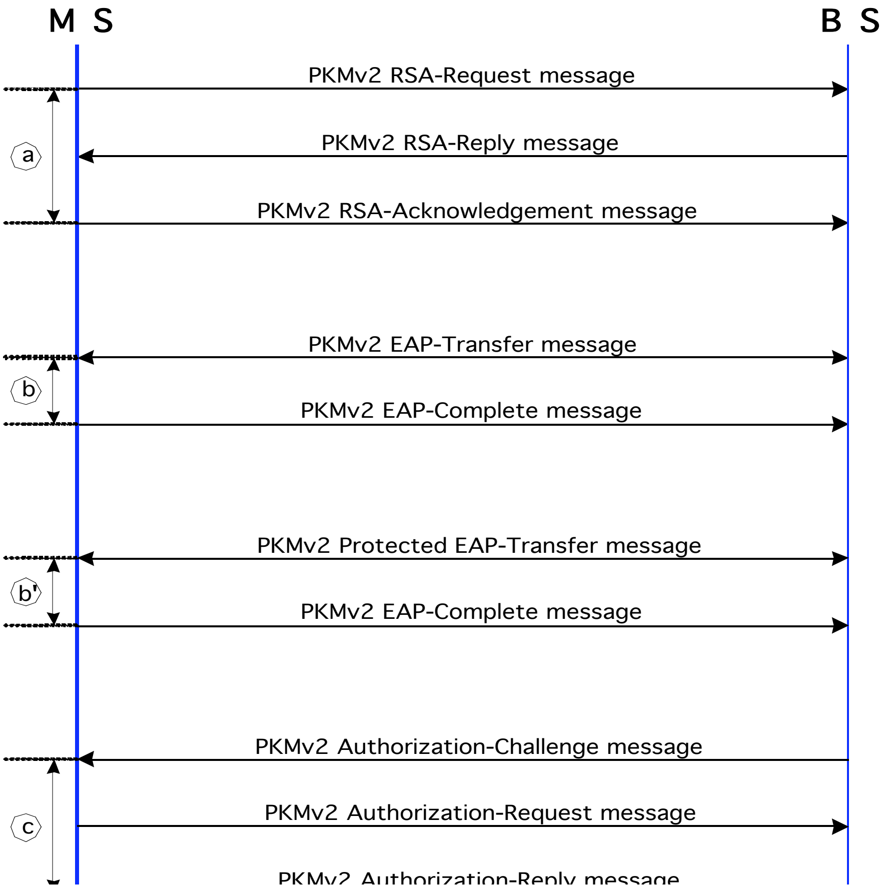

MS
BS
PKMv2 RSA-Request message
PKMv2 RSA-Reply message
PKMv2 RSA-Acknowledgement message
a
PKMv2 EAP-Transfer message
PKMv2 EAP-Complete message
b
PKMv2 Protected EAP-Transfer message
PKMv2 EAP-Complete message
b'
PKMv2 Authorization-Challenge message
PKMv2 Authorization-Request message
PKMv2 Authorization-Reply message
c

Figure xxx-MS Authorization Procedure in PKMv2

In addition, several combinations of authorization process can be executed to finally obtain the AK in PKMv2 as follows.

- Only RSA-based authorization process (_) and MS's Authorization Key Generationprocess (_)
- Only EAP-based authorization process (EAP Transfer: _) and MS's Authorization Key Generationprocess (_)
- RSA-based authorization process (_), EAP-based authorization process (EAP Transfer: _), and MS's Authorization Key Generationprocess (_)
- RSA-based authorization process (_), EAP-based authorization process (Protected EAP Transfer: _´), and MS's Authorization Key Generationprocess (_)

One of above several authorization processes shall be selected by the Authorization Policy Support field included in the SBC-REQ and SBC-RSP messages.

MS authorization, controlled by the PKMv2 Authorization state machine, is in process as follows.

[Add the whole contents in 7.2.2.1 into 7.2.2.3 as follows]

7.2.2.3 ~~Associations~~ Security Associations

Upon achieving authorization, an MS starts a separate TEK state machine for each of the SAIDs identified in the Authorization Reply message. Each TEK state machine operating within the MS is responsible for managing the keying material associated with its respective SAID. TEK state machines periodically send Key Request messages to the BS, requesting a refresh of keying material for their respective SAIDs.

The BS responds to a Key Request with a Key Reply message, containing the BS's active keying material for a specific SAID.

~~The TEK is encrypted using appropriate KEK derived from the AK.~~

TEKs and KEKs may be either 64 bits or 128 bits long. SAs employing any ciphersuite with a basic block size of 128 bits shall use 128 bit TEKs and KEKs. Otherwise 64 bit TEKs and KEKs shall be used. The name TEK-64 is used to denote a 64 bit TEK and TEK-128 is used to denote a 128 bit TEK. Similarly, KEK-64 is used to denote a 64 bit KEK and KEK-128 is used to denote a 128 bit KEK.

For SAs using a ciphersuite employing DES-CBC, the TEK in the Key Reply is triple DES (3-DES) (encrypt-decrypt-encrypt or EDE mode) encrypted, using a two-key, 3-DES KEK derived from the AK.

For SAs using a ciphersuite employing 128 bits keys, such as AES-CCM mode, the TEK in the key Reply is AES encrypted using a 128 bit key derived from the AK and a 128 bit block size.

Note that at all times the BS maintains two active sets of keying material per SAID. The lifetimes of the two generations overlap such that each generation becomes active halfway through the life of it predecessor and expires halfway through the life of its successor. A BS includes in its Key Replies both of an SAID's active generations of keying material.

~~The Key Reply provides the requesting SS, in addition to the TEK and CBC initialization vector, the remaining lifetime of each of the two sets of keying material. The receiving SS uses these remaining lifetimes to estimate when the BS will invalidate a particular TEK, and therefore when to schedule future Key Requests such that the SS requests and receives new keying material before the BS expires the keying material the SS currently holds.~~

For SAs using a ciphersuite employing CBC mode encryption the Key Reply provides the requesting MS, in addition to the TEK and CBC initialization vector, the remaining lifetime of each of the two sets of keying material. For SAs using a ciphersuite employing AES-CCM mode, the Key Reply provides the requesting MS, in addition to the TEK, the remaining lifetime of each of the two sets of keying material. The receiving MS uses these remaining lifetimes to estimate when the BS will invalidate a particular TEK, and therefore when to schedule future Key Requests such that the MS requests and receives new keying material before the BS expires the keying material the MS currently holds. For AES-CCM mode, when more than half the available PN numbers in the 31 bit PN number space are exhausted, the MS shall schedule a future Key Request in the same fashion as if the key lifetime was approaching expiry.The operation of the TEK state machine's Key Request scheduling algorithm, combined with the BS's regimen for updating and using an SAID's keying material (see 7.4), ensures that the MS will be able to continually exchange encrypted traffic with the BS.

~~The operation of the TEK state machine's Key Request scheduling algorithm, combined with the BS's regimen for updating and using an SAID's keying material (see 7.4), ensures that the SS will be able to continually exchange encrypted traffic with the BS.~~

A TEK state machine remains active as long as

a) the MS is authorized to operate in the BS's security domain, i.e., it has a valid AK, and

b) the MS is authorized to participate in that particular SA, i.e., the BS continues to provide fresh keying material during rekey cycles.

Keying material is held within associations. There are three types of association: The security associations (SA) that maintain keying material for unicast connections, group security associations (GSA) that hold keying material for multicast groups and MBSGSAs which hold keying material for MBS services.

Remedy 3. Corrections for PKMv2 Key Hierarchy

[Change 7.2.2.2: as follows]
7.2.2.2.1 ~~Certificated RSA authorization~~ RSA-based authorization

When the RSA-based authorization is negotiated as authorization policy, the PKMv2 RSA-Request, the PKMv2 RSA-Reply, the PKMv2 RSA-Reject, and the PKMv2 RSA-Acknowledgement messages are used to share the pre-PAK.

The pre-PAK (Primary Authorization Key) is sent by the BS to the MS encrypted with the public key from the certificate. Pre-PAK is mainly used to generate the PAK. The optional EIK for ~~EAP exchange~~ the Protected EAP-Transfer message (see 7.2.2.2.2) are also generated from pre-PAK:

~~|~~EIK | PAK = Dot16KDF(pre-PAK, ~~SSID |~~ "EIK+PAK", 288)

PAK will be used to generate the AK (see below) if RSA authorization was used. PAK is 160 bits long.

7.2.2.2.2 ~~EAP authentication~~ EAP-based authorization

There are two kinds of EAP-based authorization; only EAP exchange way (using the PKMv2 EAP-Transfer message) and EAP exchange way based on RSA exchange (using the PKMv2 Protected EAP-Transfer message).

In case of the only EAP exchange way, the MS's user authentication is achieved by transferring only EAP payload between a MS and the BS.

Contrary to the only EAP exchange way, in case of the EAP exchange way based on RSA exchange, the MS's user authentication is executed by exchanging PKMv2 Protected EAP-Transfer messages. ~~If a mutual authorization took place before the EAP exchange, the EAP messages~~ These messages may be protected using EIK~~ - EAP Integrity Key~~ (EAP Integrity Key) derived from pre-PAK (see 7.2.2.2.1). EIK ~~and EEK are~~ is 128 bits long.

The product of the EAP exchange which is transferred to ~~802.16~~ MAC privacy sub-layer is the AAA-key. This key is derived (or may be equivalent to the 512-bits Master Session Key (MSK)). This key is known to the AAA server, to the Authenticator* (transferred from AAA server) and to the MS. The MS and the authenticator (the serving BS or certain network node) derive a PMK (Pairwise Master Key) by truncating the AAA-key after 160 bits.

The PMK derivation from the AAA-key is as follows:

PMK = truncate (AAA-key, 160)

If more keying material is needed for future link ciphers, the key length of the PMK may be increased.

7.2.2.2.3 Authorization Key (AK) derivation

The AK will be derived by the authenticator and the MSS from the PMK (from EAP exchange) and the PAK (from RSA exchange). ~~Note that PAK can be used only in initial network entry. In cases of HO and re-authentication: Only EAP keys are applicable.~~ Note that PAK or/and PMK can be used according to the value of Authorization Policy Support field included in the SBC-REQ/RSP messages. The authorization policy shall be negotiated between MS and BS before achieving the authorization procedure, irrespective of case of initial network entry, reentry, and HO.

The exclusive-or (XOR: ⊕) value of PAK and PMK is mainly used to generate the AK. The only PAK is used to derive the AK in case of achieving RSA-based authorization procedure. On the contrary, the only PMK is used in case of executing EAP-based authorization procedure.

```
~~If (PAK and PMK)~~
    ~~AK <= Dot16KDF (PMK, SSID|BSID|PAK|"AK", 160)~~
~~Else~~
    ~~If (PAK)~~
        ~~AK <= Dot16KDF (0, SSID|BSID|PAK|"AK", 160)~~
    ~~Else~~
        ~~AK <= Dot16KDF (PMK, SSID|BSID| "AK", 160);~~
    ~~Endif~~
~~Endif~~
```

```
If (RSA-based authorization and EAP-based authorization)
        AK <= Dot16KDF (PAK⊕PMK, SS_NONCE|BS_ NONCE|SSID|BSID|“AK”, 160)
Else if (RSA-based authorization)
        AK <= Dot16KDF (PAK, SS_ NONCE|BS_ NONCE|SSID|BSID|“AK”, 160)
Else if (EAP-based authorization)
        AK <= Dot16KDF (PMK, SS_ NONCE|BS_ NONCE|SSID|BSID|“AK”, 160)
```

7.2.2.2.7 Group Traffic Encryption Key (GTEK)

The GTEK is used to encrypt multicast data packets and it is shared between all MSSs that belong to the multicast group. There are 2 GTEKs per GSA.

The GTEK is randomly generated at the BS and is encrypted using ~~AES_KEY_WRAP~~ same algorithms applied to TEK encryption and transmitted to the MS in multicast or unicast messages. ~~In multicast the message will be encrypted by the GKEK. In unicast, it will be encrypted by the KEK.~~ The GTEK will be encrypted by the GKEK.

7.2.2.2.10 Key Hierarchy

Figure 131 outlines the process to calculate the AK when the RSA-based authorization process has taken place, but where the EAP-based authentication process hasn't taken place, or the EAP method used has not yielded an AAA-key:

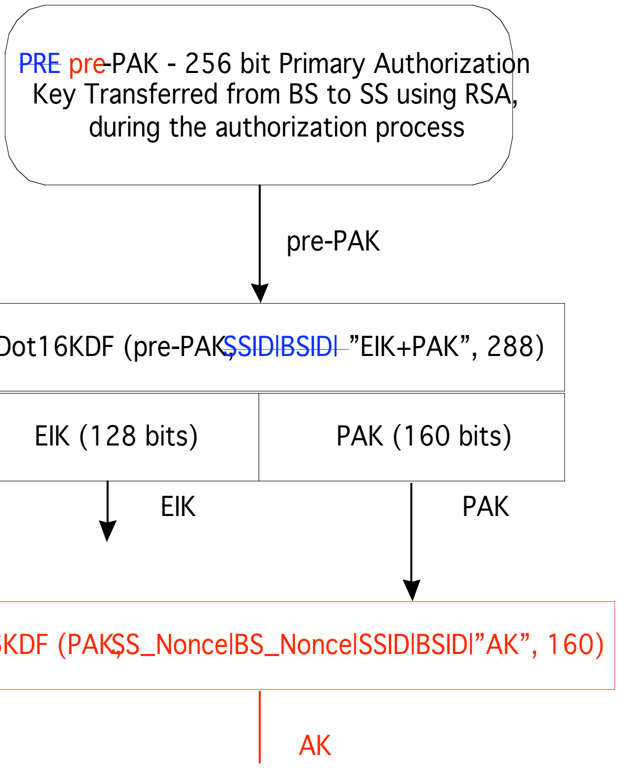

~~PRE~~ pre-PAK - 256 bit Primary Authorization Key Transferred from BS to SS using RSA, during the authorization process
pre-PAK
Dot16KDF (pre-PAK, ~~SSID|BSID|~~ ”EIK+PAK”, 288)
EIK (128 bits)
PAK (160 bits)
EIK
PAK
Dot16KDF (PAK, SS_Nonce|BS_Nonce|SSID|BSID|”AK”, 160)
AK

Figure 131-AK with the only RSA-based ~~only~~ authorization process

Figure 132 outlines the process to calculate the AK when both the RSA-based authorization exchange has taken place, yielding a PAK and the EAP based authentication exchange has taken place, yielding an AAA-key:

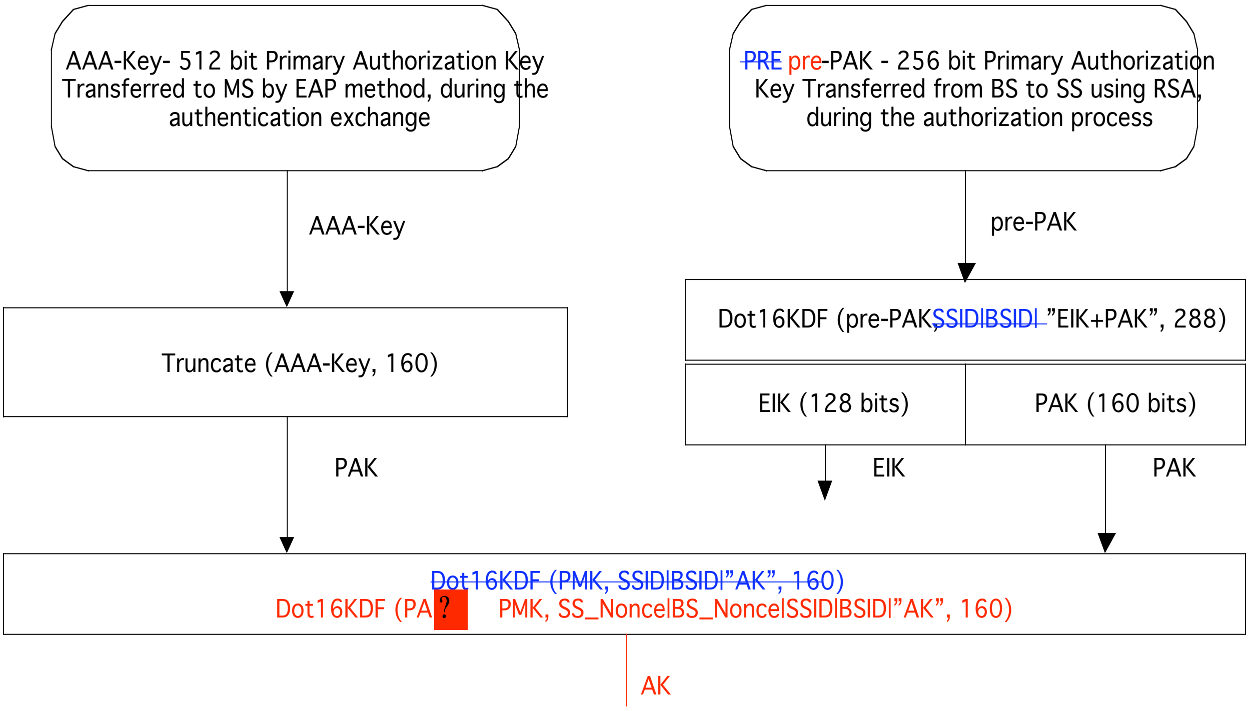

AAA-Key- 512 bit Primary Authorization Key Transferred to MS by EAP method, during the authentication exchange
AAA-Key
Truncate (AAA-Key, 160)
PAK
~~PRE~~ pre-PAK - 256 bit Primary Authorization Key Transferred from BS to SS using RSA, during the authorization process
pre-PAK
Dot16KDF (pre-PAK~~SSID|BSID|~~"EIK+PAK", 288)
EIK (128 bits)
PAK (160 bits)
EIK
PAK
~~Dot16KDF (PMK, SSID|BSID|"AK", 160)~~
Dot16KDF (PA? PMK, SS_Nonce|BS_Nonce|SSID|BSID|"AK", 160)
AK

Figure 132-AK with RSA and EAP authorization process

Figure 133 outlines the process to calculate the AK when only the EAP based authentication exchange has taken place, yielding an AAA-key:

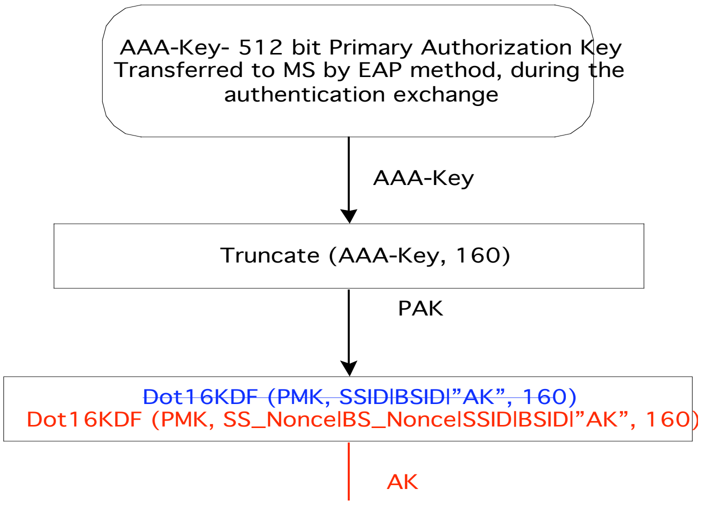

AAA-Key- 512 bit Primary Authorization Key Transferred to MS by EAP method, during the authentication exchange
AAA-Key
Truncate (AAA-Key, 160)
PAK
~~Dot16KDF (PMK, SSID|BSID|"AK", 160)~~
Dot16KDF (PMK, SS_Nonce|BS_Nonce|SSID|BSID|"AK", 160)
AK

Figure 133-AK with the only EAP-based ~~only authentication~~ authorization process

[Change 7.2.2.4.1 as follows]

7.2.2.4.1 AK Context

The context of AK includes all the parameters connected to AK and keys derived directly from it.

When one parameter from this context expires, a new AK should be obtained in order to start a new context.

Obtaining of new AK means re-authentication - doing ~~the whole EAP and/or PAK~~ the RSA-based authorization procedure or/and the EAP-based authorization procedure due to ~~the authorization policies~~ the value of the Authorization Policy Support field negotiated between the MS and BS until obtaining a new PMK and/or PAK which AK may be derived from.

Derivation of AK after HO is done separately in the MS and network from ~~a common~~ PMK, PAK, SS_Nonce, BS_Nonce, SSID, and BSID. ~~The PMK and/or PAK may be used to derive keys to several BSs sharing the same PMK and/or PAK.~~ The same PAK or/and PMK can be shared among several BSs.

In HO scenario, if the MS was previously connected to the TBS, the derived AK will be identical to the last one, as long as the PAK or PMK stays the same. In order to maintain security in this scenario: the context of the AK must be cached by both sides and to be used from the point it stopped, if context lost by one side, re-authentication is needed to establish new PAK, PMK and new AK context.

The AK context is described in the table:

Table 133-AK Context in PKMv2

Parameter	Size	Usage
Primary AK (PAK)	160bits	A key yielded from the ~~mutual authorization exchange~~ RSA-based authorization. ~~Only present at initial network entry and only if the certificated RSA exchange took place, as a result of the mutual authorization policy negotiation.~~
~~PAKID~~	~~64bits~~	~~Derived from the mutual authorization, present when PAK is present.~~
PAK Sequence Number	64bits	PAK sequence number, when the RSA-based authorization is achieved.
PAK lifetime		~~Derived from the mutual authorization, present when PAK is present.~~ PAK sequence number, when the RSA-based authorization is achieved.
PMK	160bits	A key yielded from the EAP-based authentication (only if EAP protocol generates the AAA-key)..
PMK lifetime		~~The lifetime of PMK derived from EAP.~~ PMK sequence number, when the EAP-based authorization is achieved and the AAA-key is obtained.
~~PMKID~~	~~64bits~~	~~hash 64(EAP session-id)~~
AK	160bits	~~The authentication key, calculated as f(PAK,PMK), if only EAP, AK=f(PMK).~~ The authorization key, calculated as defined in 7.2.2.2.3
~~AKID~~	~~64bits~~	~~Calculated according to the keys that contributed to AK:~~ ~~-If AK=f(PMK,PAK) then AKID=hash 64(EAP sessionid | PAKID | BSID)~~ ~~-If AK=f(PMK) then AKID=hash 64(EAP session-id | BSID)~~ ~~-If AK=PAK then AKID = PAKID~~
AK Sequence Number	64bits	AK sequence number
AK lifetime		~~This is the time this key is valid, it is calculated AK lifetime= MIN(PAK lifetime, PMK lifetime) – when this expires re-authentication is needed.~~ AK lifetime – when this expires, MS's Re-authorization Key process is needed.
H/OMAC_KEY_U	160 bits/128 bits	The key which is used for signing UL management messages.
H/OMAC_PN_U	32 bits	Used to avoid UL replay attack on management messages – when this expires re-authentication is needed.
H/OMAC_KEY_D	160 bits/128 bits	The key which is used for signing DL management messages.
H/OMAC_PN_D	32 bits	Used to avoid DL replay attack on management messages – when this expires re-authentication is needed.
KEK	160 bits	Used to encrypt ~~transport keys~~ TEK or GKEK from the BS to the SS.

Remedy 4. Corrections for Adaptation the PKMv1 Messages to PKMv2

[Delete sub-clauses 6.3.2.3.9.5 and 6.3.2.3.9.6]

~~6.3.2.3.9.5 Key Request message~~

~~Table 31-Key Request attributes~~

~~Attribute~~	~~Contents~~
~~Key Sequence Number~~	~~AK sequence number~~
~~AKID~~	~~This identifies the AK to the BS that was used for protecting this message.~~
~~NonceSS~~	~~A number chosen by the SS (once per protocol run). It can be counter or a random number.~~
~~SAID~~	~~Security association identifier.~~
~~HMAC-Digest~~	~~Keyed SHA message digest.~~

~~6.3.2.3.9.6 Key Reply message~~

~~Table 31-Key Reply attributes~~

~~Attribute~~	~~Contents~~
~~Key Sequence Number~~	~~AK sequence number~~
~~AKID~~	~~This identifies the AK to the BS that was used for protecting this message.~~
~~NonceSS~~	~~A number chosen by the SS (once per protocol run). It can be counter or a random number. This is returned by BS to MS~~.
~~SAID~~	~~Security association identifier.~~
~~TEK-Parameters~~	~~“Older” generation of key parameters relevant to SAID.~~
~~TEK-Parameters~~	~~“Newer” generation of key parameters relevant to SAID.~~
~~HMAC-Digest~~	~~Keyed SHA message digest.~~

6.3.2.3.9.22 PKMv2 Key-Request message

A MS sends a PKMv2 Key-Request message to the BS to request new TEK (or GTEK) and traffic keying material.

Code: 24

Attributes are shown in Table 37l.

Table 37l-PKMv2 Key-Request attributes

Attribute	Contents
Key Sequence Number	AK sequence number
SAID	Security association identifier
MS_ Nonce	A 64 bit random number generated in a MS
OMAC Digest	Message Digest calculated using AK

The MS_ Nonce shall be included to protect the replay attack.

The OMAC-Digest attribute shall be the final attribute in the message’s attribute list.

Inclusion of the OMAC digest allows the MS and BS to authenticate the PKMv2 Key-Request message. The OMAC-Digest’s authentication key is derived from the AK.

6.3.2.3.9.23 PKMv2 Key-Reply message

The BS responds to a MS’s PKMv2 Key-Request message with a PKMv2 Key-Reply message.

Code: 25

Attributes are shown in Table 37m.

Table 37m-PKMv2 Key-Reply attributes

Attribute	Contents
Key Sequence Number	AK sequence number
SAID	Security association identifier
TEK-Parameters	"Older" generation of key parameters relevant to SAID
TEK-Parameters	"Newer" generation of key parameters relevant to SAID
BS_ Nonce	A 64 bit random number generated in the BS
OMAC Digest	Message Digest calculated using AK

The TEK-Parameters and the SAID attributes are as defined in 6.3.2.3.9.5.

The BS_ Nonce shall be included to protect the replay attack.

The OMAC-Digest attribute shall be the final attribute in the message's attribute list.

Inclusion of the OMAC digest allows the MS and BS to authenticate the PKMv2 Key-Reply message. The OMAC-Digest's authentication key is derived from the AK.

6.3.2.3.9.24 PKMv2 Key-Reject message

The BS responds to a MS's PKMv2 Key-Request message with a PKMv2 Authorization-Reject message if the BS rejects the MS's traffic keying material request.

Code: 26

Attributes are shown in Table 37n.

Table 37n-PKMv2 Key-Reject attributes

Attribute	Contents
Key Sequence Number	AK sequence number
SAID	Security association identifier
Error-Code	Error code identifying reason for rejection of the PKMv2 Key-Request message
Display-String (optional)	Display string containing reason for the PKMv2 Key-Request message
BS_ Nonce	A 64 bit random number generated in the BS
OMAC Digest	Message Digest calculated using AK

The BS_ Nonce shall be included to protect the replay attack.

The OMAC-Digest attribute shall be the final attribute in the message's attribute list.

Inclusion of the OMAC digest allows the MS and BS to authenticate the PKMv2 Key-Reject message. The OMAC-Digest's authentication key is derived from the AK.

6.3.2.3.9.25 PKMv2 SA-Addition message

This message is sent by the BS to the SS to establish one or more additional SAs.

Code: 27

Attributes are shown in Table 37o.

Table 37o-PKMv2 SA-Addition attributes

Attribute	Contents
Key Sequence Number	AK sequence number
(one or more) SA-Descriptor(s)	Each compound SA-Descriptor attribute specifies an SA idenfier (SAID) and additional properties of the SA
BS_ Nonce	A 64 bit random number generated in the BS
OMAC Digest	Message Digest calculated using AK

The BS_Nonce shall be included to protect the replay attack.

The OMAC-Digest attribute shall be the final attribute in the message's attribute list.

Inclusion of the OMAC digest allows the MS and BS to authenticate the PKMv2 SA-Add message. The OMAC-Digest's authentication key is derived from the AK.

6.3.2.3.9.26 PKMv2 TEK-Invalid message

The BS sends a PKMv2 TEK-Invalid message to a client MS if the BS determines that the MS encrypted an uplink PDU with an invalid TEK (i.e., an SAID's TEK key sequence number), contained within the received packet's MAC Header, is out of the BS's range of known, valid sequence numbers for that SAID.

Code: 28

Attributes are shown in Table 37p.

Table 37p-PKMv2 TEK-Invalid attributes

Attribute	Contents
Key Sequence Number	AK sequence number
SAID	Security Association Identifier
Error-Code	Error code identifying reason for PKMv2 TEK-Invalid message
Display-String (optional)	Display string containing reason for the PKMv2 TEK-Invalid message
BS_Nonce	A 64 bit random number generated in the BS
OMAC Digest	Message Digest calculated using AK

The BS_Nonce shall be included to protect the replay attack.

The OMAC-Digest attribute shall be the final attribute in the message's attribute list.

Inclusion of the OMAC digest allows the MS and BS to authenticate the PKMv2 SA-Add message. The OMAC-Digest's authentication key is derived from the AK.

[Change sub-clauses 6.3.2.3.9.17 as follows]

~~6.3.2.3.9.17 Group Key Update Command message~~

6.3.2.3.9.27 PKMv2 Group-Key-Update-Command message

This message is sent by BS to push the GTEK and/or GKEK parameters to MSs served with the specific multicast service or broadcast service.

Code: 29

Attributes are shown in Table 37q.

Table ~~37g~~ 37q –~~Key update command attributes~~ PKMv2 Group Key Update Command attributes

Attribute	Contents
Key-Sequence-Number	Authorization key sequence number

GSAID	Group Security Association ID
Key Push Modes	Usage code of Key Update Command message
Key Push Counter	Counter one greater than that of older generation
GTEK-Parameters	“Newer” generation of key parameters relevant to GSAID
GKEK-Parameters	Group Key Encryption Key protected by KEK derived from shared AK and other GKEK parameter e.g. Key lifetime.
OMAC/HMAC-Digest	Message integrity code of this message

GSAID is SAID for the multicast group or the broadcast group. The type and length of the GSAID is equal to ones of the SAID.

There are two types in the Group Key Update Command message, GKEK update mode and GTEK update mode. The former is used to update GKEK and the latter is used to update GTEK for the multicast service or the broadcast service. Key Push Modes indicates this usage code of the Group Key Update Command message. The Group Key Update Command message for the GKEK update mode is carried on the Primary Management connection, but one for the GTEK update mode is carried on the Broadcast connection. A few attributes in the Group Key Update Command message shall not be used according this Key Push Modes attribute's value. See 11.9.33 for details.

Key Push Counter is used to protect for replay attack. This value is one greater than that of older generation.

The Group Key Update Command message contains only newer generation of key parameters, because this message inform an MSS next traffic key material. The GTEK-Parameters attribute is a compound attribute containing all of the keying material corresponding to a newer generation of a GSAID's GTEK. This would include the GTEK, the GTEK's remaining key lifetime, the GTEK's key sequence number, and the cipher block chaining (CBC) initialization vector. The GTEK is TEK for the multicast group or the broadcast group. The type and length of the GTEK is equal to ones of the TEK. The GKEK (Group Key Encryption Key) can be randomly generated from a BS or an ASA server. The GKEK should be identically shared within the same multicast group or the broadcast group. The GTEK is encrypted with GKEK for the multicast service or the broadcast service. GKEK parameters contain the GKEK encrypted by the KEK and GKEK lifetime. See 7.5.4.4 for details.

The OMAC/HMAC-Digest attribute shall be the final attribute in the message's attribute list. Inclusion of the keyed digest allows the receiving client to authenticate the Group Key Update Command message. The OMAC/HMAC-Digest's authentication key is derived from the AK for the GKEK update mode and GKEK for the GTEK update mode. See 7.5.4.3 for details.

Remedy 5. Corrections for 3 Way SA-TEK Exchange

[Delete sub-clause 6.3.2.3.19]

~~6.3.2.3.9.19 SA-Chanllenge message~~

~~The BS transmits the SA-Challenge message as a first step in the 3-way handshake at initial network entry and at reauthorization. It identifies an AK to be used for the Secure Association, and includes a random number challenge to be included by the MSS in its SA-TEK-Request.~~

~~Table 37i –SA-Challenge message attributes~~

~~Attribute~~	~~Contents~~
~~RandomBS~~	~~A freshly generated random number of 64bits~~
~~AKID~~	~~This identifies the AK to the BS that was used for protecting this message.~~
~~OMAC/HMAC~~	~~Message integrity tuple for this message~~

[Delete sub-clause 6.3.2.3.20]

~~6.3.2.3.9.20 SA-Chanllenge message~~

~~The MSS transmits the SA-TEK-Request message after receipt and successful HMAC/OMAC verification of an SA-Challenge from the BS. The SA-TEK_Request proves liveliness of the SS and its possession of the AK . If this message is being generated during initial network entry, then it constitutes a request for SADescriptors identifying the primary and static SAs and GSAs the requesting SS is authorized to access and their particular properties (e.g., type, cryptographic suite).~~

~~If this message is being generated upon HO, then it constitutes a request for establishment (in the target BS) of TEKs, GTEKs and GKEKs at the MSS and renewal of active primary, static and dynamic SAs and associated SAIDs used by the MSS in its previous serving BS.~~

~~Table 37j –SA-TEK-Request message attributes~~

~~Attribute~~	~~Contents~~
~~NonceSS~~	~~A 64-bit number chosen by the SS (once per protocol run). It can be a counter or a random number.~~
~~RandomBS~~	~~A freshly generated random number of 64bits~~
~~AKID~~	~~This identifies the AK to the BS that was used for protecting this message.~~
~~Security_Capabilities~~	~~Describes requesting MSS's security capabilities~~
~~OMAC/HMAC~~	~~Message integrity tuple for this message~~

[Delete sub-clause 6.3.2.3.21]

~~6.3.2.3.9.21 SA-TEK-Response message~~

~~The BS transmits the SA-TEK-Response message as a second step in the 3-way handshake.~~

~~Table 37k –SA-TEK-Response message attributes~~

~~Attribute~~	~~Contents~~
~~NonceSS~~	~~The number received from the MS~~
~~RandomBS~~	~~A freshly generated random number of 64bits This is optional~~
~~AKID~~	~~This identifies the AK to the BS that was used for protecting this message.~~
~~SA_TEK_Update~~	~~A compound TLV list each of which specifies an SA identifier (SAID) and additional properties of the SA that the MSS is authorized to access. Additionally, in case of HO, for each active SA in previous serving BS, corresponding TEK, GTEK and GKEK parameters are also included.~~
~~OMAC/HMAC~~	~~Message integrity tuple for this message~~

[Delete sub-clause 6.3.2.3.22]

~~6.3.2.3.9.22 SA-TEK-Update message~~

~~A compound TLV list each of which identifies the primary and static SAs, their SA identifiers (SAID) and additional properties of the SA (e.g., type, cryptographic suite) that the MSS is authorized to access. In case of HO, the details of any Dynamic SAs that the requesting MSS was authorized in the previous serving BS are also included.~~

~~Additionally, in case of HO, for each active SA in previous serving BS, corresponding TEK, GTEK and GKEK parameters are also included. Thus, SA_TEK_Update provides a shorthand method for renewing active SAs used by the MSS in its previous serving BS. The TLVs specify SAID in the target BS that shall replace active SAID used in the previous serving BS and also "older" TEK-Parameters and "newer" TEKParameters relevant to the active SAIDs. The update may also include multicast /broadcast Group SAIDs (GSAIDs) and associated GTEK-Paramters pairs.~~

~~In case of unicast SAs, the TEK-Parameters attribute contains all of the keying material corresponding to a particular generation of an SAID's TEK. This would include the TEK, the TEK's remaining key lifetime, its key sequence number and the cipher block chaining (CBC) initialization vector. The TEKs are encrypted with KEK.~~

~~In case of group or multicast SAs, the TEK-Parameters attribute contains all of the keying material corresponding to a particular generation of a GSAID's GTEK. This would include the newer GTEK parameter pairs, GTEK's remaining key lifetime, the GTEK's key sequence number, and the cipher block chaining (CBC) initialization vector. The type and length of the GTEK is equal to ones of the TEK. The GKEK should be identically shared within the same multicast group or the broadcast group. The GTEKs and GKEKs are encrypted with KEK because they are transmitted as a unicast here.~~

~~Multiple iterations of these TLVs may occur suitable to re-creating and re-assigning all active SAs and their (G)TEK pairs for the MSS from its previous serving BS. If any of the Security Associations parameters change, then those Security Associations parameters encoding TLVs that have changed will be added.~~

~~This TLV may be sent in a single frame along with unsolicited REG-RSP.~~

~~PKMv2 Authorization Acknowledgement (Auth Ack) message~~

~~Code: X+2~~

~~Sent by the SS to BS as an acknowledgement of successful BS Authorization~~

~~Table 37k - SA-TEK-Update message attributes~~

~~Attribute~~	~~Contents~~
~~BS_RANDOM~~	~~A 64-bit random number generated by the BS.~~
~~SS_MAC_ADDRESS~~	~~Contains the SS's MAC address.~~
~~OMAC Tuple~~	~~OMAC calculated using OMAC key derived from PAK.~~

[Delete sub-clause 7.8.1]

~~7.8.1 SA-TEK-3-way handshake~~

~~Depending on mutual authorization/EAP, AK can be derived in three different ways as documented in section XXX. Before the 3-way handshake begins, the BS and MS shall both derive a shared AK, KEK and HMAC/OMAC as per 7.2.2.2.~~

~~The SA-TEK 3-way handshake sequence proceeds as follows:~~

~~1. During initial network entry or reauthorization, the BS shall send SA-Challenge to the MS after protecting it with the OMAC/HMAC tuple. If the BS does not receive SA-TEK-Request from the BS within SAChallengeTimer, it shall send another challenge. The BS may send SA-Challenge up to SAChallenge-MaxResends times. If the BS reaches its maximum number of resends, it shall discard the AK and may initiate full re-authentication or drop the MS.~~

~~2. During network re-entry or handover, the BS begins the 3-way-handshake by appending the SaChallenge TLV to the RNG-RSP. If the BS does not receive SA-TEK-Request from the BS within SaChallengeTimer, it shall discard the AK and may initiate full re-authentication or drop the MS. If the BS receives RNG-REQ during the period that SA-TEK-Request is expected, it shall send a new RNG-RSP with another SaChallenge TLV.~~

~~3. The MS shall send SA-TEK-Request to the BS after protecting it with the OMAC/HMAC. If the MS does not receive SA-TEK-~~

~~Response from the BS within SATEKTimer, it shall resend the request. The MS may resend the SA-TEK-Request up to SATEKRequestMaxResends times. If the MS reaches its maximum number of resends, it shall discard the AK and may do full re-authentication or decide to connect to another BS or take some other action. The message shall include RandomBS, NonceSS, AKID, SS's Security Capabilities and OMAC/HMAC.~~

~~4. Upon receipt of SA-TEK-Request, a BS shall confirm that the supplied AKID refers to an AK that it has available. If the AKID is unrecognized, the BS shall ignore the message. The BS shall verify the OMAC/HMAC. If the OMAC/HMAC is invalid, the BS shall ignore the message.~~

~~5. Upon successful validation of the SA-TEK-Request, the BS shall send SA-TEK-Response back to the MS. The message shall include a compound TLV list each of which identifies the Primary and static SAs, their SA identifiers (SAID) and additional properties of the SA (e.g., type, cryptographic suite) that the MS is authorized to access. In case of HO, the details of any Dynamic SAs that the requesting MS was authorized in the previous serving BS are also included.~~

~~Additionally, in case of HO, for each active SA in previous serving BS, corresponding TEK, GTEK and GKEK parameters are also included. Thus, SA_TEK_Update provides a shorthand method for renewing active SAs used by the MS in its previous serving BS. The TLVs specify SAID in the target BS that shall replace active SAID used in the previous serving BS and also "older" TEK-Parameters and "newer" TEKParameters relevant to the active SAIDs. The update may also include multicast/ broadcast Group SAIDs (GSAIDs) and associated GTEK-Paramters pairs.~~

~~In case of unicast SAs, the TEK-Parameters attribute contains all of the keying material corresponding to a particular generation of an SAID's TEK. This would include the TEK, the TEK's remaining key lifetime, its key sequence number and the cipher block chaining (CBC) initialization vector. The TEKs are encrypted with KEK.~~

~~In case of group or multicast SAs, the TEK-Parameters attribute contains all of the keying material corresponding to a particular generation of a GSAID's GTEK. This would include the GTEK, the GKEK, the GTEK's remaining key lifetime, the GTEK's key sequence number, and the cipher block chaining (CBC) initialization vector. The type and length of the GTEK is equal to ones of the TEK. The GKEK should be identically shared within the same multicast group or the broadcast group. Contrary Key Update Command, the GTEKs and GKEKs are encrypted with KEK because they are transmitted as a unicast here.~~

~~Multiple iterations of these TLVs may occur suitable to re-creating and re-assigning all active SAs and their (G)TEK pairs for the MS from its previous serving BS. If any of the Security Associations parameters change, then those Security Associations parameters encoding TLVs that have changed will be added.~~

~~The OMAC/HMAC shall be the final attribute in the message's attribute list.~~

~~6. Upon receipt of SA-TEK-Response, an MS shall verify the OMAC and ensure the presence of correct NonceSS. If the OMAC or NonceSS is invalid, the MS shall ignore the message. Upon successful validation of the received SA-TEK-Response, the MS shall install the received TEKs and associated parameters appropriately. Verification of OMAC is done as per section XXX. If RandomBS was present in SA-TEKResponse, the MS shall send SA-TEK-Confirm to the BS and an OMAC/HMAC digest.~~

[Delete sub-clause 11.7.21]

~~11.7.21 SA-TEK-Update~~

~~This field provides a translation table that allows an MSS to update its security associations and TEK pairs so that it may continue security service after a hand-over to a new serving BS.~~

~~The following TLV values shall appear in each SA TEK Update TLV~~

~~Name~~	~~Type~~	~~Length(1 byte)~~	~~Value~~
~~SA TEK Update~~	~~?~~	~~Variable~~	~~Compound~~

~~Attribute~~	~~Type~~	~~Length(1byte)~~	~~Value~~
~~SA TEK Update Type~~	~~??~~	~~1~~	~~1: TEK parameters for a SA~~ ~~2: GTEK parameters for a GSA~~ ~~3-255: *Reserved*~~
~~New SAID~~	~~20.1~~	~~2~~	~~New SAID after hand-over to new BS~~
~~Old SAID~~	~~20.1~~	~~2~~	~~Old SAID before hand-over from old BS. In case of initial network entry, old SAID is same as new SAID.~~
~~Old TEK Parameters~~	~~13/GTEK~~	~~Variable~~	~~"Older" generation of key parameters~~

	~~Type?~~		~~relevant to SAID. The Compound field contains the subattributes as defined in Table 370.~~
~~New TEK/GTEK Parameters~~	~~13/GTEK Type?~~	~~Variable~~	~~“Newer” generation of key parameters relevant to (G)SAID. The Compound field contains the subattributes as defined in Table 370.~~
~~GKEK Parameters~~	~~GKEK Type?~~	~~Variable~~	~~GKEK and its lifetime for the corresponding GTEK pair if this TLV is for a GSA.~~