Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**	
Title	**Efficient and Secure Security Framework in PKMv2**	
Data Submitted	**2005-03-15**	
Source(s)	Seokheon Cho, Sungcheol Chang, Chulsik Yoon, ETRI Sanjay Bakshi, Yigal Eliaspur, Intel Corporation Junhyuk Song, Jicheol Lee Samsung	Voice: +82-42-860-5524 Fax: +82-42-861-1966 chosh@etri.re.kr sanjay.bakshi@intel.com junhyuk.song@samsung.com
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

Sanjay Bakshi and Yigal Eliaspur
Intel Corporation

Junhyuk Song
Samsung

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, Key Request, Key Reply, Key Reject, Auth Invalid, and TEK Invalid 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 RSA-Reject message (New message is added)
     ii. PKMv2 RSA-Acknowledgement message (New message is added)
   - 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)
     ii. Protected EAP message → PKMv2 Protected-EAP-Transfer message (Name and attributes are changed)
   - 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)
     ii. PKMv2 Key-Reply message (New message is added)
     iii. PKMv2 Key-Reject message (New message is added)
   - 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)
   - 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)

- 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)

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.

3.3 Solutions

a) 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)

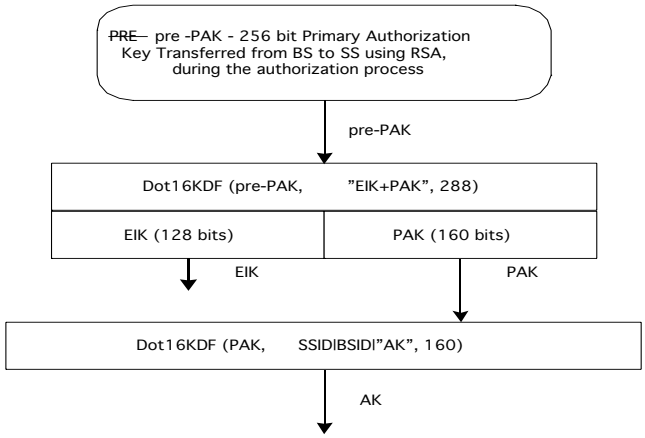

Figure -AK with the only RSA-based authorization process

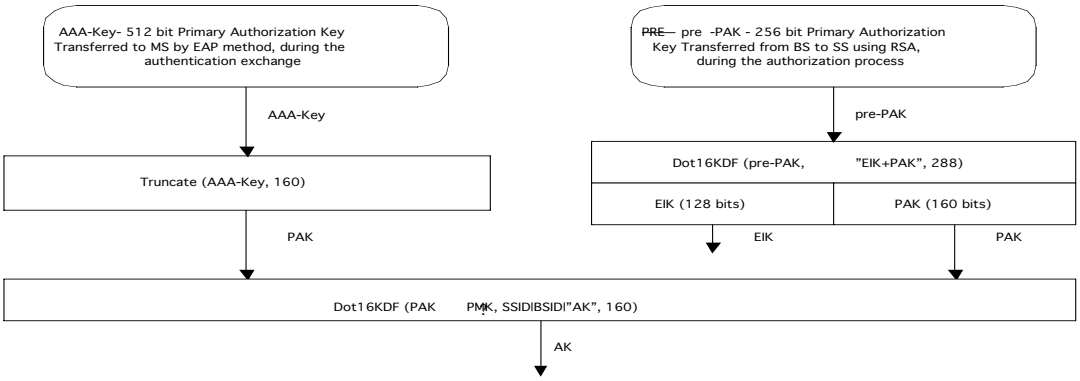

Figure -AK with RSA and EAP authorization process

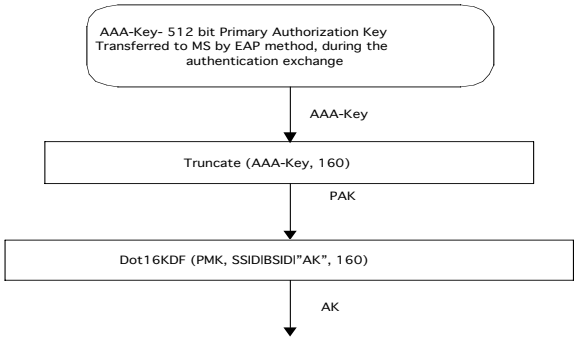

Figure -AK with the only EAP-based authorization process

- 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.

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) 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	PKMv2 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~~ 14	PKMv2 Auth-Request	PKM-REQ
~~18~~ 15	PKMv2 Auth-Reply	PKM-RSP
16	PKMv2 Auth-Reject	PKM-RSP
17	PKMv2 Auth-Ack	PKM-REQ
18	reserved	
19	PKMv2 Group Key Update Command	PKM-RSP
20	PKMv2 Protected EAP	PKM-REQ/PKM-RSP
21	PKMv2 SA-TEK-Challenge	PKM-RSP
22	PKMv2 SA-TEK-Request	PKM-REQ
23	PKMv2 SA-TEK-Response	PKM-RSP
24	PKMv2 Key-Request	PKM-REQ
25	PKMv2 Key-Reply	PKM-RSP
26	PKMv2 Key-Reject	PKM-RSP
27	PKMv2 SA-Add	PKM-RSP
28	PKMv2 TEK-Invalid	PKM-RSP
~~24~~29-255	reserved	

Remedy 2. Corrections for MS's Authorization Flow

[Change sub-clauses 6.3.2.3.9.11 as follows]
6.3.2.3.9.11 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

Attributes are shown in Table 37a.

Table 37a– PKMv2 EAP Transfer 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.

[Delete sub-clauses 6.3.2.3.9.12, 6.3.2.3.9.13, and 6.3.2.3.9.14]

[Change sub-clauses 6.3.2.3.9.15 as follows]
6.3.2.3.9.15 PKMv2 Auth-Request message

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

Code: ~~21~~ 14

Attributes are shown in Table 37e.

Table 37e-PKMv2 Auth-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~~ SS's primary SAID equal to the Basic CID

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 PKMv2 Auth-Reply message

Sent by the BS to a client MS in response to an PKMv2 Authorization Request, the PKMv2 Authorization 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 Auth-Request message, along with a random number supplied by the BS, thus enabling assurance of key liveness.

Code: ~~22~~ 15

Attributes are shown in Table 37f .

Table 37f - PKMv2 Auth-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.xx PKMv2 Auth-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: 16

Attributes are shown in Table 37c.

Table 37c-PKMv2 Auth-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.xx PKMv2 Auth-Acknowledgement message

The MS sends the PKMv2 Auth-Acknowledgement message to BS in response to a PKMv2 Auth-Reply message or a PKMv2 Auth-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: 17

Attributes are shown in Table 37d.

Table 37d-PKMv2 Auth-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 6.3.2.3.9.18 as follows]

6.3.2.3.9.18 PKMv2 Protected EAP 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~~ 20

Attributes are shown in Table 37h.

Table 37h –PKMv2 Protected EAP-Transfer message attributes

Attribute	Contents
~~Key Sequence Number~~ Key ID	~~AK Sequence Number~~ PAK ID or PMK ID
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.

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 Auth-Request, the PKMv2 Auth-Reply, the PKMv2 Auth-Reject, and the PKMv2 Auth-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 or EAP 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 or EAP 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, SSID|BSID|"AK", 160)
Else if (RSA-based authorization)
AK <= Dot16KDF (PAK, SSID|BSID|"AK", 160)
Else if (EAP-based authorization)
AK <= Dot16KDF (PMK, 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:

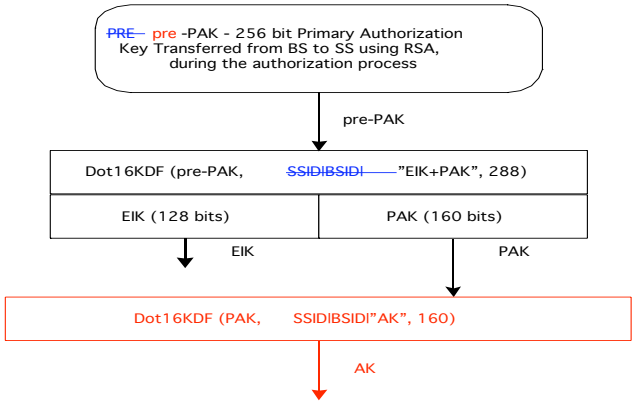

Figure 131-AK with the only RSA-based ~~only~~ authorization process

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:

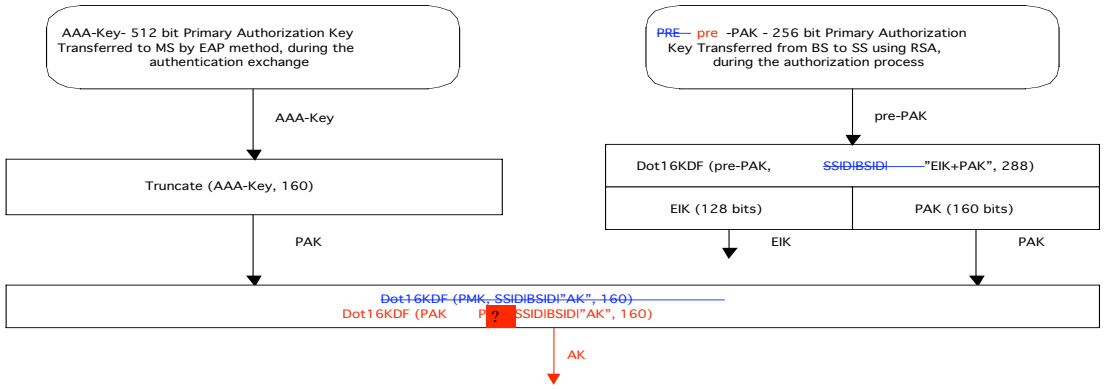

Figure 132-AK with RSA and EAP authorization process

[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:

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.xx 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 37x.

Table 37x-PKMv2 Key-Request attributes

Attribute	Contents
AK ID	AK Identifier
SAID	Security association identifier
MS_Random	A 64 bit random number generated in a MS
OMAC/HMAC Digest	Message Digest calculated using AK

The MS_Random 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.xx 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 37x.

Table 37x-PKMv2 Key-Reply attributes

Attribute	Contents
AK ID	AK Identifier
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_Random	A 64 bit random number generated in the BS
OMAC/HMAC 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_Random 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.xx 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 37x.

Table 37x-PKMv2 Key-Reject attributes

Attribute	Contents
AK ID	AK Identifier
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_Random	A 64 bit random number generated in the BS
OMAC/HMAC Digest	Message Digest calculated using AK

The BS_Random 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.xx PKMv2 SA-Add 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 37x.

Table 37x-PKMv2 SA-Addition attributes

Attribute	Contents
AK ID	AK Identifier
(one or more) SA-Descriptor(s)	Each compound SA-Descriptor attribute specifies an SA idenfier (SAID) and additional properties of the SA
BS_Random	A 64 bit random number generated in the BS
OMAC/HMAC Digest	Message Digest calculated using AK

The BS_Random 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.xx 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 37x.

Table 37x-PKMv2 TEK-Invalid attributes

Attribute	Contents
AK ID	AK Identifier
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_Random	A 64 bit random number generated in the BS
OMAC/HMAC Digest	Message Digest calculated using AK

The BS_Random 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 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: 19

Attributes are shown in Table 37g.

Table 37g – PKMv2 Group Key update command attributes

Attribute	Contents
AK ID	AK Identifier
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

[Change sub-clause 6.3.2.3.21 as follows]
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~~ BS_Random	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~~. This attribute are present in case of HO. ~~Additionally, in case of HO, for~~ For each active SA in previous serving BS, corresponding TEK, GTEK and GKEK parameters are also included.
(one or more) SA-Descriptor(s)	Each compound SA-Descriptor attribute specifies an SA idenfier (SAID) and additional properties of the SA. This attribute are present at the initial network entry.
OMAC/HMAC	Message integrity tuple for this message

[Change sub-clause 6.3.2.3.22 as follows]
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.