Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**	
Title	Security Association Establishment for PKMv2-EAP	
Date Submitted	**2004-11-04**	
Source(s)	Mi-young Yun Jung-mo Moon Chulsik Yoon Young-jin Kim, ETRI 161, Gajeong-dong,Yuseong-Gu, Daejeon, 305-350, Korea	Voice: +82-42-860-4821 Fax: +82-42-861-1966 myyun@etri.re.kr
Re:	Contribution to P802.16e/D5	
Abstract	We propose to add and modify the EAP authentication and key distribution mechanism in the current TGe document on the basis of PKMv2-EAP mechanism.	
Purpose	Adoption	
Notice	This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.	
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.	
Patent Policy and Procedures	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:chair@wirelessman.org> as early as possible, in written or electronic form, if patented technology (or technology under patent application) might be incorporated into a draft standard being developed within the IEEE 802.16 Working Group. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>.	

Security Association Establishment for PKMv2-EAP

Mi-young Yun, Jung-mo Moon, Chulsik Yoon, Young-jin Kim

ETRI

1. Purpose

In P802.16e/D5, there are ~~two alternative~~ a mechanism~~s~~ for authentication and key establishment using EAP such as ~~PKMv2-EAP mechanism (See contribution C802.16e-04/188r3 Key Hierarchy for PKMv2; currently not included in the P802.16e/D4 document, but generally agreed and accepted by the working group members) and~~ PKM-EAP mechanism via a 3-way EAP-Key exchange (See section 6.3.2.3.9.11 through 6.3.2.3.9.16, section 7.2.1.2). ~~For the simplicity, we consider using only EAP in order to authenticate and make an AK in this document. Let's say the 3-way EAP-Key exchange mechanism as a PKM-EAP mechanism, and the EAP-only mode in DJ's Key Hierarchy mechanism as a PKMv2-EAP.~~

~~Both schemes~~ It do~~es~~ not use an AAA-Key as an AK directly for the cryptography considerations. First, ~~they~~ it take~~s~~ a part of the AAA-Key as a Master-Key. Then, ~~each of them~~ it derives an authorization key from the Master-Key ~~in different ways~~. ~~In PKM-EAP of P802.16e/D4, a~~An AK is derived with the Master-Key, nonces generated by the BS and the SS and their IDs (BS-ID and SS MAC address~~)es~~. Especially nonces are delivered by the EAP-Establish-Key-Request/EAP-Establish-Key-Reply messages. ~~On the other hands, each the BS and the SS generates the AK with the AAA key and MAC addresses in PKMv2-EAP. Namely, the main difference between two mechanisms is in the key derivation method.~~

~~Additionally~~By the way, PKMv2 assumes that there is a different AK per {BS, SS} Tuple and an SS does not need to enter into a key establishment procedure with the target BS during a handover where pre-authentication is used. However, PKM-EAP needs to exchange PKM messages for a new AK after a handover because it is necessary to generate nonces in the new BS and the SS for AK derivation. So, it can take more time to process handover.

~~For~~ According to the PKMv2's ~~consistency~~keying model(See contribution C802.16e-04/188r3 Key Hierarchy for PKMv2; currently not included in the P802.16e/D5 document, but generally agreed and accepted by the working group members), we think it is the better not to take 3-way handshake in PKM-EAP.~~a key generation mechanism in PKMv2-EAP.~~

~~W~~We still need to define security capabilities negotiation in order to complete authentication and key generation mechanism using the EAP-only mode even if the PKMv2 Key Hierarchy document proposed by David Johnston of Intel (C802.16e-04/188r4) is accepted. And, if SS does not enter into key establishment procedure, then there is no way to exchange Security Association Descriptors with BS in 802.16e/D5.

In this proposal, we have two suggestions for ~~So, we add PKM messages for~~ the security capability negotiation. ~~In this document~~One is ~~we propose~~ to add new messages and ~~modify the current TGe document on the basis of PKMv2 mechanism.~~ the other is to modify Auth Request.

2. EAP authentication mechanism in PKMv2 and 802.16e/D5

1) PKM-EAP in P802.16e/D5

The BS and the SS each derive the EAP-Master-Key from the AAA-Key. The BS and SS exchange nonce and security capabilities. Then they make a TK (Transient Key) using PRF-384. The BS and the SS can derive KCK (Key Confirmation Key) and AK (Authorization Key) from the TK according to the rule. The KCK is used for message authentication.

After handover, PKM-EAP messages should be exchanged because the AK is generated during EAP-Establish-Key messages.

The EAP-only mode authorization flow between SS and BS and the AK derivation mechanism are shown in Figure 1 and Figure 2.

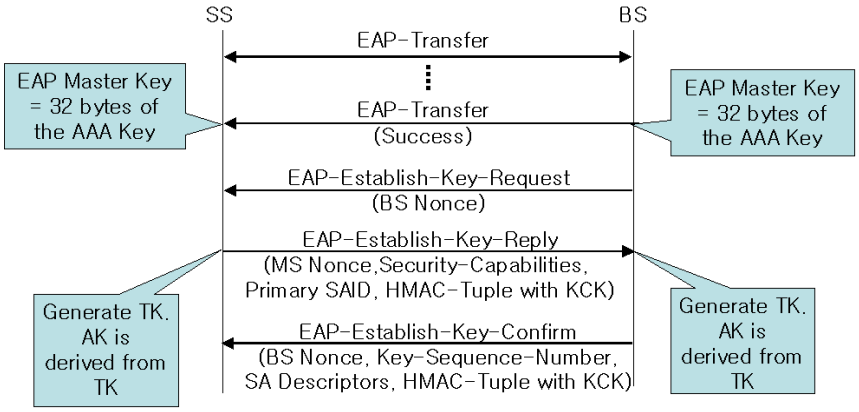

Figure 1. The EAP-based authorization flow

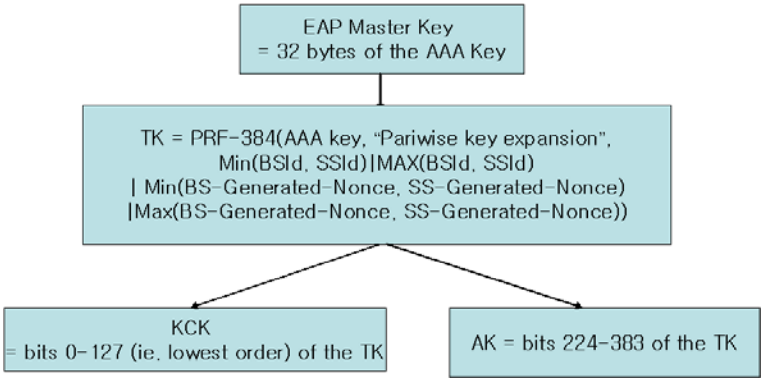

Figure 2. The AK derivation mechanism

2) PKMv2-EAP-only mode

PKMv2-EAP-only mode form the Master Key from the AAA key, and then the BS and SS each generate the AK with the AAA key and their MAC addresses.

The AAA key is generated in the AAA server and the SS as a result of an EAP based authentication exchange when the EAP-only mode is selected to provide key establishment. The Master Key, MK, is formed from the leftmost 160 bits of the AAA key. The PMK is derived from the MK and MAC addresses of the BS and the SS. And the Authorization Key is derived from the PMK using the Dot16KDF~~.~~.

New AK is generated between new BS and SS although one wants that PKM-REQ/RSP sequence may be omitted for the current HO re-entry attempt.

The PKMv2 EAP-only mode authorization flow between the SS and the BS, and the AK derivation mechanism are shown in Figure 3 and Figure 4.

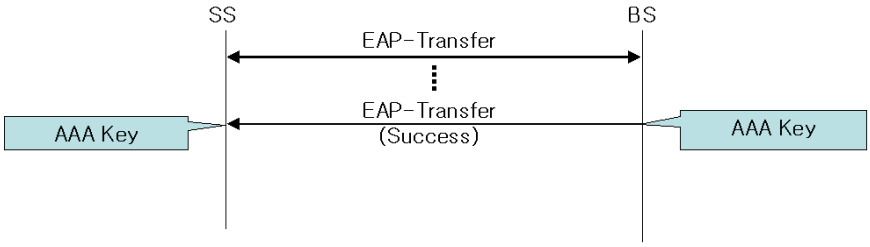

Figure 3. PKMv2 EAP-based authorization flow

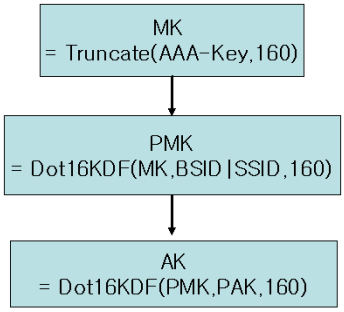

Figure 4. AK derivation mechanism in PKMv2

3. Summary of Solution

To provide an integrated and consistent way applying for various types of security modes, we propose PKMv2 EAP-only mode authorization flow and the AK derivation mechanism. To complete the EAP-only mode authentication and fast handover between the SS and the BS, it is necessary for the SS and the BS to exchange the SS's security and ciphersuite capabilities, and SA-Descriptor attributes. This information is the same as in Authorization and EAP-Establish-Key messages. It is one way to make use of existing messages but PKM-EAP adopts 3-way message handshaking. So, it is not useful for this purpose.

Accordingly, we define SA-Capability-Request and SA-Capability-Reply messages which can be validated by ~~H~~HMAC Tuple.

The ~~proposed~~ EAP-only mode authentication flow with SA-Capability messages is shown in Figure 5.

The SA-Capability negotiation flow is shown in Figure 6 when the pre-authentication is used.

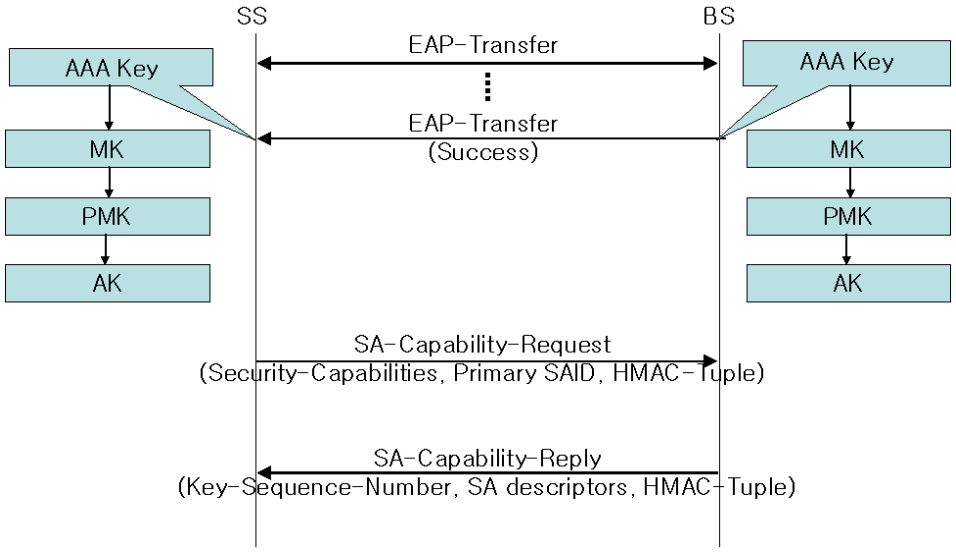

Figure 5. Proposed PKMv2-EAP-only mode authentication

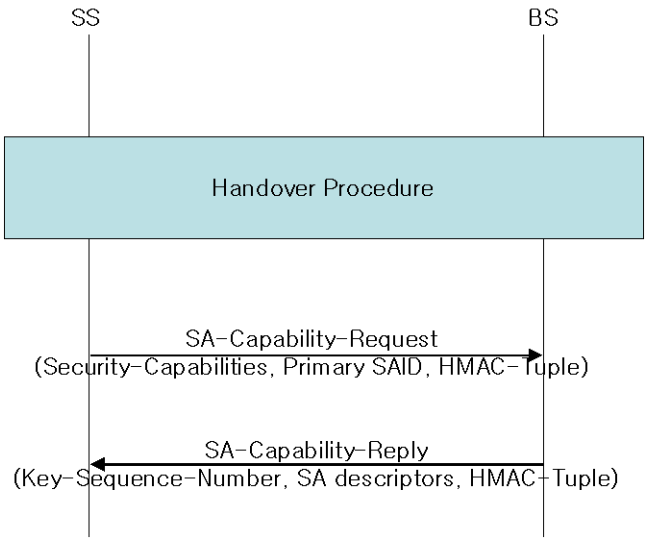

Figure 6. the flow when the pre-authentication is used

4. Specific test changes

1) Option 1

[6.3.2.3.9 Change Table 26a – PKM Message Codes]

13	EAP Transfer	PKM-REQ
14	*Reserved*	
15	EAP Establish-Key Request	PKM-RSP
16	EAP Establish-Key Reply	PKM-REQ
17	EAP Establish-Key Reject	PKM-REQ
18	EAP Establish-Key Confirm	PKM-RSP
19	Pre-Auth-Request	PKM-REQ
20	Pre-Auth-Reply	PKM-RSP
21	Pre-Auth-Reject	PKM-RSP
22	PKMv2 Auth Request	PKM-REQ
23	PKMv2 Auth Reply	PKM-RSP
24	SA Capability Request	PKM-REQ
25	SA Capability Reply	PKM-RSP
26	SA Capability Reject	PKM-RSP
27-255	Reserved	

[Add Section 6.3.2.3.9.xx SA Capability Request Message]

6.3.2.3.9.2x PKMv2 SA Capability Request message
The SS transmits the SA Capability Request message as the first step in the 2-step sequence of SA Capability Negotiation ~~after EAP-based authentication~~.
Code: 24
Attributes are shown in Table x.

Table x—SA Capability Request attributes

Attribute	Contents
Security- Capabilities	Describes SS's security and ciphersuite capabilities
Primary SAID	SS's primary SAID (equal to the Basic CID)
~~H~~MAC-Tuple	The cryptographic hash for the message. (HMAC or OMAC)

[Add Section 6.3.2.3.9.xx SA Capability Reply Message]

6.3.2.3.9.2x SA Capability Reply message
The BS transmits the SA Capability Reply message as the second step in the 2-step sequence of SA Capability Negotiation ~~after EAP-based authentication~~.
Code: 25
Attributes are shown in Table x.

Table x—SA Capability Reply attributes

Attribute	Contents
Key-Sequence-Number	Sequence Number for established AK
(one or more) SA descriptors	Each Compound SA-Descriptor attribute specifies an SAID and additional properties of the SA
~~H~~MAC-Tuple	The cryptographic hash for the message. (HMAC or OMAC)

[Add Section 6.3.2.3.9.xx SA Capability Reject Message]

6.3.2.3.9.xx SA Capability Reject message

The BS transmits the SA Capability Request message as the second step in the2-step sequence of SA Capabilities Negotiation ~~after EAP-based authentication~~.

Code: 26

Attributes are shown in Table x.

Table x—SA Capability Reject attributes

Attribute	Contents
Error-Code	Error code identifying reason for rejection of SA Capability Request message
~~H~~MAC-Tuple	The cryptographic hash for the message. (HMAC or OMAC)

[Add the following paragraph at the end of the Section 7.2.1.x Authorization via PKM Extensible Authentication Protocol:]

7.2.1.x Authorization via PKM Extensible Authentication Protocol

……

The final steps of the authorization flow:

1) The AAA key is generated in the AAA server and the SS as a result of an EAP based authentication exchange when the EAP-only mod is selected to provide key establishment. The Master Key, MK, is formed from the leftmost 160 bits of the AAA key.

2) The SS and the BS generate the PMK and AK using the Dot16KDF at each side, separately.

3) The SS sends the SA Capability Request PKM message (including Security-Capabilities, Primary SAID) to the BS. The SA Capability Request includes an HMAC Tuple TLV, which must be calculated using the AK .Upon receipt of the SA-Capability-Request, the BS validates the HMAC Tuple. In the stage of the initial authorization, the key sequence number in the HMAC Tuple can be formed by Hash (AK). If the BS cannot accept the SS's SA-Capability-Request, the BS sends SA-Capability-Reject to the SS. The BS sends the SA-Capability-Reply PKM message to supply the SS with its SA information.

2) Option 2

~~Auth-Request (version1)~~

~~FMT~~	~~1 bit~~	~~0 = Authorization based format~~ ~~1 = Security Capability based format~~
~~If(FMT == 0) {~~		
~~SS-Certificate~~		~~Contains the SS's X.509 user certificate~~
~~Security- Capabilities~~		~~Describes SS's security and ciphersuite capabilities~~
~~Primary SAID~~		~~SS's primary SAID (equal to the Basic CID)~~
~~}~~		
~~else{~~		
~~Security- Capabilities~~		~~Describes SS's security and ciphersuite capabilities~~
~~Primary SAID~~		~~SS's primary SAID (equal to the Basic CID)~~
~~MAC-Tuple~~		~~The cryptographic hash for the message. (HMAC or OMAC)~~
~~}~~		

~~Auth-Reply(version1)~~

~~FMT~~	~~1bit~~	~~0 = Authorization based format~~ ~~1 = Security Capability based format~~
~~If(FMT == 0) {~~		
~~AUTH Key~~		~~Authorization (AUTH) Key, encrypted with the target client SS's public key~~
~~Key Lifetime~~		~~AK's active lifetime~~
~~Key-Sequence-Number~~		~~Sequence Number for established AK~~
~~(one or more) SA descriptors~~		~~Each Compound SA-Descriptor attribute specifies an SAID and additional properties of the SA~~
~~}~~		
~~Else {~~		
~~Key Lifetime~~		~~AK's active lifetime~~
~~Key-Sequence-Number~~		~~Sequence Number for established AK~~
~~(one or more) SA descriptors~~		~~Each Compound SA-Descriptor attribute specifies an SAID and additional properties of the SA~~
~~MAC-Tuple~~		~~The cryptographic hash for the message. (HMAC or OMAC)~~
~~}~~		

6.3.2.3.9.20 ~~Auth-Request(Ver.2)~~ PKMv2 authorization request (auth request) message

Table 37j—PKMv2 Auth-Request attributes

Format_Indicator~~MT~~	1 bit	0 = Authorization based format 1 = Security Capability based format
If(Format_Indocator~~FMT~~ == 0) {		

SS-Random		A 64 bit random number generated in the SS
SS-Certificate		Contains the SS's X.509 user certificate
Security- Capabilities		Describes SS's security and ciphersuite capabilities
Primary SAID		SS's primary SAID (equal to the Basic CID)
}		
else{		
Security- Capabilities		Describes SS's security and ciphersuite capabilities
Primary SAID		SS's primary SAID (equal to the Basic CID)
MAC-Tuple		The cryptographic hash for the message. (HMAC or OMAC)
}		

6.3.2.3.9.21 ~~Auth-Reply(version3)~~ PKMv2 authorization reply (auth reply) message

Table 37k—PKMv2 Auth-Reply attributes

~~FMT~~Format_Indicator	1bit	0 = Authorization based format 1 = Security Capability based format
If(Format_Indicator~~FMT~~ == 0) {		
SS-Random		
BS-Random		
SS-Certificate		
AUTH Key		Authorization (AUTH) Key, encrypted with the target client SS's public key
Key Lifetime		AK's active lifetime
Key-Sequence-Number		Sequence Number for established AK
(one or more) SA descriptors		Each Compound SA-Descriptor attribute specifies an SAID and additional properties of the SA
CertiBS		The BS Certificate
SigBS		An RSA signature over all the other attributes in the message
}		
Else {		
Key Lifetime		AK's active lifetime
Key-Sequence-Number		Sequence Number for established AK
(one or more) SA descriptors		Each Compound SA-Descriptor attribute specifies an SAID and additional properties of the SA
MAC-Tuple		The cryptographic hash for the message. (HMAC or OMAC)
}		