Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	reauthorization via PKM extensible authentication protocol
Date Submitted	**2004-1-21**
Source(s)	[Feng Tian] [JianYong Chen] [Rui Li] [zte] [ZTE Plaza , Keji Road South , Hi-tech Industrial Park , Nanshan District , Shenzhen , P.R.China , 518057] Voice: [86-0755-26772017] Fax: [86-0755-26772004] [mailto:tian.feng2@zte.com.cn]
Re:	802.16e/D5a
Abstract	reauthorization via PKM extensible authentication protocol
Purpose	Adopt
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>.

reauthorization via PKM extensible authentication protocol

Feng Tian， Jianyong chen， Rui Li

In 802.16e draft , It doesn’t describe the reauthorization based on PKM extensible authentication protocol in PKMv1 . There are two phases in the authorization via PKM extensible authentication protocol. In the first phase, the BS and SS realize authentication. In the second phase, the BS and SS negotiate AK. There are two method to realize reauthorization based on PKM extensible authentication protocol according to the two phases.

The first method is that the BS periodically sends EAP-Request to SS to begin reauthorization, the request might be an EAP identity request or the beginning of an EAP method. Reauthorization is identical to authorization. This method needs to execute a full authentication in every reauthorization.

The second method is that the BS periodically reissue EAP-establish-key-Request to SS to begin reauthorization, the following steps are identical to the second phase of authorization. This method skips a full authentication in every reauthorization .

It is better to adopt the second method.

[add the following as show]

7.2.1.3.2 Authorization via PKM Extensible Authentication Protocol

The first steps of the authorization flow are as follows:

~~5)~~ 1)Upon successful completion of ranging (and capabilities exchange), a logical signal (ie. “link activation”) is sent upwards on the Logical Control Interface at the BS (ie. the EAP authenticator).This will cause the authenticator to begin the authentication sequence.

~~6)~~ 2)EAP on the Authenticator sends an EAP-Request message to the supplicant. This Request might be an EAP identity request or the beginning of an EAP method. The message is encapsulated in a MAC management PDU and transmitted.

~~7)~~ 3)EAP on the supplicant receives EAP-Request, passes it to the local EAP method for processing , and transmits EAP Response . Steps 2 and 3 (EAP-Request/Response exchange) continue as many times as needed.

Steps 2 and 3 (EAP-Request/Response exchange) continue as many times as needed.

After one or more EAP-Request/Response exchanges, the authentication server (whether local to the Authenticator or connected remotely via an AAA protocol) determines whether or not the authentication is successful.

The next steps of the authorization flow are as follows:

~~8)~~ 4)Upon success, EAP on the authenticator transmits a “success” signal on the logical control interface to fully activate the airlink.

~~9)~~ 5)EAP on the authenticator transmits EAP-success, which is then encapsulated in a MAC management message and transmitted to the supplicant.

~~10)~~ 6)EAP on the supplicant transmits a “success” indication on the logical control interface to fully activate the airlink.

~~11)~~ 7)Both EAPs (authenticator and supplicant) export the AAA-key across the logical control interface. As detailed in [3], the AAA-key is the shared “master key” that is derived by the two sides in the course of executing the EAP inner method The authentication part of the authorization flow (and the involvement of the generic EAP layer) is now complete.

The authentication part of the authorization flow (and the involvement of the generic EAP layer) is now complete.

The final steps of the authorization flow:

~~1)~~8)The BS and MSS each derive the EAP Master Key from the AAA-Key. The EAP Master Key is derived simply the taking the 32 lowest order octets of the AAA-Key.

~~2)~~9) BS sends the EAP-Establish-Key-Request PKM message (including a 32-byte nonce) to the MSS. The MSS then generates its own 32-byte nonce, and derives a Transient Key (TK) as follows:

TK = PRF-384(*EAP Master Key*, “Pairwise key expansion”,

Min(*BSId, SSId*) |

Max(*BSId, SSId*) |

Min(*BS-Generated-Nonce, MSS-Generated-Nonce*) |

Max(BS-Generated-Nonce, MSS-Generated-Nonce))

Where

```
PRF-384 (K, A, B) :=
    for i = 0 to 3 do
        R = R | HMAC-SHA-1(K, A | 0 | B | I i )
    return LeastSignificant-384-bits(R).
```

and “|” denotes bit string concatenation.

The MSS then derives Key Confirmation Key (KCK) and Authorization Key (AK) as follows:

KCK = bits 0-127 (ie. lowest order) of the TK (first 16 octets)

AK = bits 224-383 of the TK (last 20 octets)

The BS can attempt to use a cached or handover-transferred Master Key and avoid a full reauthentication .To do this, it sends EAP-Establish-Key-Request specifying the MKID attribute, which identifies by name the Master Key that the MSS should use for AK establishment if it also has the MK cached.

~~3)~~ 10)SS sends the EAP-Establish-Key-Reply PKM message (including the 32-byte nonce that it used to derive TK) to the BS. EAP-Establish-Key-Reply includes an HMAC Tuple TLV, which must be calculated using the KCK derived above.

Upon receipt of the EAP-Establish-Key-Reply, the BS computes the TK, KCK, and AK as above. BS then validates the HMAC Tuple. If the HMAC tuple is incorrect, BS discards the message without responding.

If the MSS elects not to proceed with key establishment (eg. the EAP-Establish-key-request specified an unknown MKID), the MSS sends EAP-Establish-Key-Reject instead.

~~4)~~ 11)BS sends the EAP-Establish-Key-Confirm PKM message to supply the MSS with its SA information and activate the AK.

A BS shall periodically refresh AK for each SS by reissuing EAP-establish-key-Request to SS. Reauthorization begin from the step 9. Reauthorization needn’t to execute a full authentication , it use the EAP master key derived in authorization to derive TK. Unlike reauthorization via PKM RSA authentication protocol, reauthorization via PKM extensible authentication protocol is managed by BS.