Project	**IEEE 802.16 Broadband Wireless Access Working Group** <http://ieee802.org/16>
Title	**Secure Extended MAC Header Type II**
Date Submitted	**2007-~~03-28~~05-04**
Source(s)	Shashikant Maheshwari, Yousuf Saifullah, Yogesh Swami, Haihong Zheng, Adrian Boariu Nokia Siemens Networks 6000 Connection Drive, Irving, TX voice: 972 839 1878 mailto: shashikant.maheshwari@nsn.com
Re:	IEEE 802.16j-07/013:“Call for Technical comments and contributions regarding IEEE Project P802.16j”
Abstract	This document presents a mechanism for securing Extended MAC Header Type II.
Purpose	Propose an efficient signaling acknowledgment operations for IEEE 802.16j
Notice	
Release	
Patent Policy and Procedures	

Secure Extended MAC Header Type II

Introduction

The Extended MAC header type II was accepted in the 16j baseline document [1]. The MAC management messages are sent without encryption between MR-BS and RS. A rogue RS could read these messages and could send a false response. For example, it could send NAK instead of ACK (using Extended MAC Header Type II), and distort the information relevant to the procedure in MR-BS. As the MAC headers are not authenticated, a malicious user could also send a wrong BR header using Extended MAC header type II, causing deprivation of bandwidth in the system for the legitimate users. This contribution proposes a simple secured Extended MAC header type II for relay links.

This contribution proposes a lightweight mechanism for message authentication, which does not require additional bytes. Please note that HMAC/CMAC tuple has lot of overhead. HMAC tuple is 21 bytes, and CMAC tuple is 13-19 bytes. It seems inappropriate to send so many bytes in the HMAC/CMAC tuple for protecting 6 bytes of header.

As A-HCS field is only 8 bits long, the probability of breaking into this field is 1 in 255, which is not comparable to HMAC/CMAC mechanism. Inspite of this shortcoming (which is mainly by the CRC field size, not due to our algorithm) we believe that the scheme is useful in preventing malicious uplink bandwidth wastage.

A MAC header has 8-bit mandatory Authenticated Header Check Sum (A-HCS) field. Currently, the checksum is computed as the residue of the generator polynomial (D8+D2+D+1). We propose that instead of computing the checksum as the standard residue, we compute the checksum using the message authentication code as mentioned below:

A-HCS = CMAC(CMAC_KEY_U⊗counter, 5-byte-checksum) mod (D8+D2+D+1)

A-HCS = HMAC(HMAC_KEY_U⊗counter, 5-byte-checksum) mod (D8+D2+D+1)

The operations in the above equations are described in the spec changes sections. In this way, the ACK header has error and integrity protection at the same time. The purpose of the counter is to ensure that even though the A-HCS is only 8 bits long, it's harder for the attacker to find a HCS collision and replay the message for bandwidth request.

Specific Text change

[Insert the following text after the first para in 6.3.2.1.2.2.2:]

This type of MAC header has 8-bit mandatory Authenticated Header Check Sum (A-HCS) field for providing error and integrity protection. A-HCS is computed using the message authentication code as mentioned below:

A-HCS = CMAC(CMAC_KEY_U⊗counter, 5-byte-checksum) mod (D^8+D^2+D+1)

A-HCS = HMAC(HMAC_KEY_U⊗counter, 5-byte-checksum) mod (D^8+D^2+D+1)

Where CMAC/HMAC_KEY_U is the CMAC/HMAC key that the RS has generated during key exchange and counter is a monotonically increasing number, which is of the same bit-length of CMAC/HMAC_KEY_U and ⊗ indicates the XOR operation.

[Change only the HCS field in Figure XX to A-HCS:]

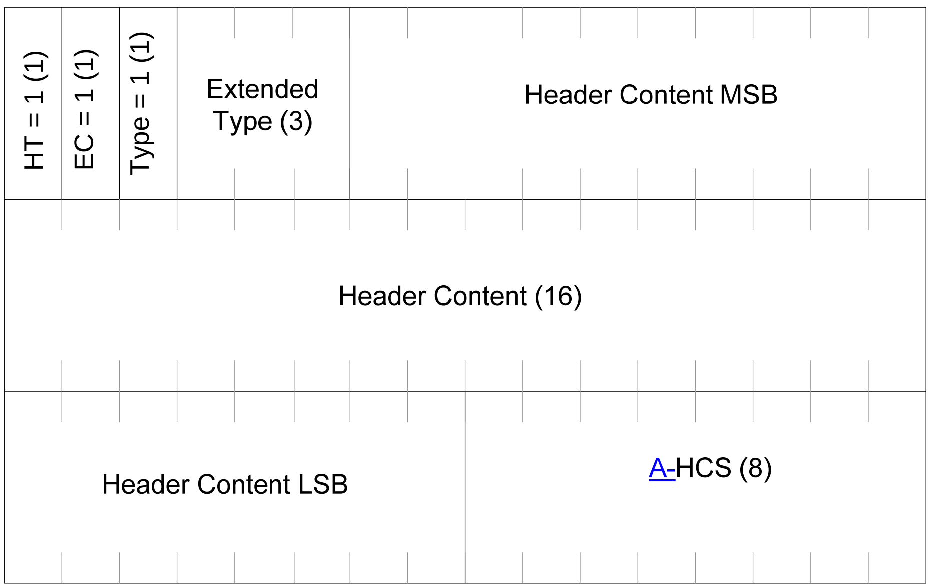

Figure XX Extended MAC Signaling Header Type II Format

References

[1] IEEE802.16j-06/026r3 Baseline Document for Draft Standard for 16j
[2] IEEE C802.16j_07/028r3 Message definition to support MS network entry in centralized allocation model