고속 무선 모뎀에서의 CCM 보안 모드 구현에 관한 연구

이현석*, 이장연*, 조진웅*
*전자부품연구원 통신네트워크연구센터
e-mail:hslee75@keti.re.kr, {jylee136, chojw}@keti.re.kr

Research of the CCM security mode in a high-speed wireless modem

Hyeon-Seok Lee*, Jang-Yeon Lee*, Jin-Woong Cho*
*Wireless Network Research Center, Korea Electronic Technology Institute

요 약

최근 UWB, IEEE802.11n과 같은 고속 무선 통신에서는 고속의 암호/복호 처리가 요구되고 있다. 본 논문은 UWB, Zigbee, IEEE802.11과 같은 최신 무선 통신 기술에서 보안 기능의 근간이 되는 CCM(CTR+CBC-MAC) 보안 모드 구현에 관한 것이다. AES와 같은 블록암호알고리즘과 결합된 CCM 기능을 하드웨어로 구현하는 방법을 제시한다. 특히, MAC, DMA모듈과 Hard-wired된 형태로 구현하여 통신속도 저하없이 무선 데이터 송/수신과 동시에 실시간으로 암호/복호 연산을 수행할 수 있으며, CCM 구동 clock을 최소화하여 고속 동작과 저전력 설계의 목적을 달성할 수 있다.

1. 서론

최근 근거리 고속 무선 통신을 목표로 하여 UWB, IEEE802.11n등이 개발되고 있다. UWB는 STB와 평판 DTV같은 영상기기에서 HD 영상 신호를 전송하는 무선 AV 관련 시장과 휴대성, 이동성에 중심을 둔 PC 주변기기 관련 시장을 목표로 하고 있다. 또한, IEEE802.11n 기술은 무선랜에서 가장 단점으로 지적되어 온 통신 속도에 대한 한계를 극복하기 위해 최대 500Mbps의 속도를 목표로 하며, 무선랜 관련 선진업체를 중심으로 칩셋을 출시하고 있다. 이 기술들은 공통적으로 물리계층에서의 고속화가 주 이슈가 되고 있지만, 이에 못지 않게 무선통신의 취약 부분인 보안이 중요한 이슈로 남아 있다. 무선 통신의 취약 부분 중 하나인 보안을 강화하기 위하여 UWB, IEEE802.11n에서는 블록암호 알고리즘인 AES(Advanced Encryption Standard)[1]를 사용한 CCM(CTR+CBC-MAC) 보안 모드를 사용한다.

2. CCM 구성

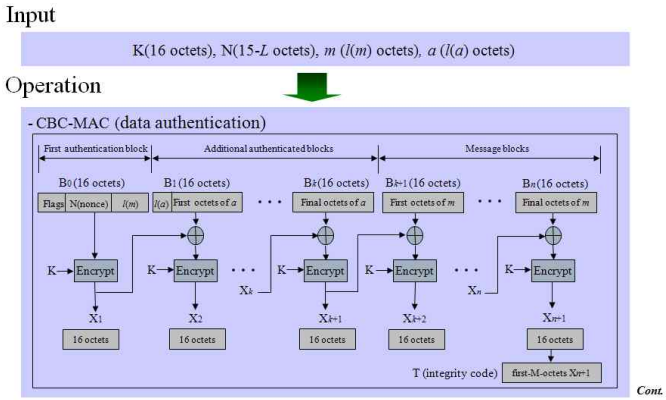
2.1 CCM 개요

CCM(Counter with CBC-MAC)은 RFC 3610에 정의되어 있다.[2] CCM은 데이터 암호화를 위해서 카운터 모드를 사용하고, 메시지 무결성을 검사하기 위해서 CBC-MAC모드를 사용한다. 암호화는 보안 페이로드의 전부 또는 일부에 적용되며, 무결성 코드는 MAC헤더와 보안 페이로드 전체에 대해서 생성한다.

CCM은 M(무결성 코드의 바이트 수)과 L(길이 필드를 나타내는 바이트 수)의 두 가지 입력 파라미터를 가진다. CCM은 보안 프레임을 보호하기 위해 갱신될 수 있는 키와 매 프레임마다 다른 값을 가지는 Nonce 사용을 요구한다. CCM nonce는 동일한 키를 사용하여 같은 값을 사용하면 암호학적으로 결점이 생기므로 같은 값을 사용하면 안 된다. 그래서 키의 갱신 주기는 CCM nonce가 반복되지 않도록 결정되어야 한다. CCM nonce는 일반적으로 13 옥텟으로 구성된다. Nonce는 인증 블록 B_0의 구성요소가 되어, CBC-MAC의 입력값이 된다. 또한, A_i의 구성요소가 되어 카운터 모드의 입력값이 된다.

2.1 CBC-MAC

CCM 무결성 생성 코드는 CBC-MAC 모드를 사용한다. 또한 CCM 생성 블록의 구조는 그림1과 같다.

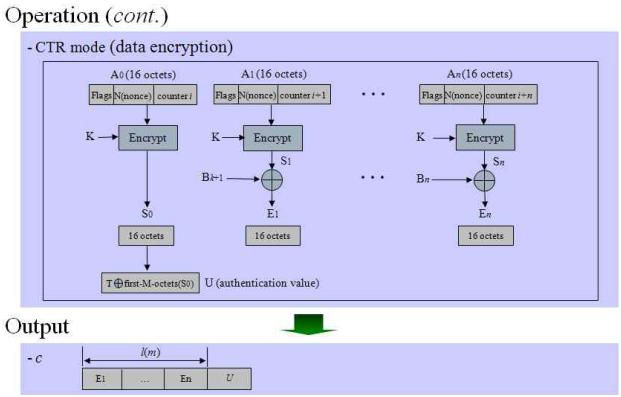


[그림 1] CBC-MAC 동작 구조

$X_1 := E(K, B_0)$
$X_{i+1} := E(K, X_i \oplus B_i)$ for $i = 1, ..., n$
$T := first-M-octets(X_{n+1})$

2.2 CTR

데이터 암호화는 카운터 모드를 사용한다. 카운터 모드의 구조는 다음 그림 2와 같다.



[그림 2] CTR 동작 구조

카운터는 보안 프레임 전송을 시작할 때 "0"으로 초기화되어, 매 암호화 블록마다 "1"씩 증가한다.

$S_i := E(K, A_i)$ for $i = 0, 1, 2, ...$

무결성 코드 U는 S_0를 이용하여 계산된다.

$U := T \oplus first-M-octets(S_0)$

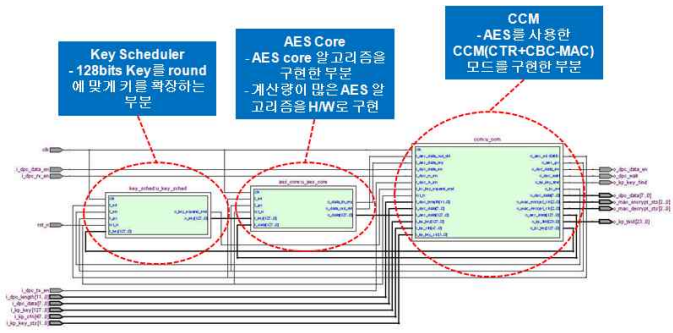
2.3 복호화

메시지를 복호화하기 위해서는 대칭키, Nonce, 암호화되지 않은 보안 메시지, 암호화된 보안 메시지, 무결성 코드가 필요하다.

복호화는 암호화된 메시지로부터 원래의 메시지를 복원한 다음 무결성 코드를 계산하여 수신된 무결성 코드와 비교한다.

3. CCM 구현

고속의 CCM 암/복호화 기능을 처리하기 위하여 Hardware기반의 CCM 처리 블록을 설계하였으며 블록 암호 알고리즘은 128bits AES를 사용하였다.



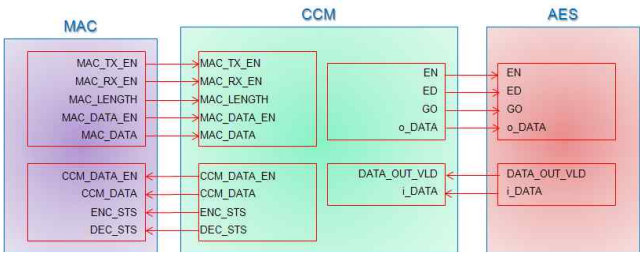
[그림 3] 보안 블록의 하드웨어 구조

가. Key Scheduler
- 128bits Key를 입력으로 하여 44 클럭후에 11개의 Encryption Key 생성 부분
- Key Scheduler를 H/W로 구현하여 별도의 소프트웨어의 작업 해소

나. AES Core
- 128bits AES 블록 암호 알고리즘을 구현한 부분
- 128bits 입력 데이터를 암호화하는데, 12클럭 소요 (Round당 1클럭, 총 11 Round에 11클럭 + 제어용 1클럭 = 12클럭)

다. CCM
- CCM(CTR+CBC-MAC) 모드를 구현한 부분
- 암/복호화 과정이 실시간으로 처리될 수 있도록 하였으며, 예외 사항으로 Key 불일치 에러나 데이터 무결성 에러등을 처리함

다음 그림 4는 MAC과 보안 관련 블록과의 인터페이스를 도시한 그림이다.



[그림 4] MAC, CCM, AES의 인터페이스 구조

송신시 MAC은 MAC헤더와 MAC페이로드를 포함한 송신 데이터를 CCM 블록에게 전송을 하고, 이에

따라 CCM 블록은 해당 키와 Nonce를 생성하여 위에서 설명한 Encryption과 무결성 코드를 생성하며, 수행 결과를 ENC_STS신호를 통해서 MAC에게 알려준다.

수신시에는 MAC이 PHY로부터 수신한 MAC헤더와 MAC페이로드를 CCM블록에게 전송하면, CCM 블록은 송신시에 사용한 키와 Nonce를 추출하여 Decryption과정을 수행한 후 그 결과를 DEC_STS 신호를 통해 MAC에게 알려준다. 이 모든 과정은 송/수신과 동시에 실시간으로 이루어지므로 송/수신 암호화 관련 F/W의 제어를 최소화할 수 있다.

[표 1] CCM 연산 파라미터

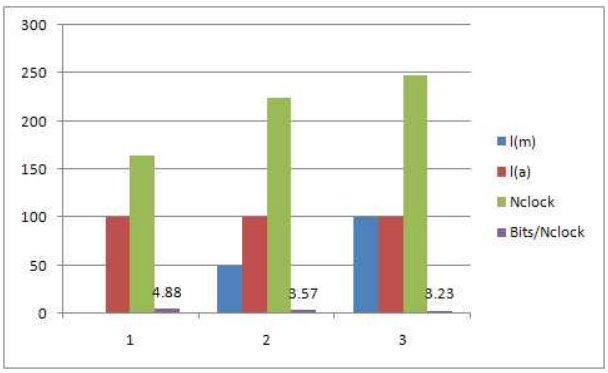
구분	기호	값	비고
암호화 단위 블록의 길이	N_{block}	16	단위 : octet (128 bits)
암호화 할(된) 메시지의 길이	$l(m)$	가변	단위 : octet
무결성 검사할 메시지의 길이	$l(a)$	가변	단위 : octet
Key Scheduling	T_{KS}	44	단위 : clock
CBC-MAC(B_0)	T_{B_0}	12	단위 : clock
CBC-MAC(B_1)	T_{B_1}	12	단위 : clock
CBC-MAC(B_i)	T_{B_i}	12	단위 : clock
CTR(A_0)	T_{A_0}	12	단위 : clock
CTR(A_i)	T_{A_i}	12	단위 : clock

$$Number\,of\,CTR(A_i): N_{ctr(A_i)} = ceiling(l(m), N_{block}) / N_{block}$$

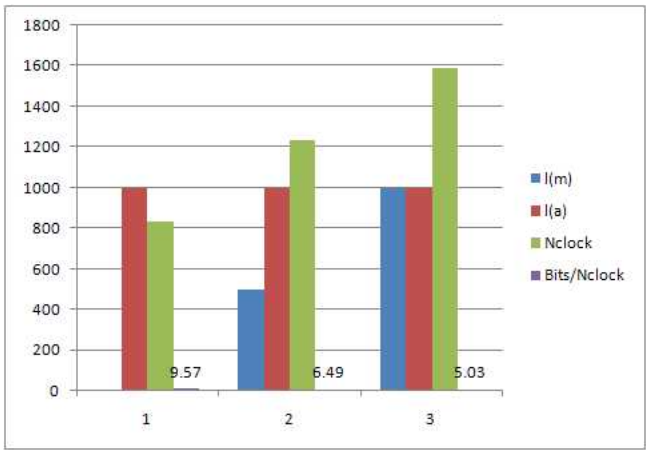
$$Number\,of\,CBC-MAC(B_i): N_{cbc-mac(B_i)} = N_{ctr(A_i)} + ceiling(l(m) - l(a), N_{block}) / N_{block}$$

$$Total\,Number\,of\,Clock = T_{KS} + T_{B_0} + T_{B_1} + T_{B_i} \times N_{cbc-mac(B_i)} + T_{A_0} + T_{A_i} \times N_{cbc-mac(B_i)}$$

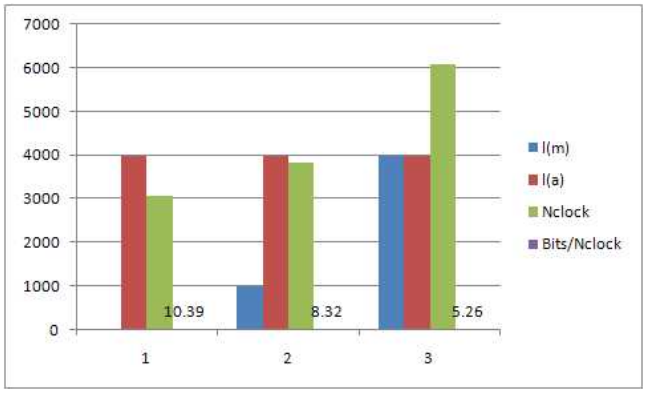
위의 수식을 바탕으로 $l(m)$과 $l(a)$를 변경하면서 CCM 연산을 수행하기 위해 필요한 클럭수와 수행연산비트 대 클럭 비율(bits/clock)을 계산하였다.



[그림 5] 패킷 길이가 짧은(100바이트) 비콘이나 제어 프레임인 경우



[그림 6] 패킷 길이가 1,000바이트인 경우



[그림 7] 패킷 길이가 4,000바이트인 경우

위의 그림 5,6,7에서 보듯이 $l(m)$, $l(a)$의 길이에 따라 Bits/Nclock값이 3.23~10.39의 값을 나타낸다. $l(m)$과 $l(a)$의 값이 동일한 경우, 즉 모든 패킷에 대해 암호화와 무결성 코드를 생성하는 경우 수행연산비트 대 클럭 비율(bits/clock)값은 3.23~5.26의 값을 나타낸다.

만약 전송속도가 500Mbps이라고 가정하면, 하드웨어 clock 동작속도는 그림 5의 경우에는 $(500Mbits/\sec)/(3.23bits/clk) \cong 155MHz$보다 커야 한다.

그림 6의 경우에는 $(500Mbits/\sec)/(5.03bits/clk) \cong 100MHz$보다 커야 한다. 마지막으로 그림 7의 경우에는 $(500Mbits/\sec)/(5.26bits/clk) \cong 96MHz$보다 커야 한다.

그러나, 패킷 길이가 짧은 비콘이나 제어 프레임은 일반적으로 낮은 전송속도로 전송하거나, 모든 프레임에 대해 암/복호화와 무결성 코드를 생성하는 경우가 아니므로, 구현상의 여유값을 고려하여 수행연산비트 대 클럭 비율(bits/clock)을 6으로 예상하였다.

4. 결론

본 논문에서는 고속의 통신 속도를 지원하는 CCM 설계 방법에 대한 고찰을 하였으며, RTL 시뮬레이션과 수식적 계산을 통해 그 성능을 검증하였다. 특히, 본 논문에서 제안한 CCM 설계 부분은 앞으로의 무선 통신 칩 설계에 적용 가능하며, MAC 하드웨어의 동작 Clock을 낮추어 저전력 설계에 중요한 역할을 할 것으로 기대한다.

참고문헌

[1] "Specification for the Advaned Encryption Standard (AES)" FIPS Publication 197, Nov. 2001.

[2] D.Whiting, R.Housley, "RFC 3610 on Counter with CBC-MAC(CCM)", IETF, 9월, 2003.