논문 2007-44SD-7-10

AES 기반 와이브로 보안 프로세서 설계

(A Design of AES-based WiBro Security Processor)

김 종 환*, 신 경 욱**

(Jong-Hwan Kim and Kyung-Wook Shin)

요 약

본 논문에서는 와이브로 (WiBro) 무선 인터넷 시스템의 보안 부계층 (Security Sub-layer)을 지원하는 와이브로 보안 프로세서 (WBSec)의 효율적인 하드웨어 설계에 관해 기술한다. 설계된 WBSec 프로세서는 AES (Advanced Encryption Standard) 블록암호 알고리듬을 기반으로 하여 데이터 암호·복호, 인증·무결성, 키 암호·복호 등 무선 네트워크의 보안기능을 처리한다. WBSec 프로세서는 ECB, CTR, CBC, CCM 및 key wrap/unwrap 동작모드를 가지며, 암호 연산만을 처리하는 AES 코어와 암호·복호 연산을 처리하는 AES 코어를 병렬로 사용하여 전체적인 성능이 최적화되도록 설계되었다. 효율적인 하드웨어 구현을 위해 AES 코어 내부의 라운드 변환 블록에 하드웨어 공유기법을 적용하여 설계하였으며, 또한 하드웨어 복잡도에 가장 큰 영향을 미치는 S-box를 체 (field) 변환 방법을 적용하여 구현함으로써 LUT (Look-Up Table)로 구현하는 방식에 비해 약 25%의 게이트를 감소시켰다. Verilog-HDL로 설계된 WBSec 프로세서는 22,350 게이트로 구현되었으며, key wrap 모드에서 최소 16-Mbps의 성능과 CCM 암호·복호 모드에서 최대 213-Mbps의 성능을 가져 와이브로 시스템 보안용 하드웨어 설계에 IP 형태로 사용될 수 있다.

Abstract

This paper describes an efficient hardware design of WiBro security processor (WBSec) supporting for the security sub-layer of WiBro wireless internet system. The WBSec processor, which is based on AES (Advanced Encryption Standard) block cipher algorithm, performs data encryption/decryption, authentication/integrity, and key encryption/decryption for packet data protection of wireless network. It carries out the modes of ECB, CTR, CBC, CCM and key wrap/unwrap with two AES cores working in parallel. In order to achieve an area-efficient implementation, two design techniques are considered; First, round transformation block within AES core is designed using a shared structure for encryption/decryption. Secondly, SubByte/InvSubByte blocks that require the largest hardware in AES core are implemented using field transformation technique. It results that the gate count of WBSec is reduced by about 25% compared with conventional LUT (Look-Up Table)-based design. The WBSec processor designed in Verilog-HDL has about 22,350 gates, and the estimated throughput is about 16-Mbps at key wrap mode and maximum 213-Mbps at CCM mode, thus it can be used for hardware design of WiBro security system.

Keywords: WiBro, Security Processor, AES, Modes of Operation, Authentication, key wrap/unwrap

I. 서 론

2.3GHz 주파수 대역을 사용하여 시속 60km/h 이상의 이동 중에도 인터넷에 접속할 수 있는 와이브로 시스템은 광대역 무선통신 국제표준인 IEEE 802.16e (WiMAX)에 반영되는 등 한국이 국제 표준화를 주도하고 있는 차세대 이동통신 기술이다. 유선 환경과는 달리 브로드캐스팅 네트워크인 와이브로는 기지국 영역 내에 있는 모든 단말기들이 다른 사람의 송수신 데이터 내용을 수신할 수 있으므로, 허가된 수신자 이외에 다른 사람이 메시지 내용을 보지 못하게 하는 데이터 기밀성과 사용자 인증 등 정보보안 기술이 필수적으로 요구된다.

와이브로 시스템의 보안 부계층 (security sub-layer)은 광대역 무선 네트워크에서의 보안과 인증, 그리고

* 정회원, 픽셀플러스(주)
(PixelPlus Inc.)
** 정회원, 금오공과대학교 전자공학부
(School of Electronic Eng., Kumoh National Institute of Technology)
접수일자: 2007년3월8일, 수정완료일: 2007년6월18일

기밀성을 제공하여 단말과 기지국간에 전달되는 패킷 데이터에 대한 암호화 기능을 통해 불법 사용자의 서비스 도난 공격에 대한 강인한 방어능력을 제공한다. 기지국에서는 네트워크 전반에 걸쳐 서비스 플로우에 대한 암호화를 수행하여 데이터 전송 서비스에 대해 권한 없이 접속하는 것을 방지한다.[1-2]

와이브로의 보안 부계층은 encapsulation 프로토콜과 키 관리 프로토콜로 구성된다. Encapsulation 프로토콜은 패킷 데이터 보안을 위한 데이터 암호화 및 인증 알고리즘 등 "cryptographic suites" 집합과 보안 알고리즘들을 적용시키는 방법을 정의한다. 키 관리 프로토콜은 기지국에서 단말로 키 관련 데이터를 안전하게 분배하는 방법을 제공한다. Cryptographic suites는 트래픽 암호키 (Traffic Encryption Key; TEK) 교환, 데이터 암호화 및 인증을 위한 알고리듬을 정의하는 SA (Security Association)의 집합이다. 데이터 암호·복호에 사용되는 알고리즘으로는 DES (Data Encryption Standard) 기반의 CBC (Cipher Block Chaining) 운영모드와 AES 기반의 CCM (Counter with CBC-MAC) 운영모드, CTR (Counter) 운영모드 및 CBC 운영모드 등이 정의되어 있다. TEK를 암호·복호하기 위한 알고리듬으로는 3중 DES, RSA, AES 기반의 ECB 운영모드와 key wrap 알고리듬이 사용된다.[1-3]

단말과 기지국 사이에서 수행되는 인증절차에서, 단말은 자신이 지원하는 모든 cryptographic suites 목록을 기지국에 알리게 되며, 기지국은 이 중에서 하나를 선택하여 TEK 암호화, 데이터 암호화 및 인증을 수행하게 된다. 따라서 기지국에서는 cryptographic suites에서 정의된 모든 알고리듬이 구현되어야 하며, 단말기에서는 TEK 및 데이터 암호화와 인증을 위한 알고리듬들 중 한 가지 이상이 구현되어야 한다. 와이브로의 안전한 서비스를 위해서는 보안모듈의 소프트웨어 또는 하드웨어 개발이 필수적이며, 시스템과 단말기에서의 성능과 저전력을 위해서는 보안모듈의 하드웨어 개발이 필요하다.

본 논문에는 ECB, CTR, CBC, CCM 그리고 key wrap/unwrap 등의 동작모드를 수행하여 와이브로 시스템의 보안 부계층을 지원하는 와이브로 보안 프로세서 (WBSec; WiBro Security)의 하드웨어 설계에 대해 기술하며, 모바일 기기에 적합하도록 저전력, 고성능을 갖는 최적화된 구조를 제안하였다. Ⅱ장에서는 와이브로 보안 부계층 알고리듬에 관해 기술하며, Ⅲ장에서는 WBSec 프로세서의 효율적인 하드웨어 설계에 관하여 기술한다. Ⅳ장에서는 WBSec 프로세서의 ASIC 구현과 설계된 WBSec 프로세서의 검증 및 성능평가에 대해 기술한다.

Ⅱ. 와이브로 보안 부계층 알고리듬

와이브로 보안 부계층을 구성하는 AES 기반 알고리듬은 아래의 표 1과 같으며[1], 패킷 데이터의 암호·복호를 위해서는 AES의 CTR, CBC, CCM 동작모드[3-7]가 사용되며, TEK의 암호·복호를 위해서는 AES의 ECB 동작모드와 key wrap/unwrap 알고리듬[8]이 사용된다.

CCM 동작모드는 미국 국가기술표준국 (NIST : National Institute of Standards and Technology)에서 제시한 AES의 동작모드 중 하나로서 CBC 모드와 CTR 모드를 결합하여 데이터의 무결성과 은닉성을 동시에 보장한다. AES의 CBC 동작모드를 이용하여 메시지의 무결성을 위한 MIC (Message Integrity Code)를 생성하고, CTR 모드를 이용하여 메시지의 은닉성을 위한 암호·복호 연산을 수행한다. CCM은 AES 암호 연산만으로 구현이 가능하다는 장점을 가지고 있다.

CBC 동작모드는 MAC PDU 페이로드의 암호·복호와 CCM 동작모드에서 MIC을 생성하기 위해 사용된다. CBC 동작모드는 Ⅳ (Initialization Vector)를 필요로 하며, 암호모드에서는 현재 블록의 암호화 결과가 다음 블록 암호화의 Ⅳ로 사용되며, 복호모드에서는 현재 암호문 블록이 다음 암호문 블록의 복호화에 Ⅳ로 사용되는 chain 형태를 갖는다. 암호·복호를 위한 CBC는 AES 암호·복호 연산이 모두 사용되며, MIC을 생성하는 CCM 동작모드의 CBC에는 AES 암호 연산만 사용된다.

CTR 동작모드는 패킷 데이터의 암호·복호와 CCM 동작모드의 암호·복호에 사용된다. CTR 동작모드의 암호화 과정은 계수기 값을 AES로 암호화한 후, 외부

표 1. AES 기반 와이브로 보안 부계층 알고리듬
Table 1. AES-based Wibro security sub-layer algorithms.

패킷 데이터 암호·복호	TEK 암호·복호
CTR	ECB
CBC	key wrap/unwrap
CCM	

에서 입력되는 평문과 XOR 연산을 거쳐 암호문이 출력된다. 복호화 과정은 암호화에서 사용된 계수기 값과 동일한 계수기 값을 AES로 암호화한 후, 입력되는 암호문과 XOR 연산을 거쳐 평문이 출력된다. CTR 동작 모드는 AES 암호 연산만을 사용하여 데이터 암호·복호 연산을 수행하므로 하드웨어로 구현 시 다른 동작모드에 비해 작은 면적으로 구현할 수 있다.

Key wrap/unwrap 알고리듬[8]은 데이터의 암호·복호에 사용되는 암호 키 (TEK)를 암호화하기 위한 알고리즘이며, 키를 암호화하는 키 싸기 (encapsulation)와 키를 복호화하는 키 풀기 (decapsulation), 그리고 데이터의 무결성을 검사하는 부분으로 구성된다.

Key wrap 모드는 데이터 암호·복호에 사용되는 TEK를 KEK (Key Encryption Key)로 암호화하는 키 싸기 모드이며, 그림 1과 같이 표현되는 pseudo 코드의 연산과정으로 처리된다. 연산과정은 *6n*번의 AES 연산으로 이루어지며, 반복 횟수는 $n=\lceil L/64 \rceil$로 (단, L은 KEK의 길이) 주어진다. TEK의 암호화를 위해 128비트의 KEK가 사용되므로 12번의 AES 연산이 수행된다. 그림 1의 pseudo 코드에서 평문 (plaintext)은 암호화될 TEK를 나타내며, 와이브로 보안에는 128비트의 키가 사용된다. TEK는 MSB 64비트와 LSB 64비트의 두 부분으로 나누어 처리되며, 첫 번째 AES 연산에는 64비트의 IV "a6a6a6a6_a6a6a6a6"가 함께 사용된다. 첫 번째 AES 연산은 64비트의 IV와 TEK의 LSB 64비트로 구성되는 128비트를 암호화하며, AES 암호연산 출력의 MSB 64비트는 계수기 출력 *t*와 XOR되어 다음 AES 입력의 MSB 64비트로 사용된다. 한편, 계수기 출력 t값은 초기값 0에서 라운드 연산이 반복될 때 마다 '1'씩 증가하는 값을 갖는다.

두 번째 AES 연산은 첫 번째 AES 결과의 MSB 64비트와 TEK의 MSB 64비트로 구성되는 128비트를 암호화하며, AES 출력 중 MSB 64비트는 계수기 출력 t와 XOR된 후, 다음 AES의 입력으로 사용된다. *(i)*-번째 AES 연산은 *(i-1)*-번째 AES 결과 중 MSB 64비트와 *(i-2)*-번째 AES 결과의 LSB 64비로 구성되는 128비트에 대해 암호화 연산이 수행된다. 총 12번의 AES 반복 연산과정이 끝나면 key wrap의 최종 결과로 192비트의 암호화된 키 값이 출력된다.

Key unwrap 모드는 key wrap 알고리듬에 의해 암호화된 TEK를 복호화하기 위한 모드이며, 그림 2와 같이 표현되는 pseudo 코드의 연산과정으로 처리된다. Key unwrap 연산과정은 key wrap 연산과정과 유사하

❑ Key Wrap
Inputs : Plaintext, n 64-bit values $\{P_1, P_2, \ldots, P_n\}$,
Key, K (the KEK)
Outputs : Ciphertext, (n+1) 64-bit values $\{C_0, C_1, C_2, \ldots, C_n\}$
1) Initialize variables
Set $A^0 = IV$, an initial value
For i = 1,…,n
$R^0_i = P_i$
2) Calculate intermediate values
For t = 1,…,s , where s = 6n
$A^t = MSB_{64}(AES_k(A^{t-1} | R^{t-1}_1)) \oplus t$
For i = 1,…,n-1
$R^t_i = R^{t-1}_{i+1}$
$R^t_n = LSB_{64}(AES_k(A^{t-1} | R^{t-1}_1))$
3) Output the results
Set $C_0 = A^t$
For I = 1,…,n
$C_i = R^t_i$

그림 1. AES 기반 key wrap 알고리듬의 pseudo 코드
Fig. 1. Pseudo code of AES-based key wrap algorithm.

❑ Key Unwrap
Inputs : Ciphertext, (n + 1) 64-bit values $\{C_1, C_2, \ldots, C_n\}$,
Key, K (the KEK)
Outputs : Plaintext, n 64-bit values $\{P_1, P_2, \ldots, P_n\}$
1) Initialize variables
Set $A^s = C_0$, where s = 6n
For i = 1,…,n
$R^s_i = C_i$
2) Calculate intermediate values
For t = 1,…,s
$A^{s-t} = MSB_{64}(AES^{-1}_k((A^s \oplus t) | R^s_n))$
$R^{s-t}_1 = LSB_{64}(AES^{-1}_k((A^s \oplus t) | R^s_n))$
For i = 2,…,n
$R^{s-t}_i = R^s_{i-1}$
3) Output the results
If A^0 is an appropriate initial value
Then
For i = 1,…,n
$P_i = R^0_i$
Else
Return an error

그림 2. AES 기반 key unwrap 알고리듬의 pseudo 코드
Fig. 2. Pseudo code of AES-based key unwrap algorithm.

며, AES 복호화 연산이 사용된다. 계수기 출력 *t*값은 wrap 모드에서는 AES 연산 후 더해졌으나 unwrap 모드에서는 AES 연산 전에 더해지며, 연산을 반복할 때 마다 초기값 12에서 1씩 감소하게 된다. AES 복호화에 사용되는 키 값은 암호화와 동일한 KEK 값을 사용한다. Unwrap 연산의 결과로 출력되는 192비트 중, MSB 64비트는 key wrap 모드에서 사용한 IV가 복호된 "a6a6a6a6_a6a6a6a6" 값이며, 나머지 128비트는 복호된 TEK 값이다.

AES 기반 key wrap/unwrap 알고리듬은 데이터의 무결성 (integrity)을 검사하기 위한 메커니즘을 포함하고 있다. Key wrap 모드에서 사용된 IV "a6a6a6a6_a6a6a6a6"값과 TEK의 LSB 64비트를 결합한 128비트를 AES로 암호화함으로써 암호화된 TEK 값에 IV가 숨겨지게 된다. 암호화된 TEK 값을 unwrap 모드로 복

호화하면, Ⅳ 값이 복호되어 출력된다. 따라서 key wrap 모드에서 사용된 Ⅳ와 unwrap 모드에서 복호화된 Ⅳ를 비교하여 데이터의 무결성을 검증할 수 있다.

Ⅲ. AES 기반 WBSec 프로세서 설계

1. WBSec 프로세서의 구조

본 논문의 WBSec 프로세서는 ECB 모드, CTR 모드, CBC 모드, CCM 모드 그리고 key wrap/unwrap 모드 등 5가지의 동작모드를 선택적으로 수행하도록 그림 3과 같은 구조로 설계되었다. CCM 모드에서 CTR 운영모드와 CBC 운영모드를 병렬처리 함으로써 WBSec 프로세서의 동작성능이 최적화되도록 두개의 AES 코어(AES-Enc, AES-Enc/Dec)를 사용하여 설계하였다. 암호 연산만을 처리하는 AES-Enc 블록은 CTR 모드와 CCM 모드의 CTR 운영모드에서 계수기 값을 암호화할 때 사용되며, 암호연산과 복호연산을 선택적으로 수행하는 AES-Enc/Dec 블록은 CTR 모드를 제외한 나머지 동작모드 (ECB, CBC, CCM의 CBC 운영모드, key wrap/unwrap)의 암호·복호연산을 처리한다.

키 스케줄러는 두 개의 AES 블록에서 사용되는 라운드 키를 생성하는 블록이며, 하나의 키 스케줄러만 사용하여 AES-Enc 블록과 AES-Enc/Dec 블록의 라운드 키가 생성되도록 설계함으로써 하드웨어가 최소화되도록 하였다. Reg-A와 Reg-B 블록은 입력 데이터나 각 동작모드의 중간 결과값을 저장하기 위한 레지스터 블록이다. op_mode 신호 값에 따라 5가지 동작모드 중 하나가 선택적으로 수행되며, ed_mode 신호를 이용하여 암호·복호모드가 선택적으로 수행된다.

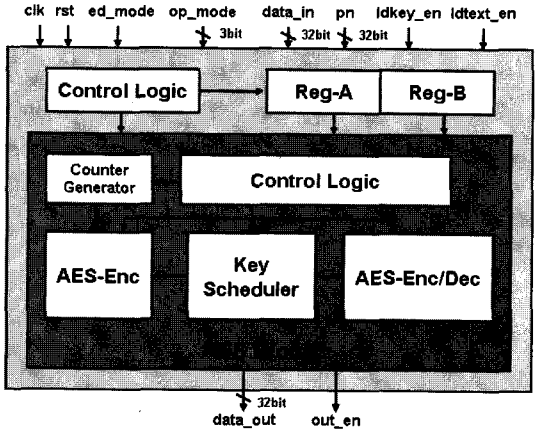


그림 3. WBSec 프로세서의 전체 블록도
Fig. 3. Block diagram of the WBSec processor.

2. WBSec의 상세 구조 및 동작모드 구현

WBSec 프로세서의 상세 구조는 그림 4와 같이 설계되었으며, 두 개의 AES 코어 (AES-Enc, AES-Enc/Dec)와 라운드 키 생성기 (key scheduler), 레지스터 블록 (Reg-A, Reg-B), XOR 및 MUX 블록 등으로 구성되어 5가지 동작모드가 선택적으로 수행된다. WBSec는 내부의 데이터 패스와 입·출력 포트가 32비트로 설계되었으며, 따라서 128비트의 데이터 (키, 평문, 암호문)는 32비트씩 4클록 주기 동안 입력·출력된다. 본 절에서는 WBSec 프로세서의 5가지 동작모드에 대해 설명한다.

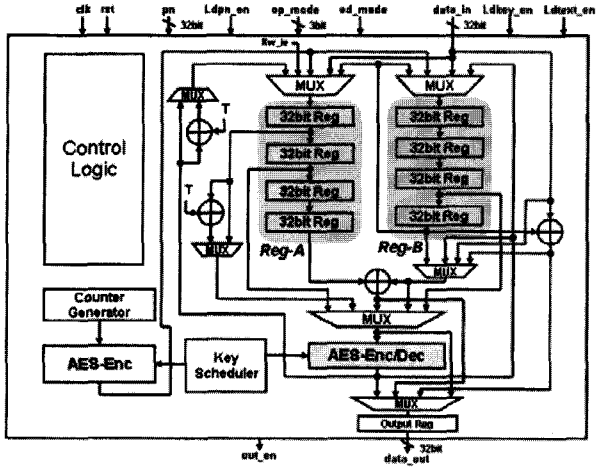


그림 4. WBSec 프로세서의 상세 구조
Fig. 4. Internal block diagram of the WBSec processor.

가. ECB 동작모드

ECB 동작모드는 가장 단순한 암호·복호 연산이며, 평문(암호문)이 Reg-B를 거쳐 AES-Enc/Dec 블록으로 입력되면 ed_mode 신호에 따라 암호(복호) 연산이 수행된 후 암호문(평문)이 출력되는 동작모드이다. 암호연산은 128비트의 키와 평문이 입력된 후, AES-Enc/Dec 블록에서 10번의 라운드 연산을 수행한 후 128비트의 암호문을 32비트씩 4클록 주기 동안 출력한다. 복호연산은 128비트 키를 입력받아 AES-Enc/Dec 블록의 암호모드 연산을 통해 10번째 라운드 키를 생성하고(이 연산은 새로운 키가 입력되었을 때만 수행됨), 생성된 라운드 키를 복호모드의 초기 키로 사용하여 10번의 복호모드 연산을 통해 128비트 평문을 생성한다. ECB 동작모드의 암호연산은 평문 입력으로부터 65 클록주기의 latency를 가지며, 복호연산은 복호 키 입력으로부터 125 클록주기의 latency를 갖는다.

나. CTR 동작모드

CTR 동작모드에서는 pn 포트로 계수기 초기화 값이 입력되며, 계수기에서는 데이터 블록이 입력될 때 마다 계수기 값을 1씩 증가시킨다. CTR 동작모드는 암호연산과 복호연산이 AES-Enc 블록을 통해 동일한 과정으로 수행된다. 평문(암호문)이 Reg-B에 저장된 후, 계수기에서 생성된 값이 AES-Enc 블록에서 암호화되어 Reg-A에 저장되며, 두 레지스터에 저장된 데이터가 XOR 연산된 후 암호문(평문)이 출력된다. CTR 동작모드의 암호·복호 연산은 평문 입력으로부터 70 클록주기의 latency를 갖는다.

다. CBC 동작모드

CBC 암호화 동작은 초기 Ⅳ가 입력되면 AES-Enc/Dec에서 암호연산 된 후 Reg-A에 저장되며, 첫 번째 평문이 입력되면 Reg-A에 저장된 데이터와 XOR 연산이 수행된 후 AES-Enc/Dec에서 암호화되어 암호문이 출력된다. 이와 동시에 암호문은 다음 평문 암호화의 Ⅳ로 사용되기 위해 Reg-A에 저장된다. 두 번째 평문이 입력되면 Reg-A에 저장된 데이터와 XOR 연산이 수행된 후 AES-Enc/Dec에서 암호화되어 암호문이 출력되고, 이와 동시에 암호문은 다음 평문 암호화의 Ⅳ로 사용되기 위해 Reg-A에 저장된다. 복호화 모드의 동작은 초기 Ⅳ가 입력되면 AES-Enc/Dec에서 암호화되어 Reg-A에 저장되며, 첫 번째 암호문이 입력되면 암호문을 AES-Enc/Dec에서 복호연산한 후 Reg-A에 저장된 데이터와 XOR 연산을 수행하여 평문을 출력한다. 두번째 암호문 블록이 입력되면 Reg-B에 저장함과 동시에 Reg-B에 저장되어 있던 이전 암호문 블록이 Reg-A로 옮겨진다. 이는 현재 암호문 블록의 복호를 위해 이전 암호문 블록이 Ⅳ로 사용되므로, 현재 암호문 블록의 복호연산이 종료될 때 까지 이를 저장하고 있기 위함이다. CBC 동작모드의 암호·복호 연산은 Ⅳ 입력으로부터 128 클록주기의 latency를 갖는다.

라. CCM 동작모드

CCM 암호모드의 동작은 Ⅳ를 AES-Enc/Dec로 암호화하여 Reg-A에 저장하며, 계수기 값을 AES-Enc로 암호화하여 Reg-B에 저장한다. 첫 번째 평문이 입력되면 Reg-B에 저장된 암호화된 계수기 값과 평문을 XOR 연산하여 암호문을 출력하고, 계수기 값을 1증가시켜 AES-Enc로 암호화하여 Reg-B에 저장한다. 이와 동시에 CBC 운영모드에서는 Reg-A에 저장된 암호화

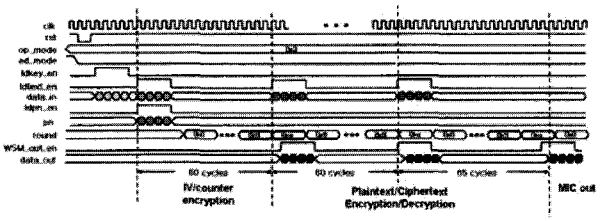


그림 5. CCM 동작모드의 타이밍도
Fig. 5. Timing diagram of CCM mode.

된 Ⅳ와 평문을 XOR 연산한 후 AES-Enc/Dec로 암호화하여 Reg-A에 저장한다. 이 과정이 평문 블록에 대해 반복되어 CTR 운영모드를 통해 암호문이 출력되며, CBC 운영모드에서는 평문 블록에 대한 암호화 연산이 chain 형태로 계속되어 마지막 평문 블록에 대한 처리가 끝나면 MIC 값을 출력한다. 이 MIC 값은 다시 CTR 운영모드를 통해 암호화되어 전송된다. CCM 복호모드는 암호모드와 동작이 유사하며, CBC의 입력 부분만 다르게 동작한다. CCM 복호모드에서의 CBC 운영모드는 CTR 모드에서 복호된 평문을 이용하여 MIC 값을 계산하게 된다. 그림 5는 CCM 모드의 동작 타이밍도이다. 키 입력 후 계수기 값과 IV가 동시에 입력되고, 이들이 각각 AES-Enc와 AES-Enc/Dec에서 암호화되기 위해 60 클록주기가 소요된다. 그 후, 각 평문 블록 처리에 60 클록주기씩이 소요되며, 마지막 평문 블록은 65 클록주기에 처리된 후 MIC 값이 출력된다.

마. Key wrap/unwrap 동작모드

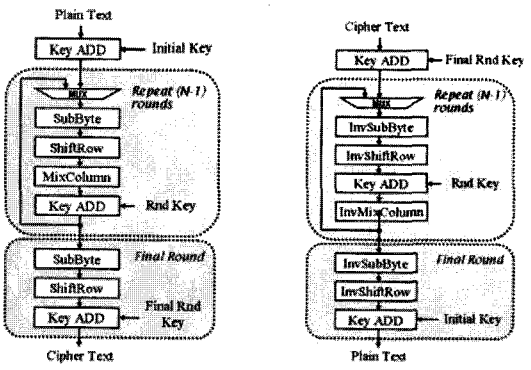
Key wrap 모드의 동작은 data_in을 통하여 입력되는 128비트 TEK를 Reg-B에 저장하고 내부에서 생성되는 64비트 Ⅳ를 Reg-A에 저장한다. Reg-A에 저장된 64비트의 데이터와 Reg-B의 LSB 64비트 데이터를 AES-Enc/Dec로 암호화하며, 그 결과 값의 MSB 64비트는 계수기 출력 t와 XOR 연산하여 Reg-A에 저장하고 LSB 64비트는 Reg-B에 저장한다. 이와 같은 연산을 12번 반복한 후 192비트의 암호화된 TEK를 출력하게 되며, Reg-A 64비트와 Reg-B 128비트 데이터를 출력한다. Key unwrap 모드의 동작은 암호화된 TEK 192비트가 입력되면 MSB 64비트는 Reg-A에 저장되며, LSB 128비트는 Reg-B에 저장된다. Key wrap 모드의 동작과 같은 방법으로 레지스터에 저장된 128비트 데이터를 AES-Enc/Dec로 복호하여 그 결과 값을 레지스터로 다시 저장하게 된다. 이와 같은 연산을 13번 반복한 후 복호된 IV 64비트와 복호된 TEK 128비트를 출력하게 된다.

3. AES 암호·복호기 및 키 스케줄러 설계

그림 4에서 보는 바와 같이, WBSec 프로세서에는 암호연산만을 처리하는 AES-Enc와 암호·복호연산을 처리하는 AES-Enc/Dec 등 두 개의 AES 코어와 라운드 키를 생성하는 키 스케줄러가 사용된다. 본 절에서는 이들 핵심 블록의 설계에 대해 기술한다.

가. AES 알고리듬

AES 코어는 블록 길이와 키 길이가 모두 128비트이고, 10번의 라운드 연산으로 구성되는 AES 알고리듬을 구현한다. AES 알고리듬의 암호·복호 연산과정은 그림 6과 같으며, 초기 라운드 키 가산 후, 9번의 반복 라운드 변환과 최종 라운드 변환의 과정으로 처리된다. 최종 라운드 변환을 제외한 9번의 반복 라운드는 4행×4열로 구성되는 State (128비트 데이터의 4바이트×4바이트 2차원 배열)에대해 SubByte, ShiftRow, MixColumn 및 KeyAdd 등의 변환으로 구성된다. AES의 복호화는 암호화의 역순으로 이루어지며, 라운드 연산의 역 변환 (InvByteSub, InvShiftRow, InvMixColumn)이 사용되고, 라운드 키도 암호화 연산과 역순으로 사용된다.[9-10]



(a) Encryption (b) Decryption

그림 6. AES 암·복호 알고리듬
Fig. 6. AES encryption/decryption algorithm.

나. AES-Enc/Dec 블록

암호·복호 연산을 수행하는 AES-Enc/Dec 블록은 WBSec의 동작속도와 전력소모 등 성능에 큰 영향을 미치는 핵심 부분이다. 그림 7은 설계된 AES-Enc/Dec의 라운드 블록 회로이며, 다음과 같은 두 가지를 고려하여 설계 최적화를 이루었다.

첫째, 암호연산과 복호연산의 하드웨어 공유를 최대화하기 위하여, 암호연산을 위한 SubByte, ShiftRow, MixColumn 연산기와 복호연산을 위한 InvSubByte, InvShiftRow, InvMixColumn 연산기를 결합하여 공유 서브바이트 (Shared SubByte) 블록, 공유 쉬프트로우 (Shared ShiftRow) 블록, 공유 믹스컬럼 (Shared MixCollum) 블록의 구조를 사용하여 구현하였다. 둘째, AES-Enc/Dec에서 가장 큰 하드웨어 면적을 차지하는 공유 서브바이트 (Shared SubByte) 블록의 설계를 최적화하였다. 공유 서브바이트 (Shared SubByte) 블록을 구성하는 $GF(2^8)$ 상의 곱의 역원 연산을 LUT로 직접 구현하는 경우, 256바이트 크기의 LUT가 필요하다. 이 방법의 경우, 암호화 연산을 위한 SubByte와 복호화를 위한 InvSubByte가 각각 독립된 LUT로 구현되어야 하고, 라운드 블록에 8개의 LUT가 사용되어야하므로 매우 큰 하드웨어 면적을 필요로 한다. 본 논문에서는 LUT+affine 변환방법을 적용하여 암호화 연산과 복호화 연산이 4개의 LUT와 affine/inv-affine 변환기로 처리되도록 하였다. 또한, LUT+affine 변환 방법에 사용되는 LUT의 크기를 더욱 줄이기 위해 $GF(2^8)$을 $GF((2^4)^2)$으로 변환하는 체 (field) 변환 연산방식[11-12]을 적용하여 설계하였으며, 이를 통해 공유 서브바이트 연산기를 8바이트의 LUT와 단순한 조합논리회로로 구현하였다. 표 2는 각 방법을 비교한 것이며, 그림 8은

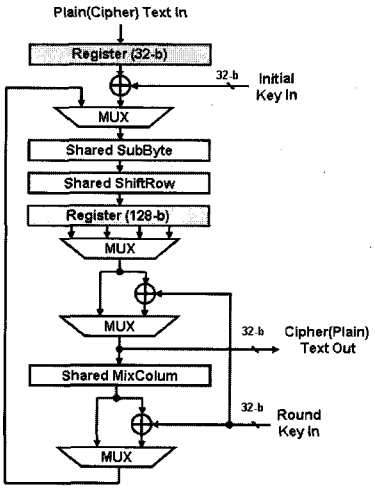


그림 7. AES-Enc/Dec의 라운드 블록 회로
Fig. 7. Round block of AES-Enc/Dec.

표 2. SubByte/InvSubByte 연산기 설계방법 비교
Table 2. Comparison of SubByte/InvSubByte block designs.

구 분	회로 크기
Direct LUT 구현방식	512bytes LUT
LUT+affine 변환방식	256Kbytes LUT + 조합회로
체 변환방식	8bytes LUT + 조합회로

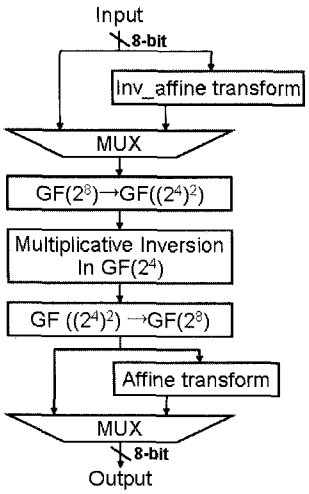


그림 8. 체 변환을 이용한 공유 서브바이트 블록
Fig. 8. Shared SubByte using composite field arithmetic.

설계된 공유 서브바이트 연산기의 구조이다. 암호모드의 경우 8비트 데이터의 곱의 역원을 구한 후 affine 변환을 하며, 복호 모드일 경우 inverse affine 변환 후 곱의 역원을 구한다. $GF(2^8)$에서 $GF((2^4)^2)$로 또는 그 역변환은 단순한 조합논리회로로 구현된다.

다. AES-Enc 블록

암호 연산만을 수행하는 AES-Enc의 라운드 변환 블록은 그림 9와 같으며, SubByte, ShiftRow, MixColumn 연산기와 키 가산을 위한 XOR 게이트 등으로 구성된다. 한편, SubByte 연산기는 AES-Enc/Dec 블록과 동일하게 LUT+affine 변환 방법과 $GF(2^8)$을 $GF((2^4)^2)$으로 변환하는 체 변환 방식을 적용하여 설계하였다.

라. 키 스케줄러 블록

AES 키 확장 알고리듬[9-10]은 128비트의 초기키 (K_0)를 입력받아 이를 seed로 사용하여 매 라운드 연산에 사용되는 키를 생성하며, i-번째 라운드 키 (K_i)가 $(i-1)$-번째 라운드 키 (K_{i-1})로부터 생성되는 chain 형태의 연산구조를 갖는다(단, $1 \le i \le 10$).

키 스케줄러는 128비트의 TEK를 입력 받아 라운드 변환에 사용되는 128비트의 라운드 키를 10번 생성하여 매 라운드 연산마다 라운드 변환블록에 공급한다. 설계된 키 스케줄러의 블록도는 그림 10과 같으며, 4개의 S-Box 블록, 라운드 상수 (Rcon) 생성기, 바이트 단위의 쉬프터 (RotWord), 다수개의 XOR 및 MUX 등으로 구성된다. 입력단은 32비트 레지스터 4개를 쉬프트 레지스터로 구성하여 4 클록주기 동안 128비트의 TEK가

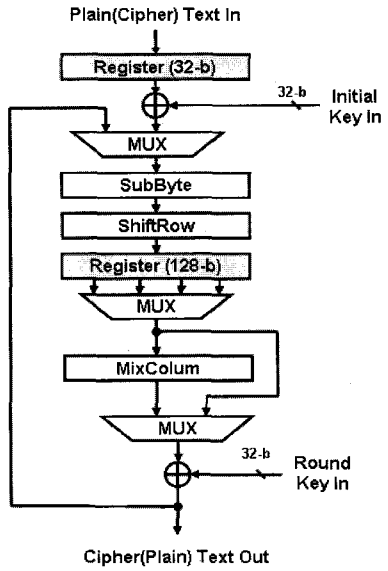


그림 9. AES-Enc의 라운드 블록 회로
Fig. 9. Round block of AES-Enc/Dec.

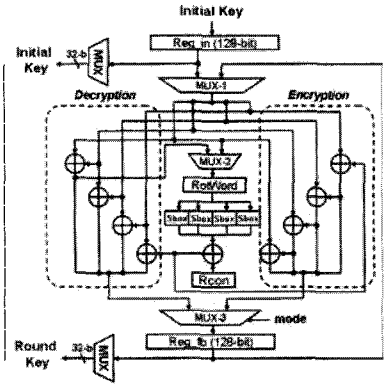


그림 10. 키 스케줄러 회로
Fig. 10. Key scheduler.

입력되도록 하였으며, 생성된 128비트의 라운드 키는 32비트 단위로 4 클록주기 동안 라운드 변환블록에 공급된다. 한편, AES-Enc/Dec에서는 암호연산과 복호연산의 라운드 키가 반대 순서로 사용되며, 따라서 암호화 연산의 마지막 라운드 키가 계산된 후에 이를 초기키로 사용하여 복호화의 라운드 키가 생성되도록 하였다. 암호화의 라운드 키는 오른쪽 부분에서 확장되며, 복호화의 라운드 키는 왼쪽 부분에서 생성되어 MUX-3의 mode 신호에 의해 선택된다.

Ⅳ. 설계 검증 및 성능 평가

설계된 WBSec 프로세서는 Verilog-HDL을 사용하여 RTL 레벨에서 모델링하였으며, ModelSim을 이용하여 동작모드별 기능검증을 수행하였다. CCM 동작모드

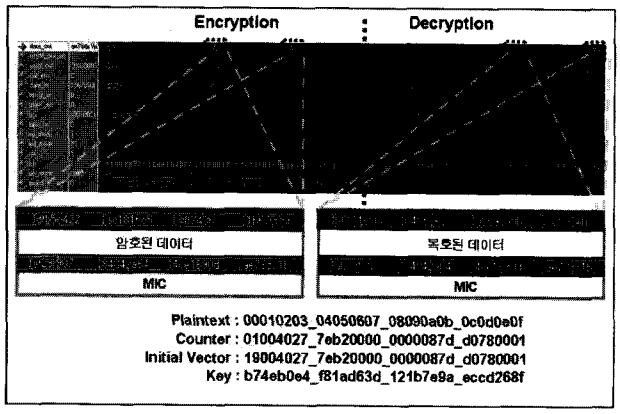


그림 11. CCM 동작모드의 기능검증 결과
Fig. 11. Simulation results of CCM mode.

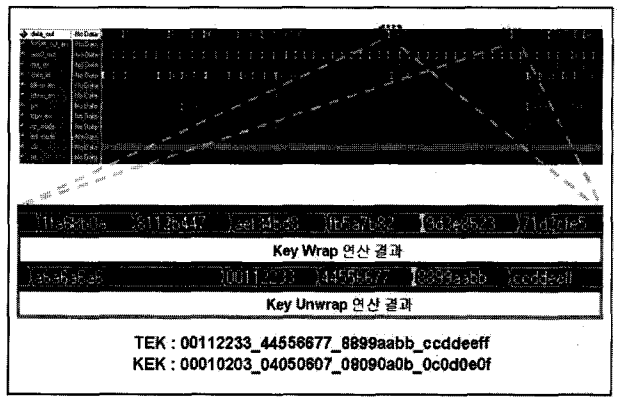


그림 12. Key wrap/unwrap 동작모드의 기능검증 결과
Fig. 12. Simulation results of key wrap/unwrap mode.

에 대한 기능검증 결과는 그림 11과 같으며, CCM 동작모드의 암호과정에서는 평문 *"00010203_04050607_08090a0b_0c0d0e0f"*가 암호 키 *"b74eb0e4_f81ad63d_121b7e9a_eccd268f"*로 암호화되어 암호문 *"dfb5e422_ac1816e5_3fff7436_1b67006f"*와 MIC 값 *"af29f907_84842a64_4ce1690d_ae78dc1b"*가 출력되었으며, 복호과정에서는 암호문 *"dfb5e422_ac1816e5_3fff7436_1b67006f"*가 복호화되어 평문 *"00010203_04050607_08090a0b_0c0d0e0f"*와 MIC 값 *"af29f907_84842a64_4ce1690d_ae78dc1b"*가 출력되었다. 복호과정의 출력이 암호과정의 입력인 평문과 동일하여 CCM 동작모드의 암호-복호과정의 연산이 올바로 동작함을 확인할 수 있으며, 또한 암호과정의 MIC과 복호과정의 MIC 값이 동일하여 CCM 동작모드에 의해 데이터의 무결성이 입증됨을 확인할 수 있다.

그림 12는 key wrap/unwrap 동작모드의 기능검증을 수행한 결과이며, key wrap 동작모드에서 TEK *"00112233_44556677_8899aabb_ccddeeff"*가 KEK *"00010203_04050607_08090a0b_0c0d0e0f"*로 암호화된 결과로 암호화된 key 값 *"1fa68b0a_8112b447_aef34bd8_fb5a7b82_9d3e8623_71d2cfe5"*가 출력되었으며, key unwrap 동작모드에서 복화화된 TEK 값 *"00112233_44556677_8899aabb ccddeeff"*가 출력되어 key wrap/unwrap 동작모드가 정상 동작함을 확인하였다.

기능검증이 완료된 WBSec 코어는 CMOS 셀 라이브러리를 사용하여 논리합성을 수행하였으며, 합성된 게이트 netlist와 타이밍 정보가 포함된 SDF 데이터를 이용하여 pre-layout 타이밍 검증을 수행하였다. 타이밍 시뮬레이션 결과와 기능검증 결과를 비교하여 설계된 WBSec 코어의 모든 기능이 정상동작 함을 확인하였다. Pre-layout 타이밍 검증이 완료된 후, Astro 툴을 이용하여 레이아웃 설계 (P&R)를 하였으며, 코어 면적은

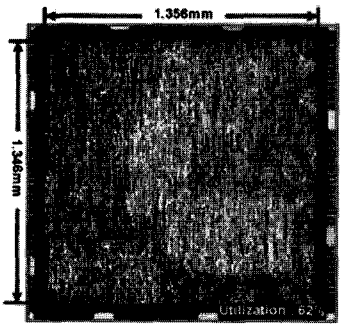


그림 13. WBSec 프로세서의 레이아웃 도면
Fig. 13. Layout of the WBSec processor.

1.36㎜×1.35㎜이며 셀 이용률은 약 62%이다. 그림 13은 완성된 레이아웃 도면을 보인 것이다.

P&R이 완료된 후, post-layout STA (Static Timing Analysis)를 수행하였으며, 100-MHz 클록주파수에 대한 최대지연경로의 worst-case slack이 3.86-ns로 분석되었다. 레이아웃에서 추출된 netlist와 SDF 데이터를 이용하여 post-layout 타이밍 시뮬레이션을 수행하였으며, RTL 레벨의 기능검증 결과와 동일하게 동작하는 것을 확인하였다. 이상과 같은 검증결과로부터 설계된 WBSec 프로세서는 100-MHz의 동작주파수에서 모든 기능이 정상동작 함을 확인하였다.

표 3은 WBSec 프로세서를 구성하는 핵심 블록인 AES 코어 내부의 SubByte/InvSubByte 연산기 설계방식에 따른 게이트 수를 비교한 것이다. 체 변환방식을 적용하여 설계된 본 논문의 방법이 기존의 방법[13]을 적용하여 설계된 경우에 비해 약 25% 적은 게이트 수로 구현되었다. 설계된 WBSec 프로세서의 동작모드에 따른 성능은 표 4와 같으며, ECB 암호/복호 모드와 CCM 암호/복호 모드에서 최대 213-Mbps의 성능을 가지며, key wrap 모드에서 최소 16-Mbps의 성능을 갖는 것으로 평가되었다.

표 3. WBSec 프로세서의 합성 결과 비교
Table 3. Comparison of Synthesis results of the WBSec.

SubByte/InvSubByte 연산기의 설계 방법		게이트 수
기존의 방법	LUT+조합논리회로[13]	29,900 (1.0)
본 논문의 방법	체 변환 방식	22,350 (0.75)

표 4. 설계된 WBSec 프로세서의 성능
Table 4. Features of the WBSec processor.

구분		성능
게이트 수		22,350
코어 면적		1.356 x 1.346 mm^2
동작 주파수		100-MHz@3.3-V
동작 성능	ECB 암/복호 모드	213-Mbps
	CTR 암/복호 모드	183-Mbps
	CBC 암/복호 모드	200-Mbps
	CCM 암/복호 모드	213-Mbps
	key wrap	16-Mbps
	key unwrap	23-Mbps
라운드 키 생성		온라인 방식
데이터 입·출력		32비트
동작모드		ECB, CTR, CBC, CCM key wrap/unwrap

V. 결 론

본 논문에서는 와이브로 무선인터넷 시스템의 보안 부계층에서 정의하고 있는 AES 기반의 CTR, CBC 및 CCM 모드, TEK 암호화를 위한 ECB 모드, key wrap/unwrap 모드 등을 지원하는 와이브로 보안 프로세서(WBSec)의 효율적인 하드웨어 설계에 관하여 기술하였다. 최적화된 하드웨어 구조와 함께 동작속도 향상을 위해 CCM 모드의 CTR 운영모드와 CBC 운영모드가 병렬처리 되도록 두개의 AES 코어를 사용하여 설계하였으며, AES 코어 내부에 서브파이프라인을 삽입하여 동작속도를 향상시켰다. AES 코어 내부의 SubByte/InvSubByte 블록을 체 변환 방식을 적용하여 설계하였으며, 이를 통해 기존의 LUT+조합논리회로 설계방법에 비해 약 25%의 면적을 감소시켰다. 100-MHz@3.3-V로 동작하도록 설계된 WBSec 프로세서는 22,350 게이트로 구현되었으며, key wrap 모드에서 최소 16-Mbps, CCM 암/복호 모드에서 최대 213-Mbps의 성능을 가져 와이브로 시스템용 SoC 설계에 IP 형태로 사용될 수 있을 것이다.

감사의 글

반도체설계교육센터(IDEC)의 CAD Tool 지원에 감사드립니다.

참 고 문 헌

[1] IEEE Standard for local and metropolitan area networks Part 16 : Air Interface for Fixed Broadband Wireless Access Systems Amendment 2 : Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, *IEEE Std 802.16e-2005 and IEEE Std 802.16-2004,* 2006.

[2] 배성수, 최동훈, 최규태, *와이브로 기술과 시스템*, 도서출판 세화, 2006.

[3] Recommendation for Block Cipher Modes of Operation-Methods and Techniques, *NIST Special Publication 800-38A,* U.S. DoC/NIST, Dec., 2001.

[4] Recommendation for Block Cipher Modes of Operation : the CMAC Authentication Mode, *Draft NIST Special Publicatin 800-38B,* U.S. Doc/NIST, Oct., 2003.

[5] Recommendation for Block Cipher Modes of Operation : The CCM Mode for Authentication and Confidentiality, *NIST Special Publication 800-38C,* May 2004.

[6] R. Housley, D. Whiting and N. Ferguson, "Counter with CBC-MAC (CCM) : AES Mode of Operation," Proposed to NIST, June 2002.

[7] 황석기, 김종환, 신경욱, "IEEE 802.11i 무선 랜 보안을 위한 AES 기반 CCMP 코어 설계", *한국통신학회논문지*, 제31권 제6A호, pp.640-647, 2006. 6

[8] AES Key Wrap Specification, *http://csrc.nist.gov/encryption/kms/key-wrap.pdf,* Nov., 2001.

[9] Advanced Encryption Standard (AES), *FIPS Publication 197,* U.S. Doc/NIST

[10] J. Daemen and V. Rijmen, "AES Proposal : Rijndael Block Cipher", NIST Document ver.2, *http://www.nist.gov/aes,* Mar., 1999.

[11] V. Rijndael, "Efficient implementation of the Rijndael S-box", *http://www.esat.kuleuven.ac.be/~rijnmen/rijndael/sbox.pdf*

[12] K. Jarvinen, M. Tommiska, J. Skytta, "Applications: A fully Pipelined memoryless 17.8 Gbps AES-128 encryptor", *Proceedings of the 2003 ACM/SIGDA 11th International symp. on Field Programmable Gate Arrays,* Feb., 2003.

[13] 안하기, 신경욱, "AES Rijndael 블록 암호 알고리듬의 효율적인 하드웨어 구현", *한국정보보호학회논문지*, 제12권 2호, pp.53-64, 2002.

저 자 소 개

김 종 환(정회원)-제1저자
2005년 금오공과대학교 전자공학부 졸업
2007년 금오공과대학교 전자공학과 석사
2007년~현재 픽셀플러스(주) 연구원
<주관심분야 : 정보보안 프로세서설계, SoC 설계, 반도체 IP 설계, ISP(Image Signal Processing>

신 경 욱(정회원)-교신저자
1984년 한국항공대학교 전자공학 학사 졸업
1986년 연세대학교 대학원 전자공학과 석사 졸업
1990년 연세대학교 대학원 전자공학과 박사 졸업
1990년~1991년 한국전자통신연구소 반도체 연구단 선임연구원
1991년~현재 금오공과대학교 전자공학부 교수
1995년~1996년 UIUC 방문교수
2003년~2004년 UCSD 방문교수
<주관심분야 : 통신 및 신호처리용 SoC 설계, 정보보호 SoC 설계, 저전력 설계>