와이브로 보안용 AES기반의 Key Wrap/Unwrap 코어 설계

김종환* · 신경욱** · 전흥우**

A Design of AES-based Key Wrap/Unwrap Core for WiBro Security

Jong-Hwan Kim* · Kyung-Wook Shin** · Heung-Woo Jeon**

이 논문은 2006년도 금오공과대학교 학술연구비 지원에 의한 결과임

요 약

본 논문에서는 휴대인터넷 와이브로 (WiBro) 시스템의 보안계층 중 암호 키 (Traffic Encryption Key; TEK)를 암호·복호하는 key wrap/unwrap 알고리듬의 효율적인 하드웨어 설계에 대해 기술한다. 설계된 key wrap/unwrap 코어 (WB_KeyWuW)는 AES (Advanced Encryption Standard) 알고리듬을 기반으로 하고 있으며, 128비트의 TEK를 128비트의 KEK (Key Encryption Key)로 암호화하여 192비트의 암호화된 키를 생성하고, 192비트의 암호화된 키를 복호화하여 128비트의 TEK로 복호하는 기능을 수행한다. 효율적인 하드웨어 구현을 위해 라운드 변환 블록에 하드웨어 공유기법을 적용하여 설계하였으며, 또한 하드웨어 복잡도에 가장 큰 영향을 미치는 SubByte/InvSubByte 블록을 체 변환 방법을 적용하여 구현하였다. 이를 통해, LUT (Lookup Table)로 구현하는 방식에 비해 약 25%의 게이트 수를 감소시켰다. Verilog-HDL로 설계된 WB_KeyWuW 코어는 약 14,300개의 게이트로 구현되었으며, 100-MHz@3.3-V의 클록으로 동작하여 16~22-Mbps의 성능이 예상되어 와이브로 시스템 보안용 하드웨어 구현을 위한 IP로 사용될 수 있다.

ABSTRACT

This paper describes an efficient hardware design of key wrap/unwrap algorithm for security layer of WiBro system. The key wrap/unwrap core (WB_KeyWuW) is based on AES (Advanced Encryption Standard) algorithm, and performs encryption/decryption of 128bit TEK (Traffic Encryption Key) with 128bit KEK (Key Encryption Key). 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 by using field transformation technique. As a result, the gate count of the WB_KeyWuW core is reduced by about 25% compared with conventional LUT (Lookup Table)-based design. The WB_KeyWuW core designed in Verilog-HDL has about 14,300 gates, and the estimated throughput is about 16~22-Mbps at 100-MHz@3.3V, thus the designed core can be used as an IP for the hardware design of WiBro security system.

키워드

Wibro, Security, Key wrap/unwrap, AES

* 픽셀플러스(주)

** 금오공과대학교 전자공학부 (교신저자 : 신경욱)

접수일자 : 2007. 3. 5

Ⅰ. 서 론

휴대인터넷 방식인 와이브로는 한국정보통신기술협회(TTA)를 중심으로 2003년 6월부터 표준화가 추진되어 최근에는 일부 지역에서 시범서비스가 이루어지고 있는 3.5세대 이동통신 서비스이다. 광대역 무선통신 국제표준인 IEEE 802.16e (WiMAX)에 반영되는 성과를 거두는 등 한국이 국제 표준화를 주도하고 있는 와이브로 시스템은 2.3GHz 주파수 대역을 사용하며, 1 km 이내의 셀 서비스 반경을 갖고 시속 60km/h 이상의 이동 중에도 인터넷에 접속할 수 있는 차세대 이동통신 기술이다. 유선 환경과는 달리 브로드캐스팅 네트워크인 와이브로는 기지국 영역 내에 있는 모든 단말기들이 다른 사람의 송수신 데이터의 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 기밀성을 제공한다. 이를 지원하기 위하여 단말과 기지국간에 전달되는 MAC PDU에 대한 암호화 기능이 적용되어 불법 사용자의 서비스 도난 공격에 대한 강인한 방어능력을 제공한다. 기지국에서는 네트워크 전반에 걸쳐 서비스 플로우에 대한 암호화를 수행하여 데이터 전송 서비스에 어떠한 권한도 없이 접속하는 것을 방지한다. 보안 부계층은 인증된 클라이언트/서버 구조의 키 관리 프로토콜을 이용하여 기지국이 단말에게 키와 관련된 정보를 분배하는 것을 제어한다. 이때 키관리 프로토콜에 디지털 인증서 기반의 단말장치 인증을 추가하여 기본적인 보안 메커니즘의 기능을 더욱 강화시킨다[1].

와이브로의 보안 부계층은 encapsulation 프로토콜과 키 관리 프로토콜로 구성된다. Encapsulation 프로토콜은 광대역 무선 네트워크에서 패킷 데이터의 보안을 위한 프로토콜로서 데이터 암호화 및 인증 알고리즘 등 "cryptographic suites" 집합과 MAC PDU 페이로드에 보안 알고리즘들을 적용시키는 방법을 정의한다. 키 관리 프로토콜은 기지국에서 단말로 키 관련 데이터를 안전하게 분배하는 방법을 제공한다. Crypto- graphic suites는 트래픽 암호키 (Traffic Encryption Key; TEK) 교환, 데이터 암호화 및 인증을 위한 알고리듬을 정의하는 SA (Security Association)의 집합이다. 데이터 암호・복호에 사용되는 알고리듬으로는 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. 트래픽 암호키를 암호화하기 위한 알고리듬으로는 3중 DES, RSA, AES의 ECB 운영모드, AES 기반 key wrap 알고리듬이 사용된다[2].

단말과 기지국 사이에서 수행되는 인증절차에서, 단말은 자신이 지원하는 모든 cryptographic suit 리스트를 기지국에 알리게 되며, 기지국은 이 리스트에서 하나의 cryptographic suites을 선택하여 TEK 암호화, 데이터 암호화 및 인증을 수행하게 된다[2]. 따라서 기지국에서는 cryptographic suit에서 제시하는 TEK 암호화, 데이터 암호화 및 인증을 위한 모든 알고리듬을 SW 또는 HW로 구현하는 것이 필요하며, 단말기에서는 TEK 및 데이터 암호화와 인증을 위한 알고리듬들 중 한가지 이상을 SW 또는 HW로 구현되어야 한다.

본 논문에서는 와이브로 시스템의 보안 메커니즘에서 TEK를 암호・복호화하기 위한 AES key wrap/ unwrap 알고리듬[3]의 효율적인 하드웨어 구현 방법을 제시하였다. 본 논문의 2장에서는 AES 기반 key wrap/unwrap 알고리듬에 대해 기술하며, 3장에서는 key wrap/unwrap 알고리듬의 효율적인 하드웨어 구현에 대해 기술한다. 4장에서는 key wrap/unwrap 코어의 기능검증과 성능평가에 대해 기술하고, 5장에서 결론을 맺는다.

Ⅱ. AES 기반 key wrap/unwrap 알고리듬

AES 기반 key wrap/unwrap 알고리듬은 데이터의 암호・복호에 사용되는 암호 키 (TEK)를 암호・복호화하기 위한 알고리듬으로써 미국 국가기술표준국 (National Institute of Standards and Technology; NIST)에서 제안하였다. AES 기반 key wrap/unwrap 알고리듬은 키를 암호화하는 키 싸기 (encapsulation)와 키를 복호화하는 키 풀기 (decapsulation), 그리고 데이터의 무결성 (integrity)를 검사하는 부분으로 구성된다.

2.1 Key wrap 알고리듬

Key wrap 모드는 데이터 암호・복호에 사용되는 TEK (Traffic Encryption Key)를 KEK (Key Encryption

Key)로 암호화하는 키 싸기 모드이며, 그림 1과 같이 표현되는 pseudo 코드의 연산과정으로 처리된다.

❑ 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 알고리듬
Fig. 1. AES-based key wrap algorithm

연산과정은 6n번의 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 (Initial Vector) *"a6a6a6a6a6a6a6a6"*가 함께 사용된다. 첫 번째 AES 연산은 64비트의 IV와 TEK의 MSB 64비트로 구성되는 128비트를 암호화하며, AES 암호연산 출력의 MSB 64비트는 계수기 출력 t와 XOR되어 다음 AES의 입력 LSB 64비트로 사용된다. 한편, 계수기 출력 t 값은 초기값 0에서 시작하여 AES 연산이 반복될 때 마다 '1'씩 증가된 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 t 와 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.

2.2 Key Unwrap 알고리듬

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 유사하며, 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.

❑ 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
$A^{s-t} = MSB_{64}(AES^{-1}_k((A^t \oplus t) | R^t_n))$
$R^{s-t}_1 = LSB_{64}(AES^{-1}_k((A^t \oplus t) | R^t_n))$
For i = 2,....n
$R^{s-t}_i = R^t_{i-1}$
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 알고리듬
Fig. 2. AES-based key unwrap algorithm

2.3 데이터의 무결성 검증

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 KEK 값에 IV가 숨겨지게 된다. 암호화된 KEK 값을 unwrap 모드로 복호화하면, IV 값이 복호되어 출력된다. 따라서 key wrap 모드에서 사용된 IV와 unwrap 모드에서 복호화된 IV를 비교하여 데이터의 무결성을 검증할 수 있다.

Ⅲ. 회로 설계

3.1 WB_KeyWuW 코어의 아키텍처

AES 기반 key wrap/unwrap 코어(WB_KeyWuW)의 구조는 그림 3과 같으며, AES 암호・복호기, 32비트 레지스터 블록, 동작모드 선택을 위한 MUX 및 제어블록으로 구성된다. 전체적인 하드웨어 복잡도를 고려하여 AES 암호・복호기의 데이터 처리를 32비트 구조로 설계하였으며, 따라서 128비트 데이터의 암호・복호 연산이 66 클록주기에 처리되도록 하였다.

입력 레지스터 블록은 크게 두 부분으로 구성된다. 그림 3에서 오른쪽의 32비트 쉬프트 레지스터 4개에는 TEK 값 128비트와 AES 결과의 LSB 64비트가 순차적으로 입력되며, 왼쪽의 32비트 쉬프트 레지스터 2개에는 IV 64비트와 AES 결과의 MSB 64비트가 순차적으로 입력된다. 제어블록은 wrap 모드와 unwrap 모드를 지정하는 *mode* 신호와 TEK 값의 입력을 지시하는 Ld_TEK 신호를 받아 전체 회로의 동작에 필요한 각종 제어신호를 생성한다.

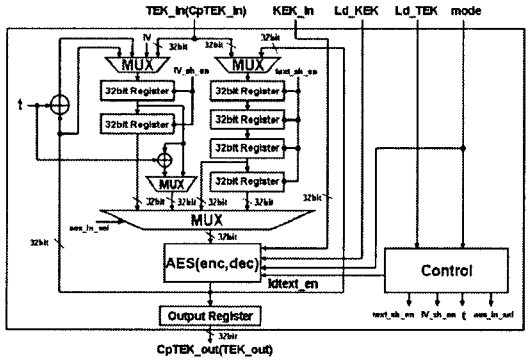


그림 3. AES 기반 key wrap/unwrap 코어
Fig. 3. AES-based key wrap/unwrap core

그림 4는 key wrap 모드의 동작 타이밍도이다. Ld_KEK 신호에 의해 128비트의 KEK 값이 4클록주기 동안 입력된 후, Ld_TEK 신호에 의해 128비트의 TEK 값이 4클록 주기 동안 입력된다. key unwrap 모드 동작과의 호환성을 위하여, Ld_TEK 신호는 TEK가 입력되기 전 2클록주기를 포함하는 6 클록주기 동안 '1'을 유지하도록 설계되었다. TEK의 입력과 동시에 라운드 연산이 시작되며, 12번의 AES 연산을 수행하기 위해 총 805 클록주기가 소요된다.

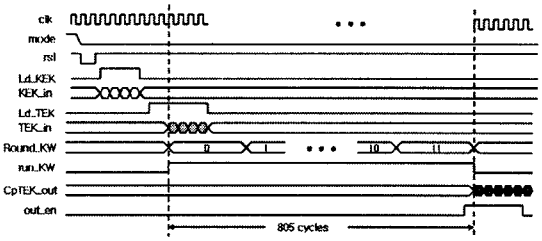


그림 4. Key wrap 모드의 동작 타이밍도
Fig. 4. Timing diagram of key wrap mode

그림 5는 암호화된 KEK 값 192비트를 복호화하기 위한 key unwrap 모드의 동작 타이밍도이다. 192비트의 암호화된 KEK 값이 32비트씩 6 클록주기 동안 입력되며, 13번의 AES 연산을 통해 복호화된 KEK 값이 나오기 까지 873 cycle이 소요된다. key unwrap 모드는 key wrap 모드 보다 한번의 AES 연산이 더 필요하며, 이는 AES 복호 연산에서는 암호연산의 역순으로 라운드 키가 사용되므로, 복호연산의 초기 라운드 키 생성에 필요한 추가적인 AES 연산을 위한 것이다.

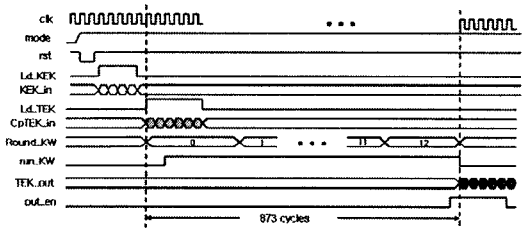
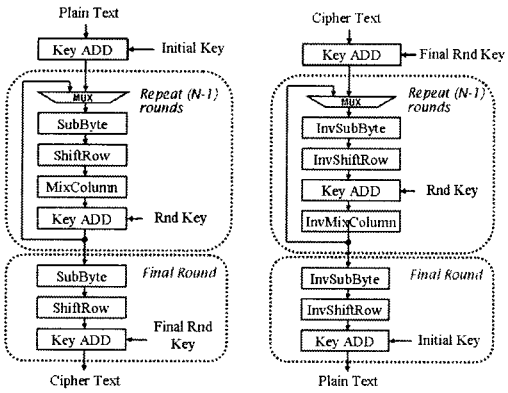


그림 5. Key unwrap 모드의 동작 타이밍도
Fig. 5. Timing diagram of key unwrap mode

3.2 AES 암호・복호기 설계

와이브로 보안에 사용되는 AES 알고리듬은 블록 길이와 키 길이가 모두 128비트이며, 따라서 10번의 라운드 연산으로 구성된다. 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 데이터를 4바이트×4바이트의 2차원 배열로 변환한 것)에 대해 SubByte, ShiftRow, MixColumn 및 KeyAdd 등의 변환으로 구성된다. AES의 복호화는 암호화의 역순으로 이루어지며, 라운드 연산의 역 변환 (InvByteSub, InvShiftRow, InvMixColumn)

이 사용되고, 라운드 키는 암호화 연산과 역순으로 사용된다.



(a) 암호화 연산과정 (b) 복호화 연산과정

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

설계된 AES 암호·복호 코어의 구조는 그림 7과 같으며, 10번의 라운드 변환을 처리하는 라운드 처리부, 라운드 키 생성 블록, 그리고 제어블록 등으로 구성된다. 외부와의 인터페이스는 32비트씩 이루어지며, 라운드 처리부의 입력단에 32비트 레지스터 4개를 쉬프트 레지스터로 구성하여 4 클록동안 128비트의 평문이 입력된다.

AES 키 확장 알고리듬[4]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 구조를 갖는다 (단, $1 \le i \le 10$). 키 스케줄러는 128비트의 KEK를 입력 받아 라운드 변환에 사용되는 128비트의 라운드 키를 10번 생성하여 매 라운드 연산마다 라운드 변환블록에 공급한다. 설계된 키 스케줄러의 블록도는 그림 8과 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 KEK가 입력되도록 하였으며, 생성된 128비트의 라운드 키는 32비트 단위로 4 클록주기 동안 라운드 변환블록에 공급된다. 본 논문의 AES 코어는 암호 연산과 복호연산에서 라운드 키의 순서가 반대로 사용되며, 따라서 암호화 연산의 마지막 라운드 키가 계산된 후에 이를 초기키로 사용하여 복호화 라운드 키가 생성되도록 하였다. 암호화 라운드 키는 오른쪽 부분에서 확장되며, 복호화 라운드 키는 왼쪽 부분에서 생성되어 MUX-3에 입력되는 *mode* 신호에 의해 선택된다.

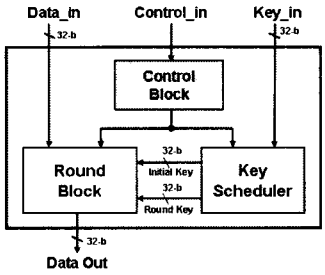


그림 7. AES 암호·복호기의 구조
Fig. 7. AES encryption/decryption core

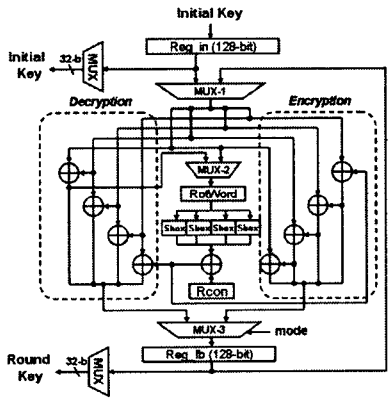


그림 8. 키 스케줄러
Fig. 8. Key scheduler

3.3 라운드 변환블록 설계

설계된 라운드 변환 블록의 내부 구조는 그림 9와 같으며, 암호화 연산과 복호화 연산의 하드웨어 공유가 극대화되도록 공유 서브바이트 (Shared SubByte) 블록, 공유 쉬프트로우 (Shared ShiftRow) 블록, 공유 믹스컬럼 (Shared MixCollum) 블록의 구조를 사용하였다. 데이터 패스는 32-b로 구성되어 4행×4-바이트의 State를 처리하는 서브 파이프라인 단은 4개의 클록으로 구현된다. 라운드 키는 해당 라운드의 전반부 처리가 진행되는 동안 키 스케줄러에서 on-the-fly 방식으로 생성되어 라운드의 후반부 기간에 가산된다.

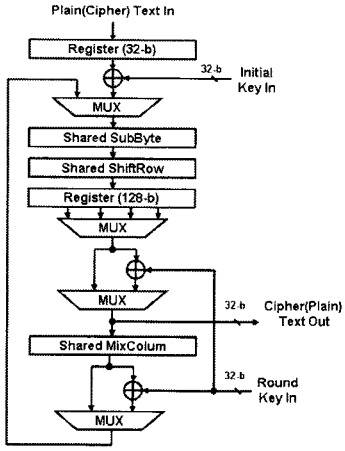


그림 9. AES 코어의 라운드 변환 블록
Fig. 9. Round transformation block of AES core

SubByte/InvSubByte 연산기는 AES 코어에서 가장 큰 하드웨어 면적을 차지하는 부분이며, 따라서 설계 최적화를 위한 고려가 필요하다. 본 논문에서는 하드웨어 면적의 최소화를 위해 다음과 같은 두 가지를 고려하여 설계 최적화를 이루었다.

첫째, 암호연산과 복호연산의 하드웨어 공유를 최대화하기 위하여 공유 서브바이트 (Shared SubByte) 블록 구조를 이용하였다. $GF(2^8)$ 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 변환방법[5]을 적용하여 암호화 연산과 복호화 연산이 4개의 LUT와 affine/inv-affine 변환기로 처리되도록 하였다. 둘째, LUT+affine 변환 방법에 사용되는 LUT의 크기를 더욱 줄이기 위해 $GF(2^8)$을 $GF(2^4)^2)$으로 변환하는 체 (field) 변환 연산방식[6-9]을 적용하여 설계하였으며, 이를 통해 단일 SubByte/InvSubByte 연산기를 8바이트의 LUT와 단순한 조합논리회로로 구현하였다.

그림 10은 위의 두 가지 방법을 적용하여 설계된 Shared SubByte 연산기 구조이다. 암호모드의 경우 8비트 데이터의 곱의 역원을 구한 후 affine 변환을 하며, 복호 모드일 경우 inverse affine 변환 후 곱의 역원을 구한다. 곱의 역원을 구하는 방법은 $GF(2^8)$상의 체를 $GF((2^4)^2)$상의 체로 변환한 후, $GF(2^4)$상에서 곱의 역원을 구하고 다시 $GF(2^8)$상의 체로 변환한다. 본 논문의 AES 연산기는 32비트 데이터 패스 구조를 가지므로 라운드 연산블록과 키 생성블록에 각각 4개의 Shared SubByte 연산기가 사용된다.

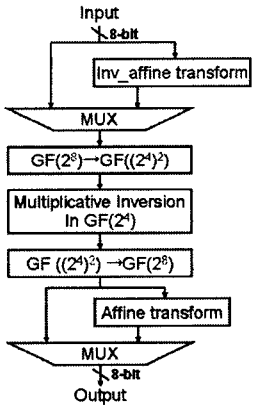


그림 10. 체 변환을 이용한 공유 서브바이트 블록
Fig. 10. Shared SubByte block using composite field

$GF(2^8)$에서 $GF((2^4)^2)$으로의 체 변환은 식(1.1)의 행렬곱셈으로 이루어지며, 그 역변환은 식(1.2)의 행렬곱셈으로 이루어진다. 따라서 체 변환은 XOR 게이트를 이용한 단순한 조합논리 회로로 구현된다. $GF(2^8)$의 원소들은 계수가 $GF(2^4)$상의 원소인 1차 다항식으로 변환이 가능하며 기약다항식이 x^2+Ax+B일 때 임의의 다항식 $cx+d$의 곱에 대한 역원은 식(2)와 같다.

$$\Phi=\begin{bmatrix}1&0&0&0&0&1&0&1\\0&0&1&0&0&0&0&0\\0&0&1&1&1&1&1&1\\0&0&0&1&1&0&0&0\\0&0&0&0&1&1&1&0\\0&1&0&0&1&0&1&1\\0&0&1&1&0&1&0&1\\0&0&0&0&0&1&0&1\end{bmatrix} \quad (1.1)$$

$$\Phi^{-1}=\begin{bmatrix}1&0&0&0&0&0&0&1\\0&0&0&0&1&1&0&1\\0&1&0&0&0&0&0&0\\0&1&0&0&0&0&1&1\\0&1&0&1&0&0&1&1\\0&0&1&0&1&0&1&0\\0&1&1&1&0&0&0&1\\0&0&1&0&1&0&1&1\end{bmatrix} \quad (1.2)$$

$$(cx+d)^{-1}=c(c^2B+cdA+d^2)^{-1}x \quad (2)$$
$$+(d+cA)(c^2B+cdA+d^2)^{-1}$$

식(2)에서 곱셈연산은 AND, XOR 게이트로 구성된 조합논리 회로로 구현되고, 제곱연산은 XOR 게이트로만 구현이 가능하며, 역원(x^{-1})을 구하는 연산은 표 1과 같이 8바이트의 LUT로 구현된다. 따라서 $GF(2^8)$상에서 256바이트의 LUT로 구현하는 기존의 방법 보다 하드웨어 크기를 크게 줄일 수 있다.

표 1. $GF(2^4)$상의 곱의 역원 LUT
Table 1. LUT of Multiplicative inverse on $GF(2^4)$

Input	Inversion	Input	Inversion
0000	0000	1000	1111
0001	0001	1001	0010
0010	1001	1010	1100
0011	1110	1011	0101
0100	1101	1100	1010
0101	1011	1101	0100
0110	0111	1110	0011
0111	0110	1111	1000

3.4 WB_KeyWuW 코어의 ASIC 구현

본 절에서는 WB_KeyWuW 코어를 칩으로 제작하기 위한 설계절차에 대해 기술한다. 결정된 설계사양에 따라 WB_KeyWuW 코어를 Verilog-HDL로 모델링한 후, 모델링된 코어는 ModelSim 툴을 이용하여 기능검증을 수행하였다. 검증이 완료된 회로는 CMOS 라이브러리를 사용하여 Design Compiler 툴로 합성하였다. 회로합성 후, SDF (Standard Delay Format)와 시뮬레이션용 Netlist를 추출하여 STA(Static Timing Analysis)를 통해 타이밍 분석을 수행하였으며, 게이트 레벨 시뮬레이션을 통하여 정상동작을 확인하였다. 검증이 완료된 코어는 Astro 툴을 이용해 P&R을 수행하여 레이아웃 설계를 하였으며, 그림 11은 설계된 WB_KeyWuW 코어의 레이아웃 도면이다. 코어의 면적은 1.155x1.148㎟이며, 셀 이용률은 약 54%이다. 레이아웃이 완료된 코어는 레이아웃에서 추출된 SDF와 parasitic 정보를 이용하여 post-layout STA와 게이트 레벨 시뮬레이션을 통해 최종적인 검증을 수행하였다.

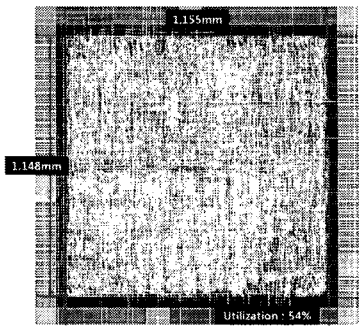


그림 11. WB_KeyWuW 코어의 레이아웃
Fig. 11 Layout of WB_KeyWuW core

Ⅳ. 설계 검증 및 성능 평가

WB_KeyWuW 코어는 Verilog-HDL을 이용하여 RTL 수준에서 설계되었으며, 다양한 테스트 벡터를 이용하여 기능검증을 수행하였다. 그림 12는 시뮬레이션 결과 중 일부를 보인 것이다. Key wrap 모드에서는 TEK *"00112233_4456677_8899aabb_ccddeeff"*를 KEK *"00010203_04050607_08090a0b_0c0d0e0f"*와 IV *"a6a6a6a6_a6a6a6a6"*로 암호화한 결과로 192비트의 암호화된 TEK 값 *"1fa68b0a_8112b447_aef34bd8_fb5a7b82_9d3e8623_71d2cfe5"*가 출력되었다. Unwrap 모드에서는, 이를 다시 동일한 KEK로 복호화한 결과 IV와 TEK가 key wrap 모드의 입력과 동일한 값으로 출력되어 설계된 회로가 정상동작 함을 확인하였다.

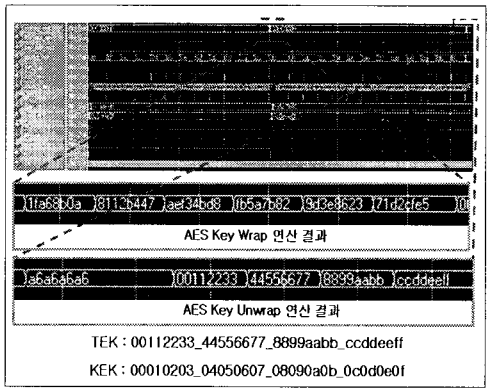


그림 12. WB_KeyWuW 코어의 기능검증 결과
Fig. 12. Simulation result of WB_KeyWuW core

설계된 코어를 0.35-um CMOS 공정으로 합성한 결과 14,300 게이트로 합성되었으며, 100-MHz@3.3-V로 동작 가능할 것으로 평가되었다. AES 코어 내부의 SubByte/InvSubByte 블록을 LUT로 구현하여 동일한 조건으로 합성하는 경우의 약 19,000 게이트에 비하여 본 논문의 설계가 약 25%의 면적 감소가 얻어졌다.

표 2는 설계된 WB_KeyWuW 코어의 특성을 요약한 것이다. Key wrap 모드에서는 16-Mbps의 성능을, 그리고 unwrap 모드에서는 약 22-Mbps의 성능을 갖는다. 무선인터넷 환경에서 데이터의 전송 빈도 보다 키의 전송 빈도가 낮음을 고려하면, 본 논문에서 설계된 WB_KeyWuW 코어는 약 1~30Mbps의 데이터 전송속도를 갖는 와이브로 시스템의 보안 하드웨어 설계에 IP 형태로 사용될 수 있을 것이다.

표 2. WB_KeyWuW 코어의 특성
Table 2. Features of WB_KeyWuW core

구분	성능
게이트 수	14,300 gates
동작 주파수	100-MHz@3.3-V
평균 클럭 수	66 클럭 / 라운드
동작 성능	16 Mbps (Wrap mode) 22 Mbps (Unwrap mode)
코어 크기	1.155mm X 1.148mm
데이터 입·출력	32비트

V. 결 론

와이브로 시스템 보안 알고리즘 중 TEK를 암호화하기 위한 AES 기반 key wrap/unwrap 알고리즘의 효율적인 하드웨어 설계에 대해 기술하였다. 설계된 WB_KeyWuW 코어는 32비트 데이터 패스를 갖는 AES 암호·복호기를 기반으로 구현되었으며, 하드웨어 공유 기법과 체 변환 방법을 적용하여 설계함으로써 기존의 LUT 기반 방식에 비해 약 25%의 면적 감소를 이루었다. 설계된 WB_KeyWuW 코어는 14,300 게이트로 구현되었으며, 100-MHz@3.3-V에서 16~22-Mbps의 성능을 가져 와이브로 시스템의 보안 하드웨어 설계에 IP 형태로 사용될 수 있을 것이다.

참고문헌

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

[2] IEEE Std 802.16e-2005 and IEEE Std 802.16-2004, "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," 2006.

[3] AES Key Wrap Specification, 16 November 2001, http://csrc.nist.gov/encryption/kms/key-wrap.pdf

[4] FIPS Publication 197, "Advanced Encryption standard(AES)," U.S. Doc/NIST

[5] 안하기, 신경욱, "AES Rijndael 블록암호 알고리듬의 효율적인 하드웨어 구현," 한국정보보호학회 논문지, 제12 권2호, pp.53-64, 2002.

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

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

[8] W.A, Arbaugh, "Your 802.11 Wireless Network has No clothes," University of Maryland, Mar. 2001

[9] K. Jarvinen, M. Tommiska, J. Skytta, "Applications: A fully Pipelined memoryless 17.8 Gbps AES-128 encryptor," Proceedings of the 2003 ACM/SIGDA eleventh international symposium on Field programmable gate arrays, February 2003.

※ 반도체설계교육센터(IDEC)의 CAD Tool 지원에 감사드립니다.

저자소개

김 종 환(Jong-Whan Kim)

2005년 2월 금오공과대학교 전자공학과 졸업
2007년 2월 금오공과대학교 전자공학과 공학석사
2007년 3월~현재 : 픽셀플러스(주) 연구원
※관심분야 : 정보보호 SoC 설계, 반도체 IP 설계 ISP(Image Signal Processing)

신 경 욱(Kyung-Wook Shin)

1984년 2월 한국항공대학교 전자공학과(공학사)
1986년 2월 연세대학교 대학원 전자공학과(공학석사)
1990년 8월 연세대학교 대학원 전자공학과(공학박사)
1990년 9월~1991년 6월 한국전자통신연구소 반도체연구단(선임연구원)
1991년 7월~현재 금오공과대학교 전자공학부(교수)
1995년 8월~1996년 7월 University of Illinois at Urbana-Champaign(방문교수)
2003년 1월~2004년 1월 University of California at San Diego(방문교수)
※관심분야 : 통신 및 신호처리용 SoC 설계, 정보보호 SoC 설계, 반도체 IP 설계

전 흥 우(Heung-Woo Jeon)

1980년 한국항공대학 전자공학과 (공학사)
1988년 고려대학교 대학원 전자공학과 (공학박사)
1989년 ~ 현재 금오공과대학교 전자공학부 교수
※관심분야 : 신경망, 영상처리, 집적회로설계