OCB-AES 암호 프로세서의 VLSI 설계

최병윤* · 이종형**

VLSI Design of OCB-AES Cryptographic Processor

Byeong-Yoon Choi* · Jong-Hyoung Lee**

본 논문의 회로 설계에 사용된 회로 소프트웨어는 IDEC 지원에 의한 것임
이 논문은 2005년도 한국정보보호진흥원(KISA) 위탁과제 연구비의 지원을 받았음

요 약

본 논문에서는 암호 기능과 함께 데이터 인증 기능을 지원하는 OCB(offsetset codebook)-AES(advanced encryption standard) 암호 알고리즘을 VLSI로 설계하고 성능을 분석하였다. OCB-AES 암호 알고리즘은 기존 암호 시스템에서 암호 알고리즘과 인증에 구별된 알고리즘과 하드웨어를 사용함에 따른 많은 연산 시간과 하드웨어 문제를 해결하였다. 면적 효율적인 모듈화된 오프셋 생성기와 태그 생성 회로를 내장한 OCB-AES 프로세서는 IDEC 삼성 0.35um CMOS 공정으로 설계되었으며 약 55,700 게이트로 구성되며, 80MHz의 동작 주파수로 930 Mbps의 암 · 복호율을 갖는다. 그리고 무결성과 인증에 사용되는128 비트 태그를 생성하는데 소요되는 클록 사이클 수는 (m+2) × (Nr+1)이다. 여기서 m은 메시지의 블록 수이며, Nr은 AES 암호 알고리즘의 라운드 수이다. 설계된 프로세서는 높은 암 · 복호율 과 면적 효율성으로 IEEE 802.11i 무선 랜과 모바일용 SoC(System on chip)에 암호 처리를 위한 소프트 IP(Intellectual Property) 로 적용 가능하다.

ABSTRACT

In this paper, we describe VLSI design and performance evaluation of OCB-AES cryptographic algorithm that simultaneously provides privacy and authenticity. The OCB-AES cryptographic algorithm solves the problems such as long operation time and large hardware of conventional cryptographic system, because the conventional system must implement the privacy and authenticity sequentially with separated algorithms and hardware. The OCB-AES processor with area-efficient modular offset generator and tag generator is designed using IDEC Samsung 0.35um standard cell library and consists of about 55,700 gates. Its cipher rate is about 930 Mbps and the number of clock cycles needed to generate the 128-bit tags for authenticity and integrity is (m+3) × (Nr+1), where m and Nr represent the number of block for message and number of rounds for AES encryption, respectively. The OCB-AES processor can be applicable to soft cryptographic IP of IEEE 802.11i wireless LAN and Mobile SoC.

키워드

AES, OCB mode, IEEE 802.11i, WLAN security, symmetric key cipher

* 동의대학교 컴퓨터공학과
** 동의대학교 전자공학과

접수일자 : 2005. 6. 13

Ⅰ. 서 론

컴퓨터와 반도체 기술의 발전에 따라 비례적으로 발전하고 있는 암호 분석 기술에 대처하기 위해, 미국 국립 기술 표준청(NIST)은 기존 64-비트 DES(Data Encryption Standard) 알고리즘을 대체하는 새로운 128-비트 암호 알고리즘으로 2000년 10월에 Rijndael을 AES(Advanced Encryption Standard) 알고리즘으로 선정하였다[1-3]. 정보 보호 서비스는 암호 알고리즘에 의해 지원되는 데이터 기밀성(confidentiality) 외에 무결성(integrity), 인증(authenticity), 부인 봉쇄(non-repudiation) 등의 지원이 필요하다. 이러한 나머지 보안 서비스를 위해 해쉬 알고리즘과 디지털 서명 기술들이 개발되었다. 현재 대부분의 암호 시스템은 암·복호 동작, 무결성과 인증 동작에 별도의 알고리즘을 적용하므로, 각 알고리즘의 개별 하드웨어 구현으로 많은 연산 시간과 하드웨어가 필요하다는 문제가 존재한다. 따라서 기존 암호 알고리즘을 사용하여 기밀성처리를 수행하며, 동시에 데이터 무결성과 인증 기능을 지원하는 인증 기능 내장 암호 기법(authenticated-encryption)이 제안되고 있다. 이러한 방식은 암호 알고리즘과 MAC(message authentication code)을 적절히 결합함에 의해서 구현될 수 있다. 그러나 이러한 방식이 기존 암호 방식과 MAC 기능을 합한 것보다 성능이나 하드웨어 면적에서 효율적이어야 한다. 기존 DES 알고리즘의 ECB, CBC, CFB, OFB 등의 동작 모드를 대신하는 대표적인 2가지 동작 모드가 CCM(Counter with CBC-MAC) 모드와 OCB(offset codebook) 모드이다[4-6]. 현재 CCM과 OCB 방식 모두 IEEE 802.11i WLAN의 표준에 채택되었지만, OCB모드는 특허 문제로 선택 구현 사양이며, CCMP(Counter CBC-MAC protocol)만이 의무 구현 사양으로 정의되었다. 본 연구에서는 특허 문제로 CCMP보다 구현 연구가 상대적으로 적지만, 안전성에서 우수하며, 향후 모바일 을 비롯한 다양한 분야에 응용이 될 것으로 판단되는 OCB-AES 알고리즘을 하드웨어로 설계하고 성능을 분석하였다.

본 논문의 2장에서는 OCB 동작 모드와 AES 알고리즘을 살펴보고, 3장에서는 OCB-AES 알고리즘의 하드웨어 구현을 다루고, 4장에서는 하드웨어 검증 및 성능 분석을 다루며, 5장에서는 결론을 기술하였다.

Ⅱ. OCB 모드와 AES 암호 알고리즘

인증 기능을 내장한 암호 동작은 그림 1과 같다.

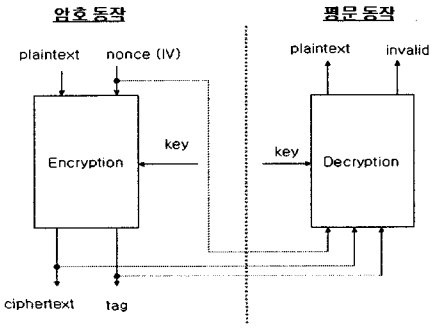

그림 1. 인증 기능을 내장한 암호 동작
Fig.1 Authenticated encryption scheme

암호 동작시 평문, nonce(IV)와 키를 입력으로 하여 암호 알고리즘을 수행하여, 암호문과 태그를 생성한다. 반면 복호 동작시 암호문, 태그와 키를 사용하여 평문을 생성함과 동시에 무결성을 검증하는 Invalid 비트를 생성한다. OCB 모드의 경우 태그의 길이가 t일 경우 공격자가 암호문을 변조할 수 있는 확률은 2^{-t}이다. 따라서 태그 길이로 일반 보안 업무의 경우 64비트, IPSec의 경우 96 비트를 권장되고 있다.

1. OCB 동작 모드

OCB(offset codebook) 모드는 캘리포니아 주립대 교수인 Phillip Rogaway에 의해 제안된 방식으로 Charanjit Jutla가 제안한 IAPM 모드를 수정한 방식이다. OCB 방식의 특징은 다음과 같다. 첫째, OCB 모드의 경우 평문이 블록(block)의 배수일 조건이 필요 없고, 암호문과 평문의 길이가 동일하다. 기존 DES 암호는 평문의 길이가 블록의 배수가 아닐 경우, 마지막 블록의 패딩(padding) 작업에 의해 암호문은 블록의 배수로 나오므로, 평문과 암호문의 길이가 다를 수 있다. 둘째, 오프셋 (offset) 계산이 단순하다. 셋째, 세션 설정이 쉽게 수행 가능하다. 넷째, 하나의 암호 키를 사용하며 확장 정밀도 덧셈(extended precision addition)이 필요치 않다. 다섯째, 블록 암호를 최소로 수행하여 인증 및 암호 동작 구현가능하다. 여섯째, IV(Initial Value) 값으로 난수일 필요가 없다. 즉, nonce값은 세션

내에서만 중복 값이 발생하지 않으면 된다.

OCB 모드를 사용한 암호 알고리즘은 알고리즘 1과 같다. N은 nonce(IV)값을 나타내며, L은 입력 비트를 모두 0으로 한 경우 암호 결과를 나타낸다. 그리고 L(i)은 L(i)=L ·xi 관계를 나타내며, 유한체 곱셈(finite field arithmetic)으로 구현된다. 유한체 곱셈을 위해 정의된 기약 다항식은 식 (1)과 같다.

$$p_{128}(x) = x^{128} + x^7 + x^2 + x + 1 \qquad (1)$$

그리고 Z[i]가 OCB 동작의 핵심인 오프셋(offset)값을 나타낸다. 그리고 γ는 n-비트 canonical gray code이며, 연속적인 코드 값은 식 (2)의 관계를 사용하여 생성된다. 단, 초기 값은 $\gamma_1 = 1$이다.

$$\gamma_i = \gamma_{i-1} \oplus (0^{n-1}1 << ntz(i)), \quad \text{for } 1 \le i \le 2^n - 1 \qquad (2)$$

식 (2)에서 "$0^{n-1}1$"은 n-1비트의 0 다음에 하나의 1이 놓임을 나타낸다. $ntz(i)$는 i에서 최하위 비트의 연속적인 0(trailing zero)의 개수를 나타낸다. 위의 정리를 알고리즘 1의 OCB-E에 적용할 경우, 오프셋 생성과 관련된 식(3)의 관계식을 유도될 수 있다.

$$\begin{aligned} \gamma_i \cdot L &= (\gamma_{i-1} \oplus (0^{n-1}1 << ntz(i))) \cdot L \\ &= (\gamma_{i-1} \cdot L) \oplus ((0^{n-1}1 << ntz(i)) \cdot L \\ &= (\gamma_{i-1} \cdot L) \oplus (L \cdot x^{ntz(i)}) \\ &= (\gamma_{i-1} \cdot L) \oplus (L(ntz(i)) \end{aligned} \qquad (3)$$

식 (3)의 동작을 분석해 보면 $\gamma_i \cdot L$의 경우 이전 반복 루프의 결과($\gamma_{i-1} \cdot L$)와 루프 인덱스 i에 바탕을 둔 $ntz(i)$값을 사용하여 구한 $L \cdot x^{ntz(i)}$을 XOR(eXclusive OR)동작을 수행해서 구할 수 있음을 알 수 있다.

알고리즘 1 : OCB-E 알고리즘
Algorithm 1: OCB-E algorithm

Algorithm OCB-E (N, M)
Partition M into M[1] ···M[m]
L ← $E_k(0^n)$
R ← E_k(N ⊕ L)
for i←1 to m do
  Z[i] ← γ_i · L ⊕ R
for i←1 to (m-1) do
  C[i] ← Ek(M[i] ⊕ Z[i]) ⊕ Z[i]
X[m] ← len(M[m]) ⊕ (L · x^{-1}) ⊕ Z[m]
Y[m] ← E_k(X[m])
C[m] ← Y[m] ⊕ M[m]
C ← C[1] ··· C[m]
Checksum ← M[1] ⊕···⊕ M[m-1] ⊕C[m]$0^{128-len}$⊕Y[m]
T← E_k(Checksum ⊕ Z[m])[first τ bits]
return C* ← C ‖ T

그림 2는 OCB-E에 대한 동작을 도식적으로 나타낸다.

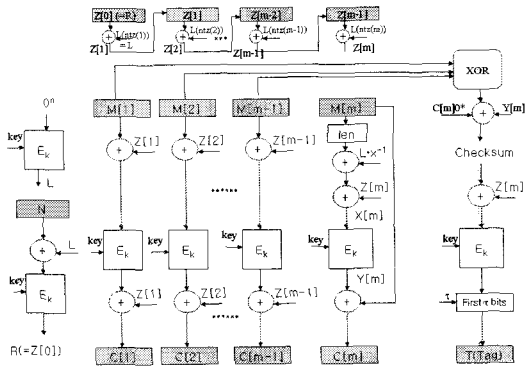

그림 2. OCB 암호 동작의 흐름도
Fig. 2. Illustration of OCB encryption

그림 2를 보면 암호 동작 전에 먼저 초기 오프셋 Z[0]을 계산하고, 암호 블록과 체크섬을 순차적으로 계산한다. 단, 마지막 블록 처리는 다른 블록 처리와 구분된다. 그리고 이러한 암호 블록 생성 동작 후에 체크섬과 마지막 오프셋, Z[m]을 사용하여, 원하는 길이의 태그 값을 생성한다. OCB 모드를 사용한 복호 동작의 경우 암호문에서 평문을 생성하는 과정에 복호 알고리즘이 적용되는 것을 제외하고는 암호 동작과 유사한 동작을 한다[6].

2. AES 알고리즘

AES 암호 알고리즘은 3개의 독립된 역변환 가능한 라운드 변환으로 구성된다[2]. AES은 블록 길이는 128 비트이고, 3가지 키 길이 128, 192, 256 비트를 사용한다. 암호 및 복호 동작에 필요한 라운드 수(Nr)는 키 길이(Nk)에 따라 10, 12, 14로 구성된다. AES 대칭키 암호 알고리즘의 연산 처리 과정은 그림 3과 같이 XOR 연산으로 구성된 초기 라운드 키 가산(AddRoundKey)후에 (Nr-1)번의 반복 라운드 및 최종 라운드의 순서로 처리된다. 최종 라운드를 제외한 각 라운드는 ByteSub, ShiftRow, MixColumn 및 AddRoundKey 등의 라운드 변환동작으로 구성된다. 라운드 키는 암호 알고리즘을 수행하는 과정에 온라인(on-the-fly) 방식으로 생성된다.

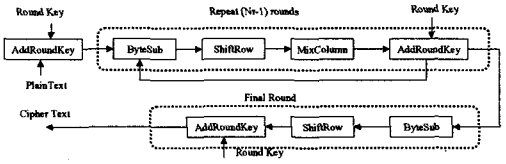

그림 3. AES 암호 알고리즘의 동작 흐름도
Fig. 3. Operational flow of AES encryption

Ⅲ. OCB-AES 프로세서의 하드웨어 설계

본 장에서는 Ⅱ 장에서 기술한 OCB-AES 암호 알고리즘의 하드웨어 구현을 기술한다. 단, AES 알고리즘을 위한 내부 코어는 참고 문헌[8]의 구조를 사용하였기 때문에, 하드웨어 기술은 주로 OCB 동작 모드를 위한 새로운 하드웨어 구현 설계에 초점을 맞추고 기술하였다.

1. 아키텍처 구조 및 설계 사양

본 논문의 암호 프로세서 코어는 IEEE 802.11i 무선랜과 모바일 보안 응용을 고려하여, OCB 동작모드를 지원하며, 내부 암호 알고리즘은 AES 암호 알고리즘을 사용하였다. 그림 4는 OCB-AES 암호 프로세서의 입출력 인터페이스를 나타낸다. OCB-AES 프로세서는 호스트 프로세서에 대해 메모리-사상(memory-mapped) 보조 프로세서로 동작한다. 제어부에 있는 TOT_BC 레지스터는 메시지의 전체 블록 길이를 나타내며, LEN_R 레지스터는 마지막 블록에 있는 타당한 비트의 수를 나타낸다. IV_R은 초기 값, 즉, nonce값을 저장하는 역할을 수행한다. 인증을 포함한 암·복호 동작 중에 다음 처리 블록의 입력과 이전 처리 블록의 출력을 중첩시켜 수행하여 입출력 오버헤드를 제거하기 위해 입출력 레지스터는 이중 버퍼(dual buffer)구조로 구성하였다. 그리고 2개의 동작 개시 신호 중 G_start의 경우 복호 동작을 위한 마지막 라운드 키를 사전에 계산하는 동작과 시스템 초기화를 제어하며 정상적인 암·복호 동작 개시는 L_start 신호를 사용하도록 하였다. 입출력 인터페이스는 32-비트 입출력 인터페이스를 사용하여 PCI, AMBA 등 범용 버스에 쉽게 인터페이스 될 수 있도록 하였다. OCB 동작 모드 회로 설계시 가장 핵심이 되는 블록은 매 사이클마다 오프셋을 생성하는 블록과 체크섬 생성 블록 설계이다.

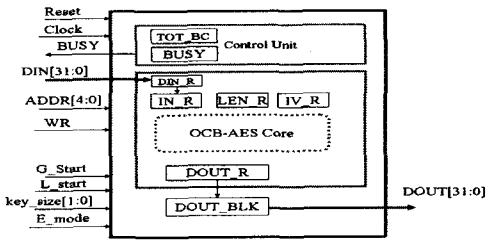

그림 4. OCB-AES 프로세서의 인터페이스
Fig. 4 I/O interface of OCB-AES processor

2. 오프셋(offset) 생성 블록 설계

식 (3)의 오프셋 생성 동작은 식(4)과 같이 변형되어 표현될 수 있다.

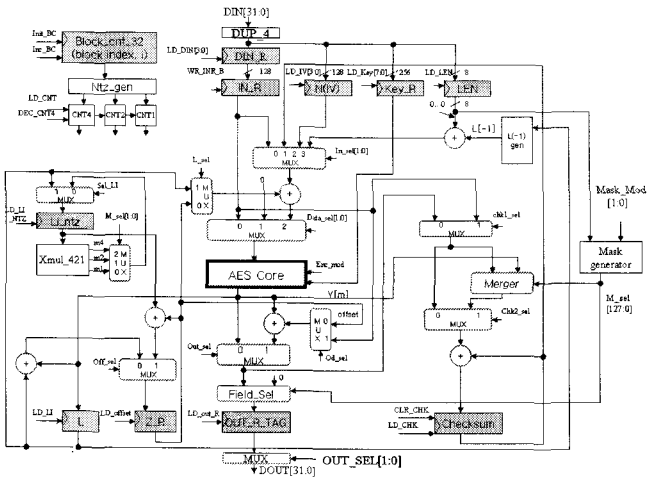

그림 5. OCB-AES 코어 블록도
Fig.5 Block diagram of OCB-AES core

$$L \leftarrow Enc[0^n]$$
$$Z[1] \leftarrow L \oplus R$$
for i=2 to m do
$$Z[i] \leftarrow Z[i\text{-}1] \oplus (L \cdot x^{ntz(i)}) \qquad (4)$$

식 (4)을 구현할 때 핵심이 되는 두 가지 동작은 암호 블록 인덱스를 나타내는 i 값을 받아, $ntz(i)$을 빠르게 계산하는 동작과 유한체 곱셈 $L \bullet x^{ntz(i)}$을 처리하는 동작이다. 이러한 두 가지 동작을 면적 효율적인 구조로 설계하기 위해, 위의 동작을 위한 처리 시간 조건을 먼저 분석하였다. OCB 동작의 경우 블록 수 i는 최대 32 비트로서, 최대 처리 가능 메시지 길이는$(2^{32}-1)\times 128-bit$, 즉 512 GBytes 이며, $ntz(i) \leq 31$의 조건을 갖는다.

본 연구는 참고문헌[8] AES 코어의 라운드 키 on-the-fly 사전 계산 기법과 유사하게 OCB-AES 동작의 평문 블록의 1 블록 시간 전에 오프셋을 생성하도록 하는 오프셋 사전 계산 (offset pre-computation) 기법을 사용한다. 이렇게 함에 의해 오프셋 생성 동작이 최악 전달 경로를 증가시키는 문제를 해결하였다. 이러한 기법이 가능한 이유는 그림 2를 보면 두 번째 AES 암호 동작(Ek)이 R값을 생성하는 과정에 첫 번째 오프셋 값 Z[1]이 Z[1] ← L ⊕ R으로 1 블록 전에 계산이 가능하여, 뒤에 이어지는 정상적인 OCB-암호 동작에 적용 가능하기 때문이다. 따라서 첫 번째 평문이 AES 암호 동작을 수행되는 동안 다음 번 블록에 대한 오프셋이 Z[2]가 생성된다. 이러한 오프셋 사전 계산 기법을 사용할 경우, 식 (4)의 하나의 루프 동작의 오프셋 생성에 할당된 시간 조건은 AES 암호 블록 처리 시간과 같다. 즉, 현재 AES 코어는 키 값이 128-비트, 192-비트, 256-비트인 경우 각각, 11, 13, 15개의 클록 수가 사용되므로, 11 클록 내에 오프셋을 생성해야 한다는 그림 6과 같은 실시간 조건을 도출할 수 있다.

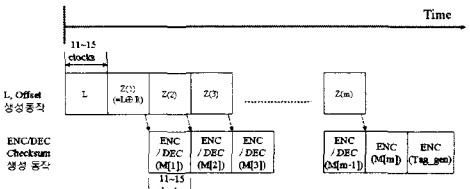

그림 6. 오프셋 값 생성 타이밍
Fig. 6. Timing for generation of offset

본 연구의 OCB-AES 회로는 n-개의 메시지 블록을 처리하기 위해 (n+3)개의 암·복호 동작이 필요하다. 즉 인증 기능을 갖지 않는 기존 AES 블록 암호에 비해 OCB-AES는 3번의 암·복호 동작이 추가적으로 필

요하다. 본 연구에서는 11 사이클의 시간 제약 조건을 고려하여, 그림 7과 같은 오프셋 생성 회로를 구현하였다. 오프셋 동작의 첫 번째 사이클에 ntz(i)를 계산하여, CNT4, CNT2, CNT1에 저장한다. 두 번째 사이클부터 CNT4 값을 순차적으로 1씩 감소시키면서, 0이 될 때 까지 순차적으로 $(L \bullet x^4) \bmod p_{128}(x)$를 누적 수행한다. CNT4가 0이 되면, CNT2 값이 따라 $(L \bullet x^2) \bmod p_{128}(x)$을 수행하고, 마지막으로 CNT1 값에 $(L \bullet x) \bmod p_{128}(x)$을 수행한다. 이러한 최대 9 사이클의 유한체 곱셈 동작 후에 얻어진 결과와 이전 오프셋 값(Z[i-1]) 사이에 XOR 동작을 하여 새로운 오프셋(Z[i])을 생성한다. 블록 수가 2^{31}인 경우 ntz(i)의 최대 값인 31이고, 31= (4 × 7)+ (2 × 1) + (1 × 1)이므로, CNT4는 최대값 이 7이 된다. 이러한 오프셋 생성 방식은 단일 사이클에 128-bit × 32-비트 구조의 유한체 곱셈기를 구현하는 대신에, 128-비트 × 4-비트 곱셈기, 128-비트 × 2-비트 곱셈기, 128-비트 × 1-비트 곱셈기를 9 사이클에 등가 구현하는 기법으로, 직접적인 오프셋 구현 방식에 비해 유한체 곱셈기의 크기를 약 1/8로 감소시킬 수 있는 장점이 있다. 그림 8은 $(L \bullet x^4) \bmod p_{128}(x)$, $(L \bullet x^2) \bmod p_{128}(x)$, $(L \bullet x) \bmod p_{128}(x)$을 단일 하드웨어로 공유하도록 구현한 유한체 곱셈기 회로이다. 그리고 i값에서 최하위 비트에 존재하는 연속적인 0의 개수를 계산하는 NTZ(i) 회로는 그림 9와 같은 모듈화된 회로로 구성된다. 단, 그림 9는 간략화된 16-비트 NTZ[i] 회로를 나타낸다. 여기서 각각의 TOD 블록은 참고문헌[7]의 LOD(Leading One Detector) 구조를 변형하여 구현하였으며, 각각 OR게이트, MUX회로, NOT 게이트로 구성되어 빠른 동작이 가능하다.

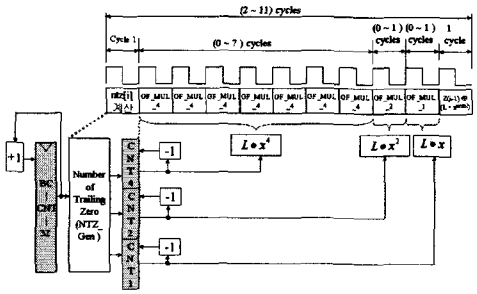

그림 7. 오프셋 발생 회로
Fig. 7 Offset generator circuit

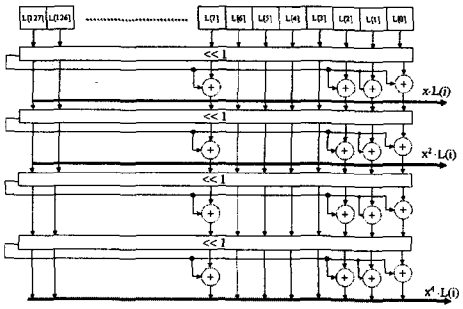

그림 8. 면적 효율적인 유한체 곱셈기
Fig. 8 Area-efficient finite field multiplier

3. 체크섬 과 태그 생성 회로 설계

체크섬 생성시 마지막 블록의 길이(len)는 128-비트가 아닌 경우가 존재한다. 알고리즘 1에서 체크섬 동작의 마지막 연산 $C[m]0^{128-len} \oplus Y[m]$ 은 $C[m] \oplus Y[m] = M[m]$ 관계식을 활용하여 식(5)과 같이 변경할 수 있다.

$$C[m]0^{128-len} \oplus Y[m] = (M[m])\text{의 상위 } len \text{ 비트} \;\|\; (Y[m])\text{의 하위 } (n-len)\text{비트} \quad (5)$$

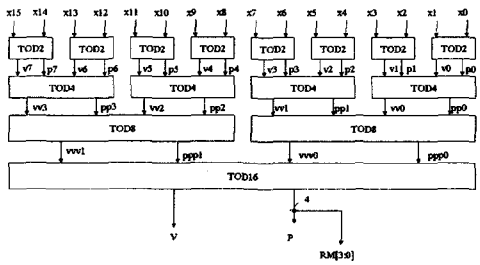

그림 9. 간략화된 16-비트 NTZ(i) 회로
Fig.9 Simplified 16-bit NTZ(i) circuit

여기서 ‖은 비트열 간 연접(concatenation)동작을 나타낸다. 그림 5에서 Merger 부분이 연접 기능을 수행한다. 이러한 연접 동작을 하려면, 마지막 블록에 대한 길이 값(len)을 사용하여 마스크 값을 생성한다. 생성된 마스크 값은 그림 5의 Merger 블록과 Field_Sel 블록의 제어 신호로 사용된다. 마지막 블록 처리와 관련된 $L \bullet x^{-1}$ 연산 회로는 기약 다항식 $p_{128}(x)$에서 $x^{-1} = x^{127} + x^6 + x + 1$을 사용하여 그림 8과 유사한 방식으로 구현된다. 마지막으로 체크섬이 생성되면

Checksum과 Z[m]을 이용하여, 추가의 암호 동작을 사용하여 128-비트 태그를 생성한다. 본 연구에서는 항상 최대 크기인 128-비트 태그를 생성하며, 호스트에서 적절하게 길이를 선택하도록 하였다. 그리고 복호 동작시 인증을 위한 태그(T^*) 생성은 OCB-AES 코어에서 수행하지만, 예상 태그와의 비교 동작은 호스트에서 하도록 하여 128-비트 태그 길이 사용에 대한 융통성을 증가시켰다.

Ⅳ. VLSI 구현 및 성능 분석

본 연구에서 설계한 OCB-AES 회로는 표준안에 정의된 테스트 벡터를 사용하여 올바른 동작을 확인하였다. 그림 10은 평문이 없는 경우에 대한 설계된 OCB-AES 회로의 Cadence NC-Verilog 검증 파형을 나타낸다. 이 경우 메시지 블록 길이는 1로 하며, len 필드는 0으로 설정된다. 이 경우 암호문은 생성되지 않고 태그 값만 출력된다. 설계된 회로는 표준안에 주어진 모든 테스트 벡터를 만족함을 확인한 후에, IDEC 삼성 0.35um CMOS 표준 셀 라이브러리[13]을 사용하여 Synopsys DC 소프트웨어로 합성하였다. 합성 결과 약 55,700 게이트로 동작 주파수가 약 80 Mhz가 나옴을 확인하였다. OCB AES를 지원하지 않은 참고 문헌[8]의 기존 AES 프로세서에 비해 게이트 수는 약 4,000 게이트가 추가됨을 알 수 있다. 즉, OCB모드 추가로 약 8%의 하드웨어 증가가 있음을 알 수 있다. 반면 공정 기술 차이를 반영한 스케일링(scaling)을 적용한 경우, 약 10%의 동작 주파수 감소가 확인되었다.

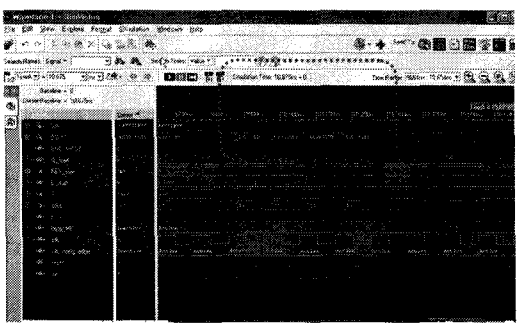

그림 10. OCB-AES 암호 동작
Fig. 10. OCB-AES encryption simulation

그 이유는 체크섬 계산과 관련된 마지막 블록의 경우 AES 코어 출력이 Merger 블록과 XOR 게이트를 거침에 따라 최악 경로의 지연 시간이 약간 증가함에 기인한 것으로 판단된다. 설계된 OCB 회로는 약 80MHz의 동작 주파수를 가지며, 930 Mbps의 암·복호율 특성을 나타낸다.

본 연구의 OCB-AES 코어의 경우 태그 계산은 맨 마지막 암호 동작에서 이루어지므로, 태그 계산에 소요되는 시간은 메시지 블록의 수(m)와 AES 암호의 라운드 수(Nr)에 의해 식(6)으로 정의된다.

$$Tag\text{계산클록 수} = (m+3)\times(Nr+1), \quad (6)$$
$$\text{여기서, } m = \text{메시지 블록 수},$$
$$Nr = AES\text{암호라운드 수}$$

표 1은 설계된 OCB-AES의 동작 특성과 기존 AES, 해쉬 프로세서와의 성능 비교를 나타낸다.

표 1. 성능 비교
Table 1. Performance comparisons

문헌	[9]	[8]	[10]	[11]	본 연구
지원 알고리즘	AES	AES	MD5 SHA-1	MD5	OCB-AES
게이트 수	173,000	51,000	16,684	19,200	55,700
키 길이 (비트)	128,192, 256	128	-	-	128,192, 256
암호 모드	ECB	ECB, CBC, CTR			OCB
암·복호율 (Mbps)	1.820 @256 비트	1,450 @128비트	-	-	930 @128 비트
동작 주파수	100 (MHz)	125 (MHz)	18 (MHz)	133 (MHz)	80 (MHz)
해쉬 처리 블록	-	-	512 비트	512 비트	128 비트
해쉬 블록 처리시간 (클록 수)	-	-	65 @MD5	65 @MD5	11 @AES Tag
구현 공정	0.18um CMOS	0.25um CMOS	Altera FPGA	0.13um CMOS	0.35um CMOS

표 1에 따르면 일반적인 해쉬 코어는 약 20,000게이트가 필요하고, 해쉬 처리를 위해 많은 연산 사이클이 필요하다. 반면 본 연구에서 설계한 OCB-AES 암호 프로세서 코어는 기존 AES 프로세서에 비해 약 8%의 면적 증가와 3번의 추가의 AES 암호 동작으로 태그

값을 생성하므로, AES와 MD5(SHA-1)알고리즘을 결합한 구조에 비해 면적과 속도 측면에서 훨씬 효율적이다. 비록 OCB-AES에서 생성한 태그가 전용 해쉬 함수인 SHA-1 알고리즘에서 생성한 해쉬 값보다 안전하다고는 판단할 수 없지만, 면적과 연산 시간 효율성 측면을 고려할 때, 전력 소비와 면적 제한 조건이 엄격한 RFID 환경과 같이 초경량 암호 및 인증 알고리즘이 요구되는 분야에서는 OCB-AES가 널리 사용될 수 있을 것으로 판단된다.

V. 결 론

본 연구에서 인증 기능과 암호 기능을 내장한 차세대 새로운 암호 방식인 OCB-AES 암호 프로세서 코어를 설계하였다. 면적 효율적인 사전 계산 기능 오프셋 발생회로와 체크섬 계산 회로를 갖는 OCB-AES 암호 프로세서는 약 55,700개의 게이트로 구성되며, 기존 AES 프로세서에 약 8%의 하드웨어 추가로 구현가능하며, 약 930 Mbps의 암·복호율 특성을 갖는다. 설계된 OCB-AES 회로는 단일 하드웨어로 암·복호 동작과 함께 인증 기능을 수행하므로, 암호 알고리즘과 해쉬 알고리즘을 별도로 설계한 기존 암호 프로세서에 비해 연산 성능과 면적 측면에서 효율적이므로, 저전력, 초경량 암호 및 인증 알고리즘이 요구되는 모바일과 유비쿼터스 SoC 칩에 암호 처리 IP(Intellectual Property)로 적용될 수 있을 것으로 판단된다.

참고문헌

[1] William Stalling, *Cryptography and Network Security*, Prentice Hall, 1999.

[2] 박창섭, 암호이론과 보안, 대영사, 1999.

[3] Joan Daemen and Vincent Rijmen, *The Design of Rijndael*, Springer, 2002.

[4] National Bureau of Standards, *DES Modes of Operation*, Federal Information Processing Standard Publication FIPS 81, December 1980.

[5] D. Whiting, etc, "Counter with CBC-MAC(CCM)", Network Working Group RFC3610, September, 2003.

[6] Phillip Rogaway, M. Bellare, and T.Krovetz,, "OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption", 8th ACM conference on computer and communications security(CCS-8), ACM Press, 2001.

[7] V. G. Oklobdzija, "An Algorithm and novel design of a leading zero detector circuit : comparison with logic synthesis", IEEE transaction on VLSI systems, vol.2, no.1, pp.124-128, 1993..

[8] 최병윤, 박영수, 전성익, "모듈화된 라운드 키 생성회로를 갖는 AES 프로세서의 설계", 정보보호학회 논문지, vol.12, no.5, pp.15-25, 2002. 10.

[9] Henry Kuo and Ingrid Verbauwhede, "Architectural Optimization for a 1.82 Gbits/sec VLSI Implementation of the AES Rijndael Algorithm", *CHESS 2001*, pp,51-64 2001.

[10] Young Kyu Kang, Dae Won Kim, Taek Won, and Jun Rim Choi, "An Efficient Implementation of Hash Algorithm for IPSEC", Asia Pacific Conference on ASIC 2002, pp.93-96, August, 2002.

[11] SCI-WORX, "High Speed MD5 Hash Engine", http://www.sci-worx.com/.

[12] Samsung Electronics, "STD90/MDL90 0.35um 3.3V CMOS standard cell library for pure logic/ MDL Products", 2000.

저자소개

최병윤(Byeong-Yoon Choi)

1985년 2월 연세대학교 전자공학과 졸업
현재 동의대학교 컴퓨터공학과 교수

※관심분야 : 그래픽프로세서 설계, SoC 설계

이종형 (Jong-Hyoung Lee)

2005년 5월 Virginia University, Ph.D
2002년 3월 - 현재 동의대학교 전자공학과 교수

※관심분야 : 저전력 CMOS 회로 설계, 광통신