ISSN: 1226-7244 (Print)
ISSN: 2288-243X (Online)
논문번호 15-01-07

j.inst.Korean.electr.electron.eng.Vol.19,No.1,045~051,March 2015
http://dx.doi.org/10.7471/ikeee.2015.19.1.045

IEEE 802.15.4 호환 WPAN 기기를 위한 낮은 복잡도를 갖는 128-bit AES-CCM* IP 설계

Design of Low-Complexity 128-Bit AES-CCM* IP for IEEE 802.15.4-Compatible WPAN Devices

최 인 준*, 이 종 열**, 김 지 훈*★

Injun Choi*, Jong-Yeol Lee**, Ji-Hoon Kim*★

Abstract

Recently, as WPAN (Wireless Personal Area Network) becomes the necessary feature in IoT (Internet of Things) devices, the importance of data security also hugely increases. In this paper, we present the low-complexity 128-bit AES-CCM* hardware IP for IEEE 802.15.4 standard. For low-cost and low-power implementation which is essentially required in IoT devices, we propose two optimization methods. First, the folded AES(Advanced Encryption Standard) processing core with 8-bit datapath is presented where composite field arithmetic is adopted for reduced hardware complexity. In addition, to support CCM* mode defined in IEEE 802.15.4, we propose the mode-toggling architecture which requires less hardware resources and processing time. With the proposed methods, the gate count of the proposed AES-CCM* IP can be lowered up to 57% compared to the conventional architecture.

요 약

최근 IoT(Internet of Things) 기기를 위한 근거리 무선 네트워크 시스템이 널리 활용되면서 점차 보안의 필요성이 증가하고 있다. 본 논문에서는 IEEE 802.15.4 호환 WPAN 기기를 위한 낮은 복잡도를 갖는 128-bit AES-CCM* 하드웨어를 효율적으로 구현하였다. WPAN 기기에서는 하드웨어 자원과 전력 소모가 매우 제한되기 때문에, 다양한 최적화 기법을 적용하여 낮은 복잡도를 갖는 AES-CCM* 하드웨어를 구현해야 한다. 본 논문은 하드웨어의 복잡도를 줄이기 위해 composite field 연산을 채택하면서 8-bit 데이터 패스를 갖는 folded AES processing core를 제안한다. 또한 IEEE 802.15.4 표준에서 정의된 CCM* 모드를 지원하기 위해 적은 하드웨어 자원을 사용하며 응답시간이 빠른 토글 구조의 AES-CCM* 제안한다. 본 논문에서 제안된 AES-CCM* 하드웨어는 기존의 하드웨어의 57%에 해당하는 게이트 수로 구현가능하다.

Key words : AES-CCM, IEEE 802.15.4, Composite field, Mode-toggling, Folding*

* Dept. of Electronics Engineering, Chungnam National University
** Div. of Electronic Engineering, Chonbuk National University
★ Corresponding author, jihoonkim@cnu.ac.kr, 042-821-6585

※ Acknowledgment
This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2012R1A1A1010064) and by the Center for Integrated Smart Sensors funded by the Ministry of Science, ICT & Future Planning as Global Frontier Project (CISS-2011-0031860)
Manuscript received Jan. 6, 2015; revised Feb. 23, 2015 ; accepted Mar. 2. 2015

Ⅰ. 서론

최근 IoT에 대한 관심이 증가하면서 무선 네트워크 시스템을 위한 WPAN 기기가 널리 사용되고 있다. WPAN 기기와 같은 무선 통신 장비가 증가함에 따라 정보 해킹 및 누설과 같은 정보 유출사건이 매해 증가하고 있기 때문에 통신 시스템에서는 점차 보안의 필요성이 증가하고 있다. 특히나 안전하지 않은 환경에서 통신을 하는 WPAN 기기는 데이터의 무결성과 기밀성을 보장하기 위한 암호화 시스템을 지원해야한다. 현재 많은 표준에서 정보 유출을 방지하고 정보의 유효성을 인증하기 위해서 다양한 암호화 기법들을 지원하고 있으며, 그 중 대부분의 암호화 기법은 NIST(National Institues of Standards and Technology)에서 표준 암호 알고리즘으로 채택한 AES(Advanced Encyrption Standard) 암호화 알고리즘을 기반으로 정의되어있다.

저속도 무선 개인 통신망을 위한 표준중 하나인 IEEE 802.15.4 표준은 AES 알고리즘과 블록 암호화 모드중 하나인 CCM(Counter with CBC-MAC)을 사용하는 AES-CCM*을 지원하고 있다. CCM은 데이터를 암호화할 수 있을 뿐만 아니라 데이터의 유효성을 인증할 수 있기 때문에 매우 강력한 보안 시스템을 제공한다. 기존 CCM를 기반으로 한 AES-CCM*은 CCM의 기능을 지원하면서 암호화 레벨에 따라 무결성과 기밀성에 대한 선택적인 서비스를 제공하기 때문에 상황에 적합한 암호문을 효율적으로 생성한다.

IEEE 802.15.4 표준에서 지원하고 있는 AES-CCM*은 다른 운용모드들 보다 보안에 강하지만 데이터 처리량이 많아 실시간 처리를 위한 전용 하드웨어가 필요하다. 또한 WPAN 기기는 메모리 용량, 하드웨어 자원 및 에너지가 제한되기 때문에 낮은 복잡도를 갖는 AES-CCM* 전용 하드웨어를 구현해야한다.

본 논문에서는 IEEE 802.15.4 표준에 호환가능하며 복잡도가 낮은 AES-CCM*을 설계하기 위한 다양한 기법들을 소개하고 있으며 WPAN 기기에 적합한 구조와 기법들을 적용함으로써 기존 하드웨어에 비해 전체 게이트 수를 57%까지 감소시켰다.

Ⅱ. AES의 구조와 최적화 기법

1. AES 암호화 알고리즘

AES는 4*4-byte의 행렬로 구성된 128-bit 블록 안에서 데이터가 처리되는 대칭키 암호화 알고리즘이다. 암호화 과정동안 *State*라고 정의된 128-bit 블록은 네 가지 변환과정을 거치며 AES의 모든 연산은 유한체 GF(Galois Field)(2^8)안에서 이루어진다. 그림 1에서

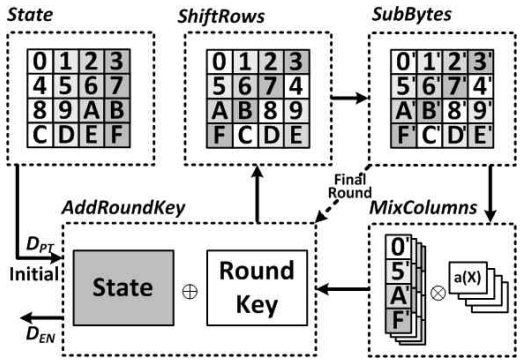


Fig. 1. Data flow of AES encryption algorithm
그림 1. AES 암호화 알고리즘의 데이터 흐름

AES의 전체 암호화 과정을 보여주고 있다. AES는 평문을 입력받아 *Cipher Key*와 XOR한 다음 마지막 라운드를 제외한 매 라운드에서 *SubBytes*, *ShiftRows*, *MixColumns*, 그리고 *AddRoundKey*라고 정의된 네 가지 변환을 수행하고, 마지막 라운드에서 *MixColumns*를 제외한 세 가지 변환을 수행한다[1].

*SubBytes*는 byte 단위의 비선형 치환을 수행하는 변환이다. 16-byte 입력의 각 byte는 Sbox LUT (Lookup Table)에 의해 새로운 byte로 치환된다. Sbox LUT는 GF(2^8) 안에서의 역수를 구하는 연산과 아핀 변환의 조합으로 구현되며 역변환이 가능하다[1]. 전체 *SubBytes*변환은 아래 수식 (1)로 표현 가능하다.

s_i^{-1}은 입력 s_i의 역수를 의미하며 $\oplus$는 GF(2)안에서의 덧셈을 의미한다. c_i는 아핀변환을 위해 더해지는 상수벡터 C {01100011}의 i번째 bit를 나타내는 상수이다. s_0'부터 s_7'의 수식을 모아 행렬식 형태로 간략화 하면, 8×8상수행렬과의 곱과 8×1상수벡터와의 합으로 표현된다.

$$s_i' = s_i^{-1} \oplus s_{(i+4)\bmod 8}^{-1} \oplus s_{(i+5)\bmod 8}^{-1} \oplus s_{(i+6)\bmod 8}^{-1} \oplus c_i \quad (1)$$

ShiftRows 변환에서는 *State*의 2행, 3행, 4행은 각각 1, 2, 3의 offset을 가지고 순환하여 이동된다. 즉, 2행의 입력이 {a b c d}일 때 *ShiftRows*의 출력은 {d a b c}가 된다. *ShiftRows*변환은 추가적인 논리회로 없이 간단한 배선만으로 구현가능하다.

*MixColumns*은 *State*의 각 열(4-byte)을 byte단위로 섞어 새로운 열로 변환한다. *MixColumns*변환을 수식으로 표현하면 행렬식 (2)와 같다. 행렬의 각 상수들은 16진수로 표현되어 있으며, A, B, C, D는 각

각 *State*의 1열, 2열, 3열, 4열을 의미하며 A', B', C', D'는 각각 *MixColumns* 변환의 출력 열을 의미한다.

$$\begin{bmatrix} A' \\ B' \\ C' \\ D' \end{bmatrix} = \begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \quad (2)$$

AddRoundKey 변환은 현재 라운드의 *State*와 *RoundKey*를 XOR연산하여 수행된다. *RoundKey*는 *Key Expansion*에 의해 구성된 key schedule로부터 생성된다. 현재 라운드의 *State*는 초기 *Cipher Key*가 아닌 *Key Expansion*에 의해 확장된 새로운 *RoundKey*가 더해지기 때문에 보안에 더욱 강하다. IEEE 802.15.4 표준의 경우 총 11개의 *RoundKey*를 생성하며 초기 *RoundKey*는 *Cipher Key*에 의해 설정된다.

2. 저비용 AES Processing Core 제안

가. Composite Field Arithmetic을 이용한 최적화

전체 시스템의 성능을 개선할 때 성능 향상 비율이 가장 큰 부분을 최적화하는 것이 중요하다. 본 논문은 AES알고리즘에서 가장 복잡하고 하드웨어 자원을 많이 필요로 하는 *SubBytes*변환을 최적화하기 위해 일반적으로 사용되는 LUT방식과 on-the-fly방식을 비교하고 WPAN 기기에 적합한 방식을 제안한다.

일반적인 LUT방식은 수식 (1)의 모든 연산결과를 미리 메모리에 저장한 (256×8)-bit ROM을 기반으로 사용한다. 또한 암호화/복호화를 모두 지원하기 위해서는 Sbox LUT와 Sbox^{-1} LUT 두 개가 필요하기 때문에 게이트 수가 많고 전력소모가 크다. 이러한 문제점은 역수 연산 하드웨어와 아핀변환을 분리함으로써 해결할 수 있다. Sbox LUT와 Sbox^{-1} LUT는 역수 연산 하드웨어를 공유할 수 있기 때문에 GF에서의 역수만을 미리 계산한 LUT를 만들면 효율적으로 암호화/복호화 가능한 *SubBytes*변환 하드웨어를 설계할 수 있다.

메모리를 사용하지 않는 on-the-fly기반의 *SubBytes*변환 하드웨어는 조합논리회로만으로 구현할 수 있지만 매 라운드 $GF(2^8)$에서의 역수를 계산하고 아핀변환을 수행해야하며 $GF(2^8)$에서의 역수를 직접 구하는 것은 매우 복잡하다. 이를 간단하게 만들기 위해 본래 유한체 $GF(2^8)$와 동형인 부분체에서 연산을 수행하는 CFA(Composite Field Arithmetic)을 사용한다. CFA를 사용하여 역수를 구하기 위한 하드웨어는 기존의 $GF(2^8)$에서 수행하는 하드웨어 보다 간단하게 구현할 수 있다.

CFA를 수행하기 위해서는 우선 $GF(2^8)$의 원소들을 isomorphic composite field로 대응시켜야한다. $GF(2^8)$의 원소를 낮은 위수의 composite field로 대응시키기 위해서 isomorphism function δ가 사용되며 δ를 구하기 위해서는 composite field를 구성하는 기약다항식을 정해야한다. [2]에서는 기약다항식을 수식 (3)과 같이 정하고 있으며, 이때 상수φ와 λ는 각각 10_2과 1100_2으로 정의되었다.

$$\begin{cases} GF(2) \rightarrow GF(2^2): P_0(x) = x^2 + x + 1 \\ GF(2^2) \rightarrow GF(2^4): P_1(x) = x^2 + x + \varphi \\ GF(2^4) \rightarrow GF(2^8): P_2(x) = x^2 + x + \lambda \end{cases} \quad (3)$$

δ은 [3]의 search 알고리즘으로 찾을 수 있다. 수식 (3)의 기약다항식으로부터 composite field의 생성원소 β와 extension field의 생성원소 α를 찾고, 본래 유한체를 생성하는 원시 다항식의 근이 될 수 있는 α^k를 찾는다. 이때 $1 \le k \le 254$이며, α^k를 찾으면 δ와 δ^{-1}가 수식 (4)와 같이 결정된다.

$$\delta = \begin{pmatrix} 1&1&0&0&0&0&1&0 \\ 0&1&0&0&1&0&1&0 \\ 0&1&1&1&1&0&0&1 \\ 0&1&1&0&0&0&1&1 \\ 0&1&1&1&0&1&0&1 \\ 0&0&1&1&0&1&0&1 \\ 0&1&1&1&1&0&1&1 \\ 0&0&0&0&0&1&0&1 \end{pmatrix}_2 \quad \delta^{-1} = \begin{pmatrix} 1&0&1&0&1&1&1&0 \\ 0&0&0&0&0&1&1&0&0 \\ 0&1&1&1&1&1&0&0&1 \\ 0&1&1&1&1&1&0&0&0 \\ 0&1&1&0&1&1&1&0 \\ 0&1&0&0&0&1&1&0 \\ 0&0&1&0&0&0&1&0 \\ 0&1&0&0&0&1&1&1 \end{pmatrix}_2 \quad (4)$$

$GF(2^8)$의 원소가 composite field $GF((2^4)^2)$로 대응되면 경우 기존의 원소는 $s_h x + s_l$로 표현되며, s_h과 s_l는 $GF(2^4)$의 원소이고 x는 $P_1(x)$의 근 이다. 이때, Karatsuba-Ofman 알고리즘을 적용하면 composite field $GF((2^4)^2)$에서의 역수를 구하는 식은 수식 (5)로 간략화 된다[3].

$$(s_h x + s_l)^{-1} = s_h \Delta x + (s_h + s_l)\Delta \quad (5)$$
$$\Delta = (s_h^2 \lambda + s_l(s_h + s_l))^{-1}$$

수식 (5)에 따라, composite field $GF((2^4)^2)$에서의 역수를 구하는 하드웨어 모듈을 그림 2의 구조로 구현할 수 있다. Composite field에서 입력의 제곱을 구하는 squaring(x^2)과 상수 곱셈기($\times\lambda$)는 XOR 게이트의 조합으로 간단히 구현된다. $GF(2^4)$안에서의 곱셈기 ($\times$)은 $GF(2^2)$에서의 곱셈기로 분해될 수 있기 때문에 GF(2)에서의 곱셈기로 구현가능하다.

또한 그림 2를 구성하는 모듈 중 $GF(2^4)$에서의 역수를 구하기 위한 모듈은 다양한 방법들로 구현가능하며,

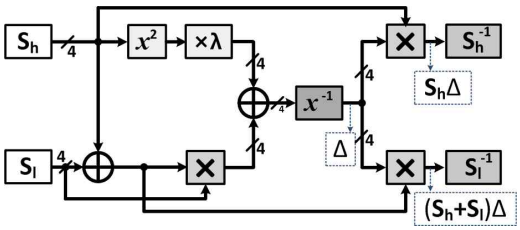


Fig. 2. Implementation of the multiplicative inversion in composite field GF($(2^4)^2$)
그림 2. Composite field GF($(2^4)^2$)에서의 역수 연산 구현

그 중 수식 (6)를 이용하여 직접 구하는 방법이 게이트 수가 적고 critical path가 가장 짧다[4].

$$\begin{cases} x_3^{-1} = x_3 + x_3x_2x_1 + x_3x_0 + x_2 \\ x_2^{-1} = x_3x_2x_1 + x_3x_2x_0 + x_3x_0 + x_2 + x_2x_1 \\ x_1^{-1} = x_3 + x_3x_2x_1 + x_3x_1x_0 + x_2 + x_2x_0 + x_1 \\ x_0^{-1} = x_3x_2x_1 + x_3x_2x_0 + x_3x_1 + x_3x_1x_0 + x_3x_0 \\ \qquad + x_2 + x_2x_1 + x_2x_1x_0 + x_1 + x_0 \end{cases} \quad (6)$$

표 1은 구현 방법에 따라 Sbox의 게이트 수를 비교한 수치를 보여주고 있다. CFA를 적용한 Sbox는 LUT기반의 Sbox의 33.5%에 해당하는 게이트 수만으로 구현된 것을 볼 수 있다. 메모리를 사용하는 LUT 기반의 Sbox 모듈은 게이트 수가 많고 전력소모가 크기 때문에 WPAN 기기에는 적합하지 않다. 본 논문에서는 하드웨어의 복잡도를 낮추기 위해 메모리를 사용하지 않는 on-the-fly방식을 사용하였다. 그 중 가장 게이트 수가 적고 지연 시간이 짧은 composite field GF($(2^4)^2$)에서의 연산을 기반으로 *SubBytes*변환 하드웨어를 구현하였다.

나. Folded AES Processing Core

AES 암호화 알고리즘은 블록 단위로 암호화 하는 대칭키 암호 시스템으로 매 라운드 하나의 *state* 즉 128-bit를 입력받아 각각의 변환과정을 거친다. 기존의 128-bit AES processing core의 경우 현재 *state*의 모든 연산을 병렬로 처리하기 위해 16개의 Sbox와 4개의 상수 행렬곱셈기를 가지고 *SubBytes*변환과 *Mixcolumns*변환을 위한 하드웨어를 구현한다. AES 모듈에 여러 개의 Sbox와 상수행렬 곱셈기를 추가할 경우 높은 데이터 처리량을 갖는 하드웨어를 구현할 수 있지만 여러 개의 중복되는 하드웨어를 병렬로 배치해야 하므로 많은 하드웨어 자원을 요구한다. 반면에 folded 구조의 AES 모듈은 중복되는 하드웨어를 공유하여 보다 작은 수의 게이트로 AES 모듈을 구현할 수 있다.

본 논문에서는 16개의 Sbox와 4개의 *MixColumns*를

Table 1. Gate counts of the three designs of Sbox
표 1. 세 가지 Sbox 디자인의 게이트 수

	LUT based Sbox	CFA Sbox	
		En/decryption	Encryption
Gate count (K)	0.86 (100%)	0.39 (45.3%)	0.29 (33.5%)

Table 2. Synthesis results of various AES designs
표 2. 다양한 AES 디자인의 합성 결과

	[7]			Proposed AES
Bit width (Bits)	128	32	8	8
Sbox gate count (K)	N/A	N/A	N/A	0.29
Total gate count (K)	15.98 (100%)	7.73 (48.3%)	4.02 (25.1%)	3.50 (21.9%)

공유함으로써 최종적으로 Sbox와 *MixColumns*모듈을 1개씩만 사용하는 8-bit folded AES 구조를 채택했으며, CFA가 적용된 on-the-fly 방식의 Sbox를 사용하여 복잡도가 낮은 8-bit AES 모듈을 구현하였다. CFA기반의 Sbox를 채택한 본 논문의 AES 모듈은 LUT기반의 Sbox를 사용한 [7]에 비해 낮은 복잡도를 갖으며 표 2와 같이 65nm공정에서 6MHz동작 주파수로 합성한 결과 [7]의 128-bit AES 모듈의 전체 게이트 수의 21.9%에 해당하는 게이트만으로 하드웨어를 구현할 수 있었다.

Ⅲ. IEEE 802.15.4 표준 기반 AES-CCM*

1. AES-CCM* 알고리즘

AES의 여러 운용모드 중 하나인 CCM(Counter with CBC-MAC)모드는 인증은 위한 CBC-MAC (Cipher Block Chaining Message Authentication code) 모드와 암호화를 위한 CTR(counter) 모드를 함께 운용하는 모드이다. 메시지의 신뢰성을 보장하기 위한 인증코드를 생성하고 메시지의 암호화를 수행하기 때문에 무결성과 기밀성을 모두 보장한다. 인증을 위한 CBC-MAC는 AES의 운용모드 중 하나인 CBC(Cipher Block Chaining)모드를 통해 인증 코드를 생성하는 모드이다. CBC모드는 이전 블록의 암호문과 다음 블록의 평문을 XOR하여 뒤섞은 다음 AES 암호화를 수행하는 방식으로 암호문 블록사이에 종속성을 부여한다. 때문에 최종 암호문은 전체 메시지에 종속되어 전체 평문과 key를 알고 있어야만 생성가능하다. CBC모드의 최종 암호문을 MAC (Message Authentication Code)로 정의하고 MAC의 종속성을 통해 전체 데이터의 무결성을 보장한다.

CCM모드의 암호화는 CTR모드를 통해 수행된다.

CTR모드는 *x*-bit의 Nonce 데이터와 *i*-bit의 정수로 이루어진 일련의 스트림을 통해 CTR 블록을 정의하며 CTR블록이 128-bit가 되도록 *x*와*i* 의 크기를 결정한다. CTR블록은 AES 암호화과정을 거치며, 암호화된 CTR블록과 메시지 블록을 XOR연산하여 최종 암호문은 생성한다. 또한 CTR모드에서의 복호화는 복호화를 위한 AES가 따로 필요하지 않기 때문에 암호화 시스템의 복잡도들 낮출 수 있다.

AES-CCM모드는 MAC를 계산하기 위한 CBC모드와 메시지 암호화를 위한 CTR모드를 모두 운용해야 한다. 그림 3은 AES-CCM 알고리즘의 데이터 처리 과정을 나타내고 있다. IV(Initial Vector)는 인증 데이터의 길이와 메시지의 길이 등의 정보를 담고 있으며, CBC모드의 체인 중에 가장 처음에 위치하는 초기벡터로 사용된다. IV가 암호화 되면 인증 데이터(ADD)를 입력받아 MAC데이터를 계산한 다음 메시지에 대한 인증 코드 계산과 암호화를 수행하게 된다. 이때 MAC을 생성하고 메시지 블록을 암호화하기 위해서 하나의 메시지 블록에 대해 CBC모드와 CTR모드를 모두 수행해야한다.

IEEE 802.15.4 표준에 채택된 AES-CCM*는 기존 AES-CCM를 기반으로 한 암호화 알고리즘으로 암호화 레벨에 따라 무결성과 기밀성에 대한 선택적인 서비스를 제공한다. AES-CCM*는 총 8가지 암호화 레벨에 따라 암호화와 인증을 각각 처리하거나 둘을 동시에 처리할 수 있기 때문에 상황에 적합한 암호문을 효율적으로 생성한다[5].

2. 토글 구조의 AES-CCM* IP 제안

AES-CCM* 하드웨어는 암호화 레벨에 따라 상황에 맞는 암호화를 수행하고 인증 코드를 생성해야하기 때문에 CBC모드와 CTR모드를 선택적으로 운용할 수 있는 하드웨어로 구성 되어야한다. AES-CCM* 전용 하드웨어는 메시지 블록을 처리하는 방식에 따라 직렬 구조, 병렬 구조, 그리고 토글 구조로 나뉜다.

직렬 구조는 CBC모드를 운용하여 전체 데이터에 의해 생성되는 MAC을 먼저 계산한 다음 CTR모드를 운용하여 메시지 블록을 암호화 한다. 각 모드를 순서대로 운용하기 때문에 하나의 AES 모듈만을 가지고 CBC와 CTR 모드를 모두 지원한다. CBC모드를 먼저 운용할 경우 하드웨어의 반응 시간이 길어지고 입력 블록을 통제할 필요가 있지만 작은 크기로 구현 가능하다는 장점이 있다.

병렬 구조는 CBC모드와 CTR모드를 동시에 운용하기 위해서 두 개의 AES 모듈을 사용한다. 하나의 AES는 CBC모드에서 MAC를 계산하기 위해서 사용되며 다른 하나는 CTR모드에서 메시지를 암호화 한다. 두 개의

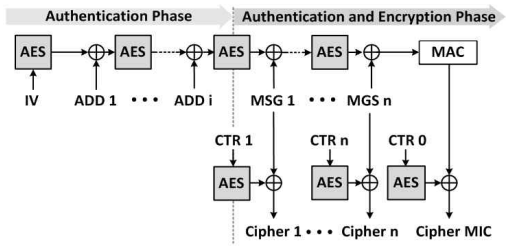


Fig. 3. Data flow of AES-CCM algorithm
그림 3. AES-CCM 알고리즘의 데이터 흐름

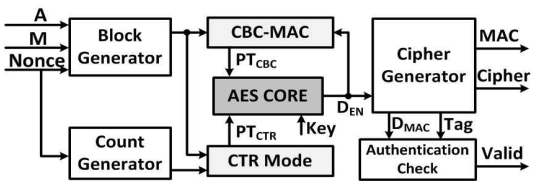


Fig. 4. The block diagram of the proposed AES-CCM*
그림 4. 제안된 AES-CCM*의 블록선도

모드를 동시에 처리하기 때문에 인증 데이터에 대한 MAC계산이 끝나고 나면 CTR모드를 통해 메시지에 대한 암호화를 시작한다. 직렬 구조에 비해 하드웨어의 반응 시간이 빠르며, 데이터 처리량 역시 높다. 그러나 두 개의 AES 모듈을 사용하기 때문에 직렬 구조보다 큰 면적으로 구현된다. 마지막으로 토글 구조는 하나의 AES 모듈을 가지고 두 개의 모드를 번갈아 가며 운용한다. 즉, 인증 데이터에 대한 MAC계산이 끝난 다음 메시지가 입력되는 구간에서 메시지를 암호화하고 MAC를 계산할 때 CTR모드와 CBC모드를 번갈아 가며 운용하여 되는 구간에서 메시지를 암호화하고 MAC를 계산할 때 CTR모드와 CBC모드를 번갈아 가며 운용하여 하나의 AES 모듈만으로 CCM* 모드를 지원한다. 또한 인증 데이터에 대한 계산이 끝나자마자 메시지에 대한 암호화를 시작하기 때문에 병렬 구조와 같은 반응 시간으로 데이터를 처리할 뿐만 아니라 직렬 구조와 같은 크기로 구현 가능하다[6].

WPAN 기기는 하드웨어 자원이 매우 제한되기 때문에 병렬 구조의 하드웨어는 적합하지 않다. 본 논문에서는 작은 면적으로 하드웨어를 구현할 수 있는 구조 중 좀 더 효율적으로 데이터를 처리할 수 있는 토글 구조의 AES-CCM* 하드웨어를 설계하였다.

Ⅳ. AES-CCM* 구현 및 비교

복잡도가 낮은 하드웨어를 구현하기 위해, 본 논문은 8-bit folded AES모듈을 사용하는 토글 구조의 AES-

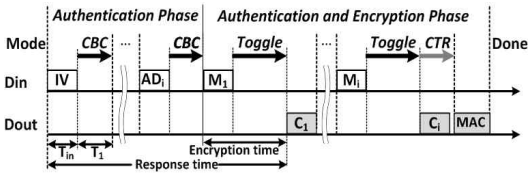


Fig. 5. The processing time of the proposed AES-CCM* IP
그림 5. 제안된 AES-CCM* IP의 처리 시간

CCM* 하드웨어를 구현하였다. 구현된 하드웨어는 하나의 AES모듈로 CCM*모드를 운용하기 위해서 두 개의 상태로 나누어졌다. 하나의 AES모듈만을 사용하는 구간에서는 CBC모드 또는 CTR모드 중 하나의 모드만 동작하며 두 개의 AES을 사용하는 메시지 입력 구간에서는 토글 상태로 전환하여 CTR모드와 CBC모드를 차례대로 운용한다.

그림 4는 본 논문에서 구현한 AES-CCM* IP의 알고리즘을 나타내는 블록선도이다. Block Generator와 Count Generator는 인증데이터 A, 메시지 M, 그리고 Nonce 데이터를 입력받아 암호화와 인증에 사용될 데이터 블록을 생성한다. AES 모듈은 CBC- MAC 모듈과 CTR Mode 모듈에서 보내는 평문 PT를 받아 암호화한 데이터 D_{EN}을 CBC-MAC 모듈로 전달하여 block-chain을 구성하고 Cipher Generator 모듈로 전달하여 암호문과 MAC를 생성한다. CBC모드를 통해 생성된 MAC는 CTR모드에서 한 번 더 암호화 되어 암호화된 MAC를 생성한다. 모듈로 전달하여 암호문과 MAC를 생성한다. CBC모드를 통해 생성된 MAC는 CTR모드에서 한 번 더 암호화 되어 암호화된 MAC를 생성한다. Authentication check 모듈에서는 수신기에서는 전달받은 MAC데이터를 복호화 한 D_{MAC}와 복호화 된 메시지로 수신단에서 생성한 Tag를 서로 비교하여 수신데이터의 무결성을 체크한다.

본 논문의 AES-CCM* 하드웨어는 byte단위로 데이터를 입력받기 때문에 16 cycles 동안 16-byte를 입력받아 하나의 평문 블록을 구성하며 설계한 AES 모듈은 매 라운드 20 cycles이 소요되어 10라운드 동안 총 200 cycles의 동작시간이 필요하다. AES-CCM* 하드웨어의 동작을 시간 변화에 따라 나타내면 그림 5와 같다. 그림 5에서의 T_{in}은 데이터를 입력받는 시간으로 16 cycles을 의미하고 T_1은 AES 모듈의 동작시간인 200 cycles 의미한다. 하나의 평문을 처리하는데 CBC상태의 경우 $T_{in}+T_1$이 소요되며 토글 상태에서는 $T_{in}+2T_1$의 시간이 소요된다.

그림 5를 살펴보면 제안된 AES-CCM* 하드웨어의 응답시간과 암호화시간을 분석할 수 있다. 그림 5의 Response time은 AES-CCM* 하드웨어의 동작이 시작한 시간부터 첫 암호문이 나올 때까지의 시간으로 응답시간을 의미한다. 그림 5의 Encryption time은 하나의 메시지를 암호화하는데 걸리는 시간으로 데이터처리시간을 의미한다. 데이터의 크기에 비례하는 응답시간을 조금 더 쉽게 분석하기 위해 인증 데이터는 32-byte 메시지는 64-byte로 가정하면, 응답시간은 864 cycles, 암호화시간은 416 cycles가 소요된다. 암호화 시간은 128-bit 데이터 블록을 암호화하는데 걸리는 시간으로 수식 (7)에 따라 6MHz 동작 주파수에서 1.8Mbps의 데이터 처리량을 갖는다.

Table 3. Comparisons of synthesis results for the various implementation approaches of AES-CCM*
표 3. AES-CCM*의 구현 방식에 따른 합성 결과의 비교

	[8]	[7]	Proposed AES-CCM*
Technology/ frequency	90nm/ 264MHz	250nm/ 10MHz	65nm/ 6MHz
Bit width (Bits)	128	8	8
AES-CCM* gate count (K)	20.5 (100%)	14.9 (72.6%)	11.8 (57.5%)
Throughput (Mbps)	2690	8.1	1.8

$$\text{처리량} = \frac{\text{데이터 블록}(128\,bits)}{\text{블록 처리 시간}(cycles)} \times \text{동작주파수}(Hz) \quad (7)$$

본 논문에서 제안하는 AES-CCM* IP는 6MHz로 동작하는 IEEE 802.15.4호환 O-QPSK 기반의 ZigBee 모뎀과 연동하기 위해 같은 동작 주파수인 6MHz로 운용되며, 표준에서 요구하는 250kbps의 데이터 처리량을 충분히 지원한다.

AES-CCM* 하드웨어는 Verilog HDL으로 기술하고 Synopsys의 Design Compiler에서 65nm 공정 라이브러리를 사용하여 6MHz 동작주파수로 합성하였다. 표 3은 AES-CCM* IP의 합성결과를 나타내며 토글 구조로 구현된 본 논문의 AES-CCM* IP는 기존의 병렬 구조의 하드웨어에 비해 면적 측면에서 최적화가 이루어졌다. 128-bit AES 모듈을 병렬로 사용한 [8]의 하드웨어의 57%에 해당하는 게이트 수만으로 구현할 수 있다. 또한 Authentication check를 지원하지 않아서 추가적인 프로세서에서의 소프트웨어적인 처리를 요구하는 [7],[9]의 하드웨어와 달리 본 논문에서 제안하는 AES-CCM* IP는 그림4에 나타난 것과 같이 Authentication check를 위한 하드웨어를 포함하고 있기 때문에 프로세서 자원이 제한되는 WPAN 기기에 매우 적합하다.

Ⅴ. 결론

IEEE 802.15.4표준은 250kbps의 데이터 처리량만을

요구하기 때문에, 기존의 고속처리를 위한 128-bit 방식의 AES 연산유닛은 적합하지 않으며, 최소한의 면적으로 요구되는 데이터 처리량을 만족시키는 것이 최우선이다. 이에 본 논문에서는 8-bit 데이터 처리에 기반한 AES 암호화 모듈에 CFA를 적용한 Sbox를 채택함으로써 AES 연산유닛의 복잡도를 줄였다. 또한 AES-CCM*을 토글 구조로 구현함으로써 하나의 AES 암호화 모듈만으로 AES-CCM*을 운용할 수 있도록 설계하였다.

구현된 하드웨어는 IEEE 802.15.4 표준에서 요구하는 처리량을 만족하는 범위 내에서 기존의 하드웨어의 57%에 해당하는 게이트 수만으로 구현되었다. 이는 하드웨어 자원이 매우 제한되어 낮은 복잡도의 하드웨어를 필요로 하는 WPAN 기기에 매우 적합할 뿐만 아니라 IEEE 802.15.4 표준 역시 만족하기 때문에 WPAN시스템의 비용 및 면적 최적화에 기여할 수 있을 것으로 판단한다.

References

[1] FIPS-197:Advanced Encryption Standard, National Institute of Standards and Technology (NIST), 2001

[2] Satoh A., Morioka S., Takano K., Munetoh S., "A Compact Rijndael Hardware Architecture with S-Box Optimization", Theory and Application of Cryptology and Information Security (ASIACRYPT 2001), Gold Coast, Australia, 2001.

[3] C. Paar, "Efficient VLSI architecture for bit-parallel computations in Galois field," Ph.D. dissertation, Institute for Experimental Mathematics, University of Essen, Essen, Germany, 1994.

[4] X. Zhang and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm," IEEE Trans. VLSI Systems, vol. 12, no. 9, pp. 957 - 967, 2004.

[5] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks(LRWPAN), IEEE Std. 802.15.4, 2006.

[6] Ohyoung Song and Jiho Kim, "An Efficient Design of Security Accelerator for IEEE 802.15.4 Wireless Senor Networks",Consumer Communications and Networking Conference (CCNC), Las Vegas, Jan, 2010.

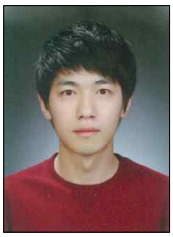
[7] Lian Huai, Xuecheng Zou, Zhenglin Liu, and Yu Han, "An Energy-Efficient AES-CCM Implementation for IEEE802.15.4 Wireless Sensor Networks", Networks Security, Wireless Communications and Trusted Computing, Wuhan, Hubei, 2009.

[8] Dang Khoa Nguyen, Leonardo Lanante and Hiroshi Ochi, "High Throughput - Resource Saving Hardware Implementation of AES-CCM for Robust Security Network", Journal of Automation and Control Engineering Vol. 1, No. 3, September 2013

[9] IP Cores Inc, CCMZ1/CCMZ2 IEEE 802.15.4 CCM AES Cores, www.ipcores.com/images/ccmzcore.pdf, July 2006

BIOGRAPHY

Injun Choi (Student Member)



2014 : BS degree in Electrical Engineering, Chungnam National University.
2014~ : MS degree in Electrical Engineering, Chungnam National University.

Jong-Yeol Lee (Member)

1993 : BS degree in Electronic Engineering, KAIST.
1996 : MS degree in Electronic Engineering, KAIST.
2002 : Ph.D. degree in Electronic Engineering, KAIST.
2002~2004 : Senior Researcher at Hynix Semiconductor
2004~ : Professor in Div. of Electronic Engineering, Chonbuk National University.

Ji-Hoon Kim (Member)

2004 : BS degree in Electrical Engineering, KAIST.
2009 : Ph.D. degree in Electrical Engineering, KAIST.
2009~2010 : Senior Engineer at DMC research center, Samsung Electronics.
2010~ : Associate Professor in Dept. of Electronics Engineering, Chungnam National University.