JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 19, No. 4 : 888~894 Apr. 2015

128비트 경량 블록암호 LEA의 저면적 하드웨어 설계

성미지 · 신경욱*

A Small-area Hardware Design of 128-bit Lightweight Encryption Algorithm LEA

Mi-Ji Sung · Kyung-Wook Shin*

School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Kyungbuk 730-701, Korea

요 약

국가보안기술연구소(NSRI)에서 개발된 경량 블록암호 알고리듬 LEA(Lightweight Encryption Algorithm)의 효율적인 하드웨어 설계에 대해 기술한다. 마스터키 길이 128비트를 지원하도록 설계되었으며, 라운드 변환블록과 키 스케줄러의 암호화 연산과 복호화 연산을 위한 하드웨어 자원이 공유되도록 설계하여 저전력, 저면적 구현을 실현했다. 설계된 LEA 프로세서는 FPGA 구현을 통해 하드웨어 동작을 검증하였다. Xilinx ISE를 이용한 합성결과 LEA 코어는 1,498 슬라이스로 구현되었으며, 135.15 MHz로 동작하여 216.24 Mbps의 성능을 갖는 것으로 평가 되었다.

ABSTRACT

This paper describes an efficient hardware design of Lightweight Encryption Algorithm (LEA) developed by National Security Research Institute(NSRI). The LEA crypto-processor supports for master key of 128-bit. To achieve small-area and low-power implementation, an efficient hardware sharing is employed, which shares hardware resources for encryption and decryption in round transformation block and key scheduler. The designed LEA crypto-processor was verified by FPGA implementation. The LEA core synthesized with Xilinx ISE has 1,498 slice elements, and the estimated throughput is 216.24 Mbps with 135.15 MHz.

키워드 : 경량 블록암호, LEA, IoT 보안, 정보보안, 비밀키 암호

Key word : Lightweight Encryption Algorithm, LEA, IoT Security, Information Security, Secret Key Encryption

Received 23 February 2015, **Revised** 30 March 2015, **Accepted** 06 April 2015

* **Corresponding Author** Kyung-Wook Shin(E-mail:kwshin@kumoh.ac.kr, Tel:+82-54-478-7427)
School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Kyungbuk 730-701, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2015.19.4.888 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

정보통신 기술의 비약적인 발달은 오늘날 우리 주변의 사물들을 네트워크로 연결시켜 주고 이들에 대한 정보를 언제, 어디서나 쉽게 접할 수 있는 사물인터넷(IoT: Internet of Things) 시대의 도래를 촉진하고 있다. IoT 서비스는 스마트기기, 센서 등 다양한 단말 및 이종 네트워크, 애플리케이션 등을 활용하므로, 발생할 수 있는 보안 위협도 많을 것으로 예상된다[1].

오늘날 널리 알려져 있는 정보보안 기술 중 하나가 암호이며, 그리스어에서 기원한 암호(Cryptography)라는 용어는 '비밀기록(secret writing)'을 의미하지만 현재는 '메시지가 공격에 안전하도록 메시지 변형을 하는 예술과 과학'을 뜻한다. 과거에는 암호가 단지 비밀 키를 이용하여 메시지의 암호화와 복호화를 하는 것으로 간주되었지만, 오늘날 암호는 세 개의 별개의 메커니즘, 즉 공개키 암호화, 비밀 키 암호화, 해쉉(Hashing)으로 정의된다. 암호화는 컴퓨터에 저장되어 있거나 네트워크를 통해 전달되는 정보를 제삼자가 가로채어 그 내용을 노출시키거나 의도적으로 내용을 조작・변경하는 등의 보안공격으로부터 정보를 보호하기 위한 수단으로 사용되며, 컴퓨터와 인터넷을 중심으로 한 정보화 사회가 도래함에 따라 그 중요성이 점점 증대되고 있는 핵심기술이다[2,3].

비밀 키 암호방식은 암호화와 복호화에 하나의 비밀 키를 사용한다. 송수신자 양측이 동일한 비밀 키를 가지고 통신을 하므로 대칭키 암호(Symmetric-Key Encipherment)방식이라고도 불리며, 안정성은 공개키 방식에 비해 다소 떨어지지만 구현 방식이 간단하고 동작이 빠르다. 블록 암호는 비밀 키 방식의 하나이며 암호문을 만들기 위해 암호 키와 알고리듬이 데이터 블록 단위로 적용되는 특징이 있다.

미국 NIST(National Institute of Standards and Technology)가 AES(Advanced Encryption Standard) [4]를 블록암호 표준으로 채택한 이래로 강력한 보안성능과 다양한 운영모드의 장점으로 인해 무선 랜, Zigbee 등 다양한 유・무선 통신 시스템의 보안 알고리듬으로 폭 넓게 사용되고 있다.

최근, IoT 기술이 상용화되기 시작함에 따라 정보유통의 물리적인 안전성이 중요 이슈로 부각되고 있으며, 전용 하드웨어를 이용한 보안 시스템의 구현에 관한 연구가 활발히 이루어지고 있다. 대용량 데이터의 고속 암・복호에 초점을 맞춘 하드웨어 구현을 비롯해서 스마트카드, NFC, IoT와 같은 휴대장치에 적합한 소면적, 저전력 하드웨어 구현 결과들이 발표되고 있다[5-9]. 고속 처리에 초점을 맞춘 하드웨어 구조는 높은 처리율을 갖는 장점이 있지만, 회로 복잡도나 전력소모 측면에서 휴대용 기기에 적합하지 않다.

본 논문에서는 국가보안기술연구소(NSRI)에서 개발한 128비트 블록암호 알고리듬 LEA[10]를 IoT 환경에 적합하도록 최적화한 LEA 암・복호 코어를 설계하였으며, FPGA 구현과 UART 통신, MFC를 이용한 데모 프로그램을 통해 하드웨어 동작을 검증하였다.

본 논문은 다음과 같이 구성된다. Ⅱ장에서는 블록암호 알고리듬 LEA에 대해 간략히 설명하고, Ⅲ장에서는 LEA-128 암호・복호 프로세서 설계에 대해 설명한다. 설계된 회로의 기능 검증 및 FPGA 구현에 대해 Ⅳ장에서 기술하며, Ⅴ장에서 결론을 맺는다.

Ⅱ. 경량 블록암호 알고리듬 LEA

블록암호 LEA는 128비트의 평문(암호문) 블록을 128/192/256비트의 마스터키로 암호화(복호화)하여 128비트의 암호문(평문)을 생성하는 대칭키 방식의 암호 알고리듬이다. LEA는 ARX(Addition, Rotation, XOR) 연산을 기반으로 한 Feistel 유사 구조이며, 마스터키의 길이에 따라 24/28/32 라운드의 연산을 통해 암호화(복호화)가 이루어진다[2]. 128비트의 마스터키로부터 생성되는 192비트의 라운드 키가 라운드 변환에 사용된다. 라운드 함수는 32비트 단위의 ARX 연산으로 구성되어, 이들 연산을 지원하는 범용 32비트 소프트웨어 플랫폼에서 고속으로 동작한다. 또한 라운드 함수 내부의 ARX 연산 배치는 충분한 안전성을 보장함과 동시에 S-box의 사용을 배제하여 경량 구현이 가능하도록 한다[3]. LEA의 암호화・복호화 과정은 그림 1과 같으며, 라운드 키를 생성하는 키 스케줄러와 암호화・복호화를 수행하는 라운드 함수로 구성된다. 암호화 와 복호화 과정은 서로 역순으로 이루어지며, 복호화는 암호화 과정의 역연산 과정이다. 따라서, 암호화 과정에서의 모듈로 가산은 복호화 과정에서 모듈로 감산으로 구현되고, 순환이동도 반대방향으로 이루어진다.

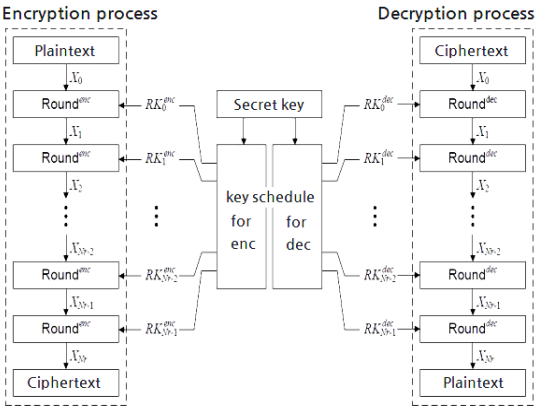


그림 1. LEA 블록암호 알고리듬
Fig. 1 LEA block cipher algorithm

암호화 과정은 비밀키 K로부터 Nr개의 192비트 암호화 라운드 키 RK_i^{enc} (단, $0 \le i \le Nr-1$)를 생성하는 암호화 키 스케줄 함수 $KeySchedule_i^{enc}$와 라운드 키 RK_i^{enc} 와 라운드 함수 $Round^{enc}$를 이용하여 128비트 평문 P를 128비트 암호문 C로 변환하는 암호화 함수 $Encrypt$로 구성된다. 암호화 함수의 슈도 코드는 그림 2(a)와 같다. 복호화 과정은 비밀키 K로부터 Nr개의 192비트 복호화 라운드 키 RK_i^{dec} (단, $0 \le i \le Nr-1$)를 생성하는 키 스케줄 함수 $keySchedule_i^{dec}$와 라운드 키 RK_i^{dec}와 라운드 함수 $Round^{dec}$를 이용하여 128비트 암호문 C를 128비트 평문 P로 변환하는 복호화 함수 $Decrypt$로 구성된다. 복호화 함수의 슈도 코드는 그림 2(b)와 같다.

암호화 과정의 i (단, $0 \le i \le Nr-1$) 번째 라운드연산은 그림 3(a)와 같이 구성된다. 128비트 상태변수 $X_i = (X_i[0], X_i[1], X_i[2], X_i[3])$와 192비트 라운드 키 $RK_i^{enc} = (RK_i^{enc}[0], RK_i^{enc}[1], \cdots, RK_i^{enc}[5])$의 가산, 32비트 modulo 가산, 비트순환이동(ROL_9, ROR_5, ROR_3), 32비트 워드 순환이동을 통해 128비트의 상태변수 $X_{i+1} = (X_{i+1}[0], X_{i+1}[1], X_{i+1}[2], X_{i+1}[3])$가 생

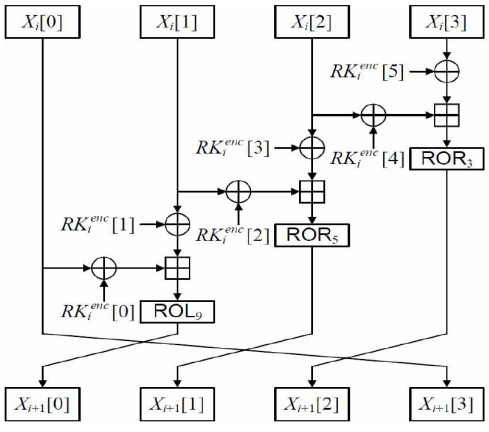
Encryption function : $C \leftarrow \text{Encrypt}(P, RK_0^{enc}, RK_1^{enc}, \cdots, RK_{Nr-1}^{enc})$
Input : 128-bit plaintext P, 192-bit round key $RK_0^{enc}, RK_1^{enc}, \cdots, RK_{Nr-1}^{enc}$
Output : 128-bit ciphertext C
1: $X_0 \leftarrow P$
2: for i = 0 to Nr-1 do
3: $X_{i+1} \leftarrow \text{Round}^{enc}(X_i, RK_i^{enc})$
4: end for
5: $C \leftarrow X_{Nr}$

(a)

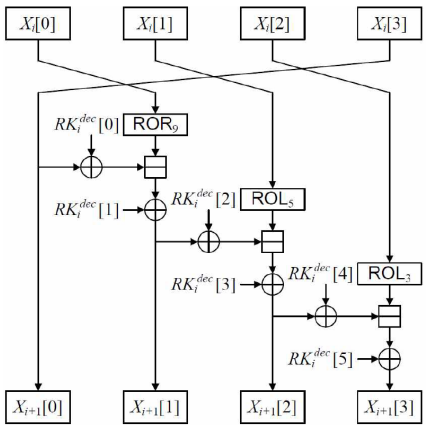
Decryption function : $P \leftarrow \text{Decrypt}(C, RK_0^{dec}, RK_1^{dec}, \cdots, RK_{Nr-1}^{dec})$
Input : 128-bit ciphertext C, 192-bit round key $RK_0^{dec}, RK_1^{dec}, \cdots, RK_{Nr-1}^{dec}$
Output : 128-bit plaintext P
1: $X_0 \leftarrow C$
2: for i = 0 to Nr-1 do
3: $X_{i+1} \leftarrow \text{Round}^{dec}(X_i, RK_i^{dec})$
4: end for
5: $P \leftarrow X_{Nr}$

(b)

그림 2. LEA (a) 암호화 (b) 복호화의 슈도 코드
Fig. 2 Pseudo code for (a) encryption (b) decryption of LEA



(a)



(b)

그림 3. LEA (a) 암호화 (b) 복호화의 라운드 함수
Fig. 3 Round function for (a) encryption (b) decryption of LEA

성된다.

복호화 과정의 i (단, $0 \le i \le Nr-1$) 번째 라운드연산은 그림 3(b)와 같이 구성된다. 128비트 상태변수 $X_i = (X_i[0], X_i[1], X_i[2], X_i[3])$와 192비트 라운드 키 $RK_i^{dec} = (RK_i^{dec}[0], RK_i^{dec}[1], \cdots, RK_i^{dec}[5])$의 가산, 32비트 modulo 가산, 비트순환이동(ROR_9, ROL_5, ROL_3), 32비트 워드 순환이동을 통해 128비트의 상태변수 $X_{i+1} = (X_{i+1}[0], X_{i+1}[1], X_{i+1}[2], X_{i+1}[3])$가 생성된다.

LEA의 키 스케줄링은 AR(Addition-Rotation) 연산과 32비트 modulo 연산에 사용되는 상수를 생성하는 상수생성부로 구성된다. 입력되는 비밀키의 길이에 따라 상수 생성부에서 생성되는 상수의 개수가 달라지며, 암호화 과정인지 복호화 과정인지에 따라 AR 연산의 순서가 반대로 적용된다. 암호화 키 스케줄링은 비밀키 K로부터 암호화에 필요한 Nr개의 192비트 암호화 라운드 키 RK_i^{enc} (단, $0 \le i \le Nr-1$)를 생성하는 과정이며, 입력된 비밀키와 상수를 32비트 modulo 가산한 후, 비트 순환이동을 통해 이루어진다. 복호화 라운드 키는 $RK_i^{dec} = RK_{Nr-i-1}^{enc}$ 인 관계를 제외하면 암호화 라운드 키와 동일한 방법으로 생성된다.

Ⅲ. LEA-128 암호 · 복호 회로 설계

본 논문에서는 128비트의 평문(암호문) 블록을 128비트의 마스터키로 암호화(복호화)하여 128비트의 암호문(평문)을 생성하는 LEA-128 암호 · 복호 프로세서를 설계하였다. 설계된 LEA-128 코어의 전체 구조는 그림 4와 같으며, 라운드 블록, 키 스케줄링 블록, 제어 블록으로 구성된다. 라운드 블록은 24번의 라운드 변환을 통해 암호 · 복호 연산을 수행하며, 각 라운드의 연산은 3클록 주기로 처리된다. 키 스케줄링 블록은 각 라운드 연산에 사용되는 192비트의 라운드 키를 on-the-fly 방식으로 생성한다. 저면적 구현을 위해 라운드 블록을 32비트 회로로 설계하였으며, 입력 핀 (data_in)을 공유하여 마스터키와 평문(암호문)이 시분할 방식으로 입력되도록 하였다. 또한, 암호화와 복호화에서 하드웨어 자원이 공유되도록 설계하였다.

3.1. 라운드 블록

라운드 블록은 128비트의 평문(암호문) 입력과 키 스케줄러에 의해 생성되는 192비트의 라운드 키를 받아 라운드 연산을 반복적으로 처리하여 암호(복호)연산을 수행한다. 라운드 블록은 그림 5와 같이 구현되었으며, 4개의 32비트 내부 레지스터($X0 \sim X3$), XOR 연산기, 32비트 modulo 가산기, 비트 순환이동(ROL, ROR) 등으로 구성된다.

Text_in 포트를 통해 입력되는 128비트의 평문/암호문은 32비트 단위로 4클록 주기에 걸쳐 상태변수 레지스터 $X0$, $X1$, $X2$, $X3$에 저장된다. 키 스케줄링 블록에서 생성된 라운드 키 rk0와 레지스터 $X2$ 값이 XOR 연산되고, 라운드 키 rk1과 레지스터 $X3$ 값이 XOR 연산된다. XOR 연산의 두 결과 값이 32비트 modulo 가산되고, modulo 가산 결과는 라운드 연산의 클록 순서에 따

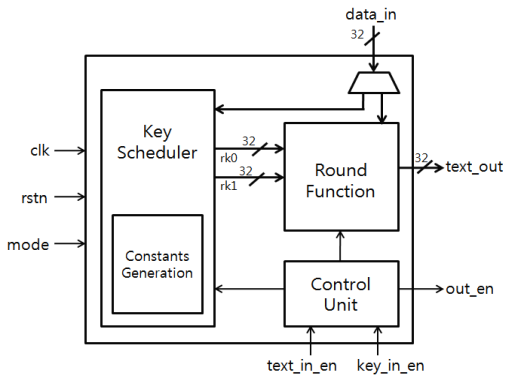


그림 4. LEA-128 암호/복호 프로세서
Fig. 4 LEA-128 encryption/decryption processor

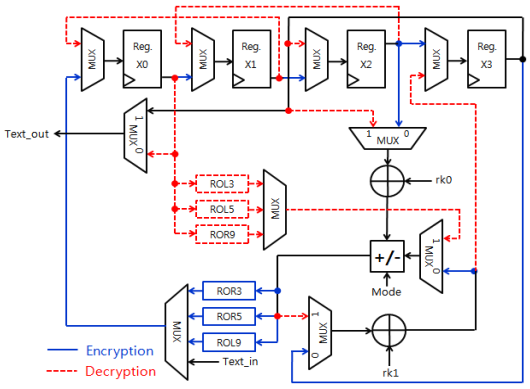


그림 5. 라운드 블록
Fig. 5 Round block

라 첫 번째 클록에서는 오른쪽으로 3비트 순환이동(ROR_3) 되고, 두 번째 클록에서는 오른쪽으로 5비트 순환이동(ROR_5) 되며, 세 번째 클록에서는 왼쪽으로 9비트 순환이동(ROL_9) 된다. 순환 이동된 결과 값은 다시 레지스터 $X0$에 저장되며, 동시에 각 레지스터에 저장되어 있던 값은 $X0 \rightarrow X1 \rightarrow X2 \rightarrow X3$으로 이동된다. 이와 같이 한 라운드의 연산에는 세 클록 주기가 소요된다. 복호화 모드에서는 암호화 모드와 반대의 순서로 라운드 키가 사용된다. 암호화 과정의 XOR→Addition→Rotation의 연산 순서는 복호화 과정에서는 Rotation→Subtraction→XOR의 순서로 적용된다. 비트 순환이동도 반대 방향으로 이루어지며, 32비트 modulo 가산은 감산으로 대체된다. 또한, 내부 레지스터의 데이터 이동방향도 $X3 \rightarrow X2 \rightarrow X1 \rightarrow X0$로 암호화 모드와 반대 방향이 된다. 암호화 모드와 동일하게 한 라운드 연산은 세 클록 주기 동안 이루어진다.

3.2. 키 스케줄링 블록

LEA의 암호화·복호화의 라운드 연산에 사용되는 192비트의 라운드 키는 키 스케줄링 알고리듬에 의해 생성된다. 설계된 키 스케줄링 블록은 그림 6과 같으며, 6개의 32비트 내부 레지스터($T0 \sim T5$), XOR 연산기, 32비트 modulo 가산기, 비트 순환이동 시프트 연산기(ROL, ROR) 등으로 구성되며, 키 생성에 사용되는 상수 δ를 생성하는 상수생성기를 포함한다. 상수값 생성기는 그림 7과 같이 구현하였다.

128비트의 마스터키 $K0, K1, K2, K3$는 32비트 단위로 4클록 주기 동안 레지스터 $T0$와 $T1$에 교대로 입력되며, $T0$에 입력될 때 32비트씩 $T0 \rightarrow T2 \rightarrow T4$으로 시프트 되고, 마찬가지로 $T1$에 입력될 때 32비트씩 $T1 \rightarrow T3 \rightarrow T5$으로 시프트 되어 T 레지스터에 저장된다. 128비트의 마스터키를 저장하기 위해 32비트 레지스터 4개가 필요하지만, 그림 6과 같이 $T0, T2, T4$와 $T1, T3, T5$를 병렬구조로 배치하여 32비트 레지스터 6개($T0 \sim T5$)를 사용하면, 라운드 키 생성을 3클록 주기로 구현할 수 있다. 이와 같은 방법은 종래의 문헌에 발표된 방법(32비트 레지스터 4개를 직렬로 배치한 구조)의 4클록 주기에 비해 1클록 주기를 감소시킬 수 있다. 복호화 과정의 첫 번째 라운드에서 사용되는 라운드 키는 암호화 과정의 마지막 라운드 키와 동일하므로, 암호화 과정과 달리 초기에 키를 생성하지 않아도 된다. $(K0, K1), (K2, K1), (K3, K1)$로 짝을 이루어 순차적으로 연산이 이루어져야하므로, $T5 \rightarrow T3 \rightarrow T1$으로 시프트 시키면서 라운드 키를 생성하면 초기의 레지스터 $T1$ 값이 유지되지 못하고 변하게 된다. 이와 같은 문제를 해결하기 위해 복호화의 첫 라운드에서 $T0 \rightarrow T4$, $T1 \rightarrow T5$로 시프트시킴으로써 $T1$의 값이 보존되도록 하였다.

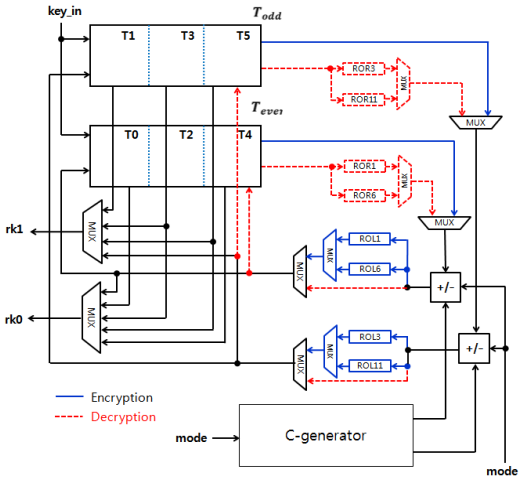


그림 6. 키 스케줄링 블록
Fig. 6 Key scheduling block

Ⅳ. 기능검증 및 FPGA 구현

Verilog HDL로 설계된 LEA-128 코어의 기능검증 결과는 그림 8과 같으며, 128비트의 평문 "10111213 14151617 18191a1b 1c1d1e1f"와 128비트의 마스터키 "0f1e2d3c 4b5a6978 8796a5b4 c3d2e1f0"를 입력벡터로 사용한 시뮬레이션 결과를 보이고 있다. 암호화의 결과로 128비트 암호문 "9fc84e35 28c6c618 5532c7a7 04648bfd"이 출력되었고, 이를 다시 복호한 결과는 암호에서 입력으로 사용된 평문 "10111213 14151617

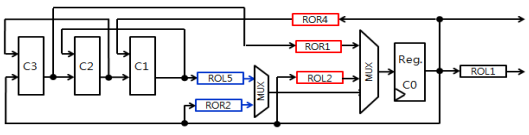


그림 7. 상수값 δ 생성 블록
Fig. 7 Constant δ generation block

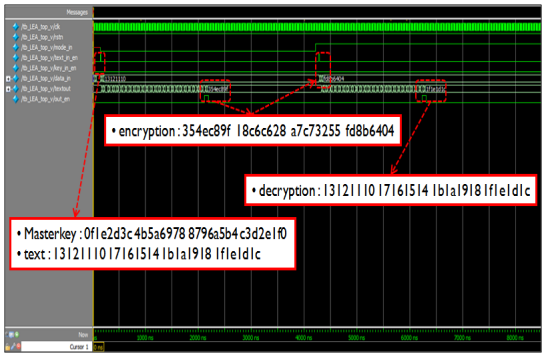
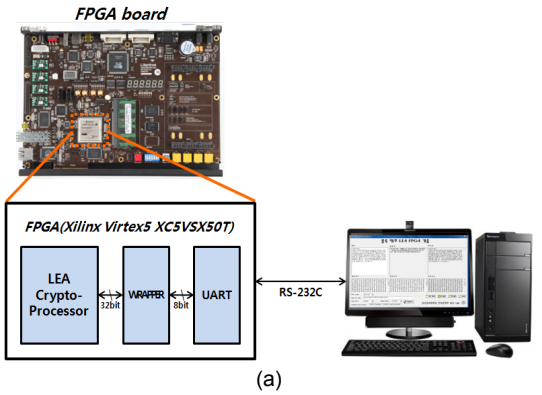
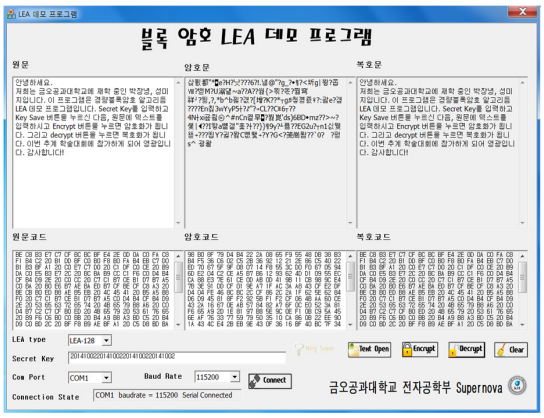


그림 8. LEA-128 코어의 기능검증 결과
Fig. 8 Simulation result of LEA-128 crypto-core



(a)



(b)

그림 9. LEA-128 코어의 FPGA 검증 결과 (a) FPGA 검증 시스템 (b) FPGA 검증 결과
Fig. 9 FPGA verification result of LEA-128 core (a) FPGA verification system (b) FPGA verification result

18191a1b 1c1d1e1f"이 출력됨을 확인함으로써 설계된 LEA-128 프로세서의 논리기능이 정상적으로 동작함을 확인하였다.

기능검증이 완료된 LEA-128 코어는 FPGA 구현을 통해 하드웨어 동작을 검증하였다. 그림 9(a)는 FPGA 보드, UART 인터페이스, 구동 소프트웨어로 구성된 검증시스템 구성되며, Xilinx Virtex5 XC5VSX-50T 디바이스가 사용되었다. PC에서 입력된 비밀키와 평문(암호문) 데이터가 RS232C 통신을 통해 FPGA에 인가되고, FPGA에서 출력되는 암호문(평문) 데이터가 화면에 표시된다. 그림 9(b)는 FPGA 검증 결과를 보이고 있다. 평문을 암호화하고, 암호문을 복호화하여 원래의 평문과 일치하는 복호결과가 출력되어 FPGA에 구현된 LEA-128 프로세서가 올바로 동작함을 확인할 수 있다. 설계된 LEA-128 프로세서는 암호/복호과정이 연속적으로 수행되도록 고려되었으며, FPGA 합성 결과 레지스터 600슬라이스, LUT 1418슬라이스, 총1498슬라이스로 구현되었고, 135.15MHz로 동작하여 216.24Mbps의 성능을 갖는 것으로 평가 되었다.

V. 결 론

본 논문에서는 한국정보통신기술협회(TTA) 표준으로 등록된 128비트 블록암호 알고리듬 LEA-128을 하드웨어로 구현하여 동작을 확인하였다. 저면적과 저전력 구현을 위해 암호화와 복호화 과정에서 하드웨어 자원이 공유되도록 설계하였다. 설계된 LEA-128 암호·복호 프로세서는 IoT 및 모바일 기기 보안 등과 같이 저전력, 경량화가 요구되는 응용분야의 정보보호 코어로 활용이 가능할 것으로 예상된다.

감사의 글

- 산업통상자원부 출연금으로 수행한 산업핵심기술개발사업(10049009, 사물인터넷 기반 영상보안용 초저전력 SoC 핵심 IP 기술 개발)의 지원을 받았음.
- 반도체설계교육센터(IDEC)의 CAD Tool 지원에 감사드립니다.

REFERENCES

[1] Dong-hui Kim et al, "Security for IoT Service", *Journal of Korea Institute of Communication and Information Services*, vol. 30, no. 8, pp.53, July 2013.

[2] Chang-seop Park, "Cryptography and Security", Daeyoung Press, 1999.

[3] W. Stalling, Cryptography and Network Security, Prentice Hall, 1999.

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

[5] Dong-hyeon Kim and Kyung-wook Shin, "An Efficient Hardware Implementation of ARIA Block Cipher Algorithm Supporting Four Modes of Operation and Three Master key Lengths", *Journal of Korea Institute of Information and Communication Engineering*, vol. 16, no. 1, pp. 2517-2524, 2012. 11.

[6] Hae-won Park and Kyung-wook Shin, "An efficient hardware implementation of 64-bit block cipher algorithm HIGHT", *Journal of Korea Institute of Information and Communication Engineering*, vol. 15, no. 호, pp. 1993-1999, 2011. 9.

[7] Donggeon Lee et al, "Efficient Hardware Implementation of the Lightweight Block Encryption Algorithm LEA", *Sensors*, pp. 982-983, 2014.

[8] T. Akishita and H. Hiwatari, "Very Compact Hardware Implementations of the Blockcipher CLEFIA", in *Selected Areas in Cryptography –SAC 2011, ser. LNCS*, vol. 7118, pp. 278-292, Springer-Verlag, 2012.

[9] E. B. Kavun and T. Yalcin, "RAM-Based Ultra-Lightweight FPGA Implementation of PRESENT," in *International Conference on Reconfigurable Computing and FPGAs – ReConFig 2011, IEEE Computer Society*, pp. 280-285, 2011.

[10] Telecommunications Technology Association, "128-Bit Block Cipher LEA", TTA Standard, TTAK.KO-12.0223, 2013.

성미지(Mi-Ji Sung)

2015년 2월 금오공과대학교 전자공학부(공학사)
※관심분야 : 통신 및 신호처리용 반도체 IP 설계, 정보보호용 반도체 IP 설계

신경욱(Kyung-Wook Shin)

1984년 2월 한국항공대학교 전자공학과(공학사)
1986년 2월 연세대학교대학원 전자공학과(공학석사)
1990년 8월 연세대학교대학원(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)
2013년 2월~2014년 2월 Georgia Institute of Technology(방문교수)
※관심분야 : 통신 및 신호처리용 SoC 설계, 정보보호 SoC 설계, 반도체 IP 설계