경량 블록암호 LEA용 암호/복호 프로세서 설계

성미지* · 신경욱*

*금오공과대학교

A Design of Crypto-processor for Lightweight Block Cipher LEA

Mi-ji Sung* · Kyung-wook Shin*

*Kumoh National Institute of Technology

E-mail : smj920307@kumoh.ac.kr

요 약

128비트 블록암호 알고리듬 LEA(Lightweight Encryption Algorithm)의 효율적인 하드웨어 설계에 대해 기술한다. 저전력, 저면적 구현을 위해 라운드블록과 키 스케줄러의 암호화와 복호화 연산의 하드웨어 자원이 공유되도록 설계하였다. 키 스케줄러 레지스터의 구조를 개선하여 키 스케줄링에 소요되는 클록 사이클 수를 감소시켰으며, 이를 통해 암호화/복호화 성능을 향상시켰다. 설계된 LEA 프로세서는 FPGA 합성결과, 2,364 슬라이스로 구현되었으며, 113 MHz로 동작하여 128/192/256비트 마스터키 길이에 대해 각각 181/162/109 Mbps의 성능을 갖는 것으로 평가되었다.

ABSTRACT

This paper describes an efficient hardware design of 128-bit block cipher algorithm LEA(lightweight encryption algorithm). In order to achieve area-efficient and low-power implementation, round block and key scheduler block are optimized to share hardware resources for encryption and decryption. The key scheduler register is modified to reduce clock cycles required for key scheduling, which results in improved encryption/decryption performance. FPGA synthesis results of the LEA processor show that it has 2,364 slices, and the estimated performance for the master key of 128/192/256-bit at 113 MHz clock frequency is about 181/162/109 Mbps, respectively.

키워드

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

Ⅰ. 서 론

다양한 기기들이 인터넷에 연결되어 인간의 조작이나 도움 없이 서로 정보를 교환하고 공유하는 사물인터넷(IoT: Internet of Things) 기술이 빠른 속도로 실용화되고 있으며 스마트 홈, 스마트 보안 시스템 등 다양한 분야에 폭 넓게 적용되고 있다. IoT는 무선 네트워크를 통해 데이터를 전송하므로 해커나 악의적인 소프트웨어, 바이러스 등에 노출될 수 있으며, 정보보안 기술의 적용이 필수적이다.[1] IoT 보안에 적합하도록 개발된 경량(lightweight) 블록암호 알고리듬들이 제안되고 있으며, 본 논문에서는 우리나라 국가 표준으로 채택된 128비트 블록암호 알고리듬 LEA[2]를 IoT 환경에 적합하도록 최적화한 LEA 암호/복호 프로세서를 설계하였다.

Ⅱ. LEA 블록암호 알고리듬

블록암호 LEA는 128비트의 평문/암호문 블록을 128/192/256비트의 마스터키로 암호화/복호화하여 128비트의 암호문/평문을 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/복호화가 이루어진다. 128/192/256비트의 마스터키로부터 생성되는 192비트의 라운드 키가 라운드 변환에 사용된다. 라운드 함수는 32비트 단위의 ARX 연산으로 구성되어 충분한 안전성을 보장함과 동시에 S-box의 사용을 배제하여 경량 구현이 가능하다.[3]

LEA 알고리듬의 전체 구조는 그림 1과 같으며, 라운드키를 생성하는 키 스케줄러와 생성된

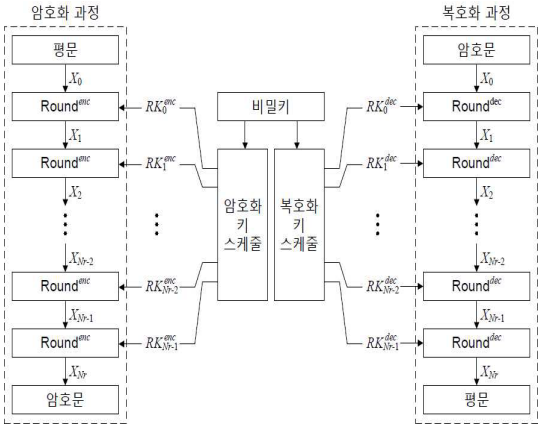

그림 1. LEA 블록암호
Fig. 1. LEA block cipher

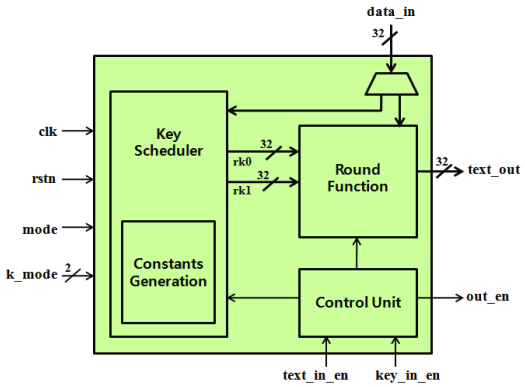

그림 2. LEA 암호/복호 프로세서
Fig. 2. LEA encryption/decryption processor

라운드 키를 사용하여 암호화/복호화 과정을 수행하는 라운드 함수로 구성된다. 암호화 과정과 복호화 과정은 서로 역 연산으로 이루어진다.

III. LEA 암호/복호 회로 설계

본 논문에서는 128비트의 평문/암호문 블록을 128/192/256비트의 마스터키로 암호화/복호화하여 128비트의 암호문/평문을 생성하는 LEA 암호/복호 프로세서를 설계하였다. 설계된 LEA 코어의 전체 구조는 그림 2와 같으며 라운드 블록, 키 스케줄링 블록, 제어 블록으로 구성된다. 라운드 블록은 마스터키 길이 128/192/256비트에 따라 24/28/32번의 라운드 변환을 통해 암호/복호 연산을 수행하며, 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/암호문이 시분할 방식으로 입력되도록 하였다. 또한, 마스터키의 길이에 따른 3가지 모드와 암호화/복호화 연산의 하드웨어 자원이 공유되도록 설계하였다.

3.1 라운드 블록

라운드 블록은 128비트의 평문/암호문과 키 스케줄러에 의해 생성되는 192비트의 라운드 키를 입력받아 라운드 연산을 반복적으로 처리하여 암호/복호 연산을 수행한다.

Text_in 포트를 통해 입력되는 128비트의 평문/암호문은 32비트 단위로 4클록 주기에 걸쳐 상태변수 레지스터에 저장된다. 128비트의 평문/암호문 입력이 완료된 후, mode 신호에 따라 mode=0이면 암호화 라운드 연산이 수행되고, mode=1이면 복호화 라운드 연산이 수행된다. 라운드 블록의 내부 레지스터, XOR, 32비트 모듈로(modulo) 가산/감산 등이 암호화 동작과 복호화 동작에 공유되어 하드웨어 자원이 최소화되도록 하였다. 라운드 연산은 마스터키의 길이 128/192/256비트에 따라 각각 3/3/4 클록 주기가 소요되며, 라운드 연산이 24/28/32회 반복된 후, 암호문/복호문이 출력된다.

3.2 키 스케줄러 블록

암호/복호 라운드 연산에 사용되는 192비트의 라운드 키는 키 스케줄링 알고리듬에 의해 생성된다. 본 논문에서 설계된 키 스케줄링 블록은 마스터키의 길이(128/192/ 256비트)와 암호화/복호화 동작모드에 따라 회로의 동작이 달라진다. mode 신호에 따라 암호화 키 스케줄링 또는 복호화 키 스케줄링이 이루어지고, 마스터키의 길이를 지정하는 2비트의 신호 k_mode에 따라 멀티플렉서의 동작이 제어되어 LEA-128, LEA-192, LEA-256 모드의 키 스케줄링이 선택적으로 수행된다. 키 레지스터($T0 \sim T7$), XOR 연산기, 32비트 모듈로 가산기/감산기 등이 암호화/복호화와 마스터키 길이에 따른 모드에서 공유되도록 하여 하드웨어 자원이 최소화되도록 하였다.

32비트 키 레지스터 8개($T0 \sim T7$)를 짝수 인덱스 레지스터 $T0, T2, T4, T6$와 홀수 인덱스 레지스터 $T1, T3, T5, T7$로 분할하여 구성하고 이들을 병렬로 동작시키는 방법을 적용하였다. 마스터키는 32비트 단위로 레지스터 $T0$와 $T1$에 교대로 입력되며, 32비트 단위로 시프트 되어 레지스터에 저장된다.

LEA-128 모드는 32비트 레지스터 6개를 사용하여 생성된 키를 저장하고 멀티플렉서를 통해 출력하는 방법을 적용하였으며, 라운드 키 생성에 라운드 당 3클록 주기가 소요되어 암호 연산과 복호 연산에 각각 20%, 23%의 속도 개선이 얻어진다. LEA-256 모드의 동작에서는 해당 라운드의 키 스케줄링에 사용되지 않는 중간 변환 값이 레지스터 $T6, T7$에 저장되도록 함으로써 라운드 키 생성에 라운드 당 4클록 주기(키 스케줄링을 위한 3클록 주기와 키 시프트/저장을 위한 1클록 주기)가 소요되어 암호 연산과 복호연산에 각각 33%, 32%의 속도 개선이 얻어진다.

Ⅳ. 기능검증 및 FPGA 구현

Verilog HDL로 설계된 LEA 코어의 기능검증 결과는 그림 3과 같으며, 128비트의 평문 "3031 3233 34353637 38393a3b 3c3d3e3f"와 256비트의 마스터키 "0f1e2d3c 4b5a6978 8796a5b4 c3d2e1f0 f0e1d2c3 b4a59687 78695a4b 3c2d1e0f"를 사용한 시뮬레이션 결과를 보이고 있다. 암호화 결과로 암호문 "f6af51d6 c189b147 ca00893a 97e1f927"이 출력되었고, 이를 다시 복호화하여 평문 "3031 3233 34353637 38393a3b 3c3d3e3f"이 출력되었으며, 설계된 LEA 프로세서의 논리기능이 정상 동작함을 확인하였다.

기능검증이 완료된 LEA 코어는 FPGA 구현을 통해 하드웨어 동작을 검증하였다. 검증 플랫폼은 FPGA 보드, UART 인터페이스, 구동 소프트웨어 등으로 구성되며, FPGA 디바이스는 Xilinx Virtex5 XC5VSX-50T가 사용되었다. PC에서 입력된 비밀키와 평문/암호문 데이터가 RS232C 통신을 통해 FPGA로 입력되고, FPGA에서 출력되는 암호문/평문 데이터가 표시된다. 그림 4는 FPGA 검증 플랫폼을 이용한 검증 결과를 보이고 있다. 평문을 암호화하고, 암호문을 복호화하여 원래의 평문과 일치하는 복호결과가 출력되어 FPGA에 구현된 LEA 프로세서가 올바르게 동작함을 확인하였다.

설계된 LEA 프로세서는 FPGA 합성 결과 2,364 슬라이스로 구현되었다. 113 MHz 클록 주파수로 동작할 때, 128/192/256비트의 3가지 마스터키 길이에 따른 동작모드에서 암호화 동작 시 각각 181/162/109 Mbps, 복호화 동작 시 각각 188/162/109 Mbps의 성능을 갖는 것으로 평가되었다.

Ⅴ. 결 론

본 논문에서는 한국정보통신기술협회(TTA) 표준으로 등록된 128비트 블록암호 알고리듬 LEA를 하드웨어로 구현하여 동작을 확인하였다. 마스터키 길이에 따른 3가지 동작모드와 암호/복호 연산의 하드웨어 자원이 공유되도록 최적화하여 설계하였다. 키 스케줄러의 구조와 동작방식을 개선하여 암호/복호 동작속도를 향상시켰다. 설계된 LEA 프로세서는 Xilinx Vertex5 FPGA에서 최대 113 MHz 클록으로 동작 가능하며, 마스터키 길이 128/192/256비트 모드에서 각각 181/162/109 Mbps의 성능을 갖는 것으로 평가 되었다. 설계된 LEA 암호/복호 프로세서는 IoT 및 모바일 기기 보안 등과 같이 저전력과 경량화가 요구되는 응용분야의 보안 IP로 활용이 가능하다.

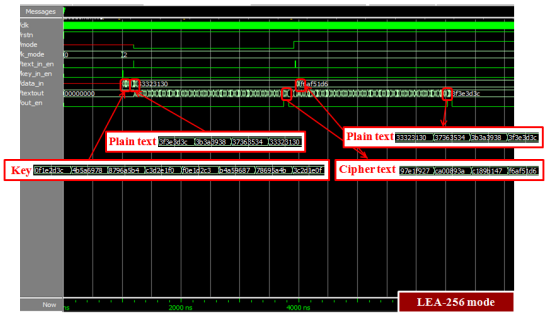

그림 3. LEA 코어의 기능검증 결과
Fig. 3. Simulation result of LEA crypto-core

그림 4. LEA 코어의 FPGA 검증 결과
Fig. 4. FPGA verification result of LEA crypto-core

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.

참고문헌

[1] 김동희 외 2, "IoT 서비스를 위한 보안", 한국통신학회지, 제30권 제8호, pp.53, 7월 2013년
[2] "Block Cipher LEA Validation System", pp.4
[3] 한국정보통신기술협회, "128비트 블록 암호 LEA", TTA Standard, TTAK.KO-12.0223, 12월 2013년
[4] Donggeon Lee et al, "Efficient Hardware Implementation of the Lightweight Block Encryption Algorithm LEA", Sensors, pp. 982-983, 2014.