블록암호 알고리듬 LEA의 효율적인 하드웨어 구현

성미지* · 박장녕* · 신경욱*

*금오공과대학교

An Efficient Hardware Implementation of Block Cipher Algorithm LEA

Mi-ji Sung* · Jang-nyeong Park* · Kyung-wook Shin*

*Kumoh National Institute of Technology

E-mail : smj920307@kumoh.ac.kr

요 약

LEA(Lightweight Encryption Algorithm)는 2012년 국가보안기술연구소(NSRI)에서 개발한 128비트 고속·경량 블록암호 알고리듬이다. LEA는 128/192/256비트 마스터키를 사용하여 128비트 평문을 128비트 암호문으로, 또는 그 역으로 변환한다. 라운드 변환블록의 암호화 연산과 복호화 연산의 하드웨어 자원이 공유되도록 설계하였으며, 또한 키 스케줄러도 암호화와 복호화의 하드웨어 자원이 공유되도록 설계하여 저전력, 저면적 구현을 실현했다. 설계된 LEA 프로세서는 FPGA 구현을 통해 하드웨어 동작을 검증하였다.

ABSTRACT

The LEA(Lightweight Encryption Algorithm) is a 128-bit high-speed/lightweight block cipher algorithm developed by National Security Research Institute(NSRI) in 2012. The LEA encrypts plain text of 128-bit using cipher key of 128/192/256-bit, and produces cipher text of 128-bit, and vice versa. To reduce hardware complexity, we propose an efficient architecture which shares hardware resources for encryption and decryption in round transformation block. Hardware sharing technique for key scheduler was also devised to achieve area-efficient and low-power implementation. The designed LEA cryptographic processor was verified by using FPGA implementation.

키워드

LEA, block cipher, cryptographic processor, information security

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] 본 논문에서는 국가보안기술연구소(NSRI)에서 개발한 128비트 블록암호 알고리듬 LEA[2]를 IoT 환경에 적합하도록 최적화한 LEA 암·복호 코어를 설계하였으며, FPGA 구현과 UART 통신, MFC를 이용한 데모 프로그램을 통해 하드웨어 동작을 검증하였다.

Ⅱ. LEA 블록암호 알고리듬

LEA는 128비트 크기의 평문(암호문) 블록을 128/192/256비트의 마스터키로 암호화(복호화)하여 128비트의 암호문(평문)을 생성하는 대칭키 방식의 블록암호 알고리듬이다. 전체 구조는 대부분의 CPU에서 효율적으로 지원되는 ARX(Addition, Rotation, XOR) 연산을 기반으로 한 Type-3 Feistel 유사 구조로, 비밀키의 길이에 따라 24/28/32 라운드의 연산을 통해 암호화(복호화)가 이루어지며[2], 128 비트의 마스터키로부터 생성되는 192비트의 라운드키가 라운드 변환에 사용된다. 라운드 함수는 32비트 단위의 ARX 연산만으로 구성되어, 이들 연산을 지원하는 범용 32비트 소프트웨어 플랫폼에서 고속으로 동작한다. 또한

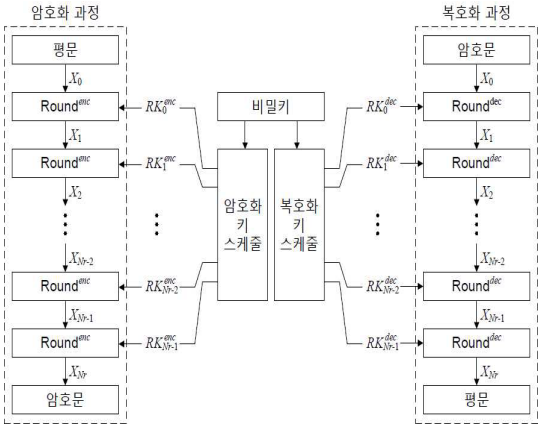

그림 1. LEA 블록암호 알고리듬
Fig. 1. LEA block cipher algorithm

라운드 함수 내부의 ARX 연산 배치는 충분한 안전성을 보장함과 동시에 S-box의 사용을 배제하여 경량 구현이 가능하도록 한다. [3]

LEA 알고리듬의 전체 구조는 그림 1과 같으며, 라운드키를 생성하는 키 스케줄러와 생성된 라운드키를 사용하여 암·복호화 과정을 수행하는 라운드 함수로 구성된다. 암호화 과정과 복호화 과정은 서로 역순으로 이루어지며, 암호화 과정에서의 모듈로 가산은 복호화 과정에서 모듈로 감산으로 구현되고, 순환이동도 반대방향으로 이루어진다.

III. LEA128 코어 설계

LEA128 암·복호 코어의 전체 구조는 그림 2와 같으며, 라운드 블록, 키 스케줄러, 제어블록으로 구성된다. 라운드 블록은 24번의 라운드 변환을 통해 암·복호 연산을 수행하며, 각 라운드의 연산은 4클록 주기로 처리된다. 키 스케줄러는 각 라운드 연산에 사용되는 192비트의 라운드키를 on-the-fly 방식으로 생성한다. 저면적 구현을 위해 data-path를 32비트로 설계하였고, 입력 핀(data_in)을 공유하여 마스터키와 평문(암호문)이 시분할 방식으로 입력되도록 하였다. 또한, 암호화과정과 복호화과정에서 하드웨어 자원을 공유하도록 설계하였다.

3.1 라운드 함수 블록

128비트의 평문(암호문) 입력과 키 스케줄러에 의해 마스터키로부터 생성되는 192비트의 라운드키를 받아 라운드 변환을 반복적으로 처리하여 암호(복호)연산을 수행한다. 평문은 1클록 당 32비트씩 4클록을 소요하여 입력을 받고 최상위 32비트와 키 스케줄 함수로부터 생성된 32비트 라운드키 rk1, 다음 상위 32비트와 32비트 라운드키 rk0가 각각 XOR 연산을 수행한다. 이 연산의 수행결과를 mod-2^{32} 가산을 하고 비트 순환이동을

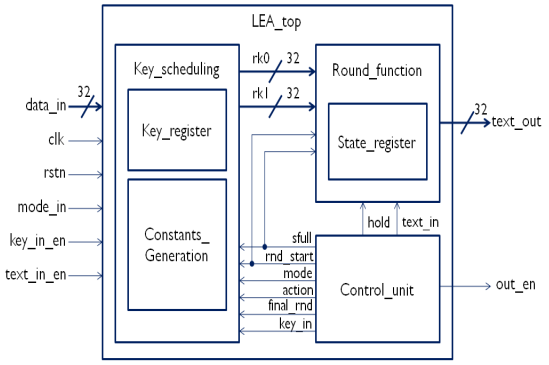

그림 2. LEA128 코어의 구조
Fig. 2. Architecture of LEA128 core

하게 되며 입력문의 최하위 32비트의 위치에 들어간다. 이때 원래 들어있던 평문은 32비트만큼 이동한 상태이며, 위의 과정을 3번 반복하여 한 번의 라운드가 수행된다. 키 스케줄에서 한 라운드의 첫 번째 라운드키를 생성하는 동안은 라운드 함수의 연산이 이루어지지 않고 나머지 세 클록 동안에 연산이 이루어진다. 즉, 한 번의 라운드는 K, K/R, K/R, K/R의 형태로 4클록동안 처리된다.[4]

LEA 알고리듬의 암호화과정과 복호화과정은 라운드 변환의 순서와 라운드키의 사용이 역순으로 처리된다. 본 논문에서는 하드웨어 공유를 통해 암호연산과 복호연산을 통합하여 구현하였으며, mode 신호에 의해 암호연산(mode=0)과 복호연산(mode=1)이 구분된다.

3.2 키 스케줄러 블록

LEA의 암·복호화에 사용되는 192비트의 라운드키는 키 스케줄링 알고리듬에 의해 생성된다. 외부로부터 입력받은 128비트 마스터키와 상수생성부에서 출력되는 상수를 32비트씩 mod-2^{32} 가산을 하고 비트 순환 이동하는 방식으로, 이때 가산에 사용되는 상수는 'L', 'E', 'A'의 ASCII 코드인 76, 69, 95에서 비롯한 766995의 제곱근 값을 16진수로 표현하여 얻은 델타상수 "0xc3efe9db, 0x44626b02, 0x79e27c8a, 0x78df30ec"를 순환 이동하여 생성된 32비트 값이다.[5] 라운드 함수에서 3번 반복 사용되는 rk1를 가장 먼저 생성하여 마지막 내부 상태변수 레지스터에 입력하고 MUX를 이용하여 클록에 따라 이동하는 rk1값을 출력하며, rk0는 내부 상태변수 레지스터에 입력되기 전의 값이다.

LEA는 대칭키 방식의 알고리듬이므로 복호화과정에 사용되는 키는 암호화과정의 역순으로 생성되며, 암호화과정의 파이널키와 복호화과정의 첫 번째 키가 일치하므로, 마스터키와 최초의 암호화 키 생성과정에서 생성된 24번째 라운드키를 저장해두었다가 라운드마다 키를 생성해서 사용하도록 설계하였다.

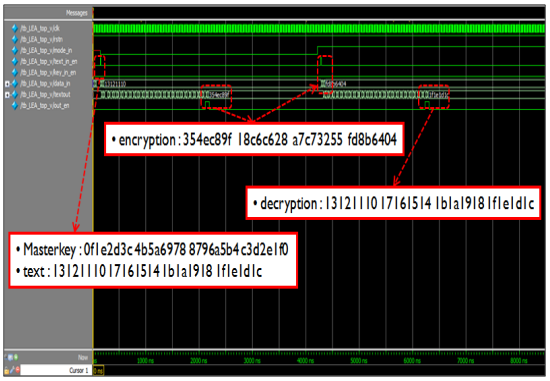

그림 3. LEA128 코어의 기능검증 결과
Fig. 3. Simulation result of LEA128 crypto-core

Ⅳ. 기능검증 및 FPGA 구현

Verilog HDL로 설계된 LEA128 코어의 기능검증 결과는 그림 3과 같으며, 128 비트의 평문 "10111213 14151617 18191a1b 1c1d1e1f"와 128 비트의 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 암호과정에서 입력으로 사용된 평문 "10111213 14151617 18191a1b 1c1d1e1f"이 출력됨을 확인함으로써 설계된 회로의 암·복호 기능이 정상적으로 동작함을 확인하였다.

기능 시뮬레이션이 완료된 LEA128 코어는 FPGA 구현을 통해 하드웨어 동작을 검증하였다. 검증 시스템은 FPGA 보드, UART 인터페이스, MFC 구동 소프트웨어 등으로 구성되며, FPGA 디바이스는 VIRTEX5 XC5VSX50T가 사용되었다. 그림 4는 데모 프로그램을 이용한 FPGA 검증결과 화면이며, 평문을 암호화한 후 이를 다시 복호화하면 원래의 평문이 출력되어 정상적으로 동작함을 확인하였다.

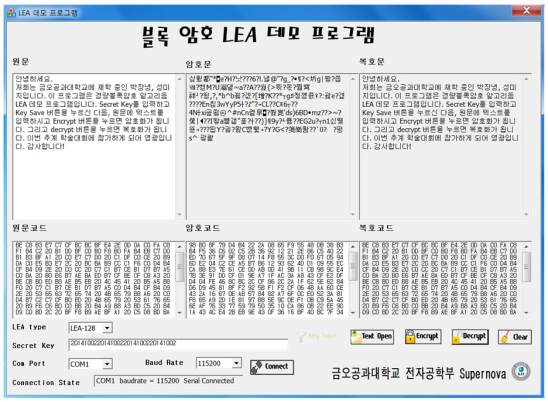

그림 4. LEA128 코어의 FPGA 검증 결과
Fig. 4. FPGA verification result of LEA128 core

Ⅴ. 결 론

본 논문에서는 한국정보통신기술협회(TTA: Telecommunications Technology Association) 표준으로 등록되어있는 128비트 블록암호 알고리듬 LEA128을 하드웨어로 구현하여 동작을 확인하였다. 저면적과 저전력 구현을 위해 암호화 과정과 복호화 과정에서 하드웨어 자원이 공유되도록 설계하였다. 설계된 LEA128 암·복호 코어는 IoT 및 모바일 기기 보안 등과 같이 저전력, 경량화가 요구되는 응용분야의 정보보호 코어로 활용이 가능할 것으로 예상된다.

감사의 글

※ 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.
[5] Deukjo Hong et al, "LEA: A 128bit Block Cipher for Fast Encryption on common Processors", WISA, ,2013.