JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 20, No. 6 : 1163~1170 Jun. 2016

ECB/CBC/OFB/CTR 운영모드와 80/128-비트 키 길이를 지원하는 PRESENT 암호 프로세서 설계

김기쁨 · 조욱래 · 신경욱*

A Design of PRESENT Crypto-Processor Supporting ECB/CBC/OFB/CTR Modes of Operation and Key Lengths of 80/128-bit

Ki-Bbeum Kim · Wook-Lae Cho · Kyung-Wook Shin*

School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Kyungbuk 39177, Korea

요 약

본 논문은 ISO/IEC 29192-2 경량 암호 표준으로 지정된 초경량 블록암호 알고리듬 PRESENT의 하드웨어 구현에 대해 기술한다. PRESENT 암호 프로세서는 80, 128비트의 마스터키 길이와 ECB, CBC, OFB, CTR의 4가지 운영모드를 지원하도록 설계되었다. 마스터키 레지스터를 갖는 on-the-fly 키 스케줄러가 포함되어 있으며, 저장된 마스터키를 사용하여 평문/암호문 블록의 연속적인 암호/복호화 처리가 가능하다. 경량화 구현을 위해 80, 128 비트의 키 스케줄링 회로가 공유되도록 최적화하였다. 라운드 블록을 64 비트의 데이터 패스로 설계하여 암호/복호화의 라운드 변환이 한 클록 사이클에 처리되도록 하였다. PRESENT 암호 프로세서를 Virtex5 FPGA로 구현하여 정상 동작함을 확인하였다. 0.18㎛ 공정의 CMOS 셀 라이브러리로 합성을 한 결과, 8,100 gate equivalents(GE)로 구현되었으며, 최대 454 MHz의 클록 주파수로 동작하여 908 Mbps의 처리율을 갖는 것으로 평가되었다.

ABSTRACT

A hardware implementation of ultra-lightweight block cipher algorithm PRESENT which was specified as a standard for lightweight cryptography ISO/IEC 29192-2 is described. The PRESENT crypto-processor supports two key lengths of 80 and 128 bits, as well as four modes of operation including ECB, CBC, OFB, and CTR. The PRESENT crypto-processor has on-the-fly key scheduler with master key register, and it can process consecutive blocks of plaintext/ciphertext without reloading master key. In order to achieve a lightweight implementation, the key scheduler was optimized to share circuits for key lengths of 80 bits and 128 bits. The round block was designed with a data-path of 64 bits, so that one round transformation for encryption/decryption is processed in a clock cycle. The PRESENT crypto-processor was verified using Virtex5 FPGA device. The crypto-processor that was synthesized using a 0.18㎛ CMOS cell library has 8,100 gate equivalents(GE), and the estimated throughput is about 908 Mbps with a maximum operating clock frequency of 454 MHz.

키워드 : 경량 블록암호, IoT 보안, PRESENT 알고리듬, 정보보안, 운영모드

Key word : Lightweight Block Cipher, IoT Security, PRESENT algorithm, Information Security, Mode of Operation

Received 18 May 2016, **Revised** 19 May 2016, **Accepted** 08 June 2016

* **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 39177, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2016.20.6.1163 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

ICT(Information and Communication Technology)의 지속적인 발전에 의해 일상생활 주변의 사물들이 인터넷을 통해 연결되어 다양한 서비스를 제공하는 이른바 사물인터넷(Internet of Things; IoT) 기술이 보편화되고 있다. IoT는 인간과 주변 환경과의 상호 연결을 위해 센싱 기술과 각종 유무선 네트워크 기술이 사용되고 있으며, 연결망을 통해 사물과 각종 서비스가 연결된다. 각종 네트워크에 연결된 다량의 센서 노드와 단말기 간에 다양한 데이터가 수집되어 처리되고, 전송 및 공유되는 IoT 기술의 특성상 다양한 보안 위험에 노출될 수 있다. IoT를 통해 유통되는 데이터가 암호화되지 않은 상태로 유통될 경우 악의적인 공격자에 의한 정보 유출 및 정보 조작이 이루어져 매우 심각한 보안 위협 문제가 발생할 수 있다. IoT 보안을 위해서는 디바이스 간 통신에서의 기밀성(confidentiality), 무결성(integrity), 및 기기 간 인증(authentication) 등을 고려해야 한다[1].

IoT 보안은 기존의 유무선 인터넷 보안과 유사하게 대칭키 블록암호(symmetric key block cipher) 방식과 공개키 암호(public-key cryptosystem) 방식을 기반으로 한다. 센서 네트워크, RFID 태그와 같이 제한된 자원을 갖는 IoT 환경에서는 저전력 소모와 작은 하드웨어 구현이 중요하며, AES[2], ARIA[3] 등 기존의 블록암호 알고리즘을 대체할 수 있는 경량 암호기술(lightweight cryptography)에 대한 연구가 활발하게 진행되어 왔다. 대표적인 경량 블록암호 알고리듬으로는 PRESENT[4, 5], HIGHT[6], CLEFIA[7], KATAN/KTANTAN[8], LEA[9] 등이 있으며, 다양한 하드웨어 구현 사례들이 발표되고 있다[10-13].

블록암호는 임의의 길이의 평문(또는 암호문)을 고정된 크기의 블록(64 비트 또는 128 비트)로 나누어 암호(또는 복호)화한다. 기본 운영모드인 ECB(Electronic Code Book) 모드는 각 블록을 독립적으로 암호(복호)화 하므로, 키 값이 고정된 상태에서는 동일한 평문(암호문)에 대해 동일한 암호문(평문)이 얻어져 보안성이 떨어지는 단점을 갖는다. 이와 같은 블록암호 ECB 모드의 보안 취약점을 보완하여 다양한 응용 환경에서 적절한 기밀성을 유지할 수 있도록 운영모드(mode of operation)가 지원되어야 한다[14].

본 논문에서는 독일 보훔루르 대학에서 개발한 64 비트 블록암호 PRESENT를 네 가지 운영모드인 ECB, CBC(Cipher Block Chaining), OFB(Output Feed-Back), CTR(Counter) 모드와 80 비트, 128 비트의 두 가지 마스터키 길이를 지원하도록 설계하고, FPGA 구현을 통해 하드웨어 동작을 검증하였다. Ⅱ장에서는 PRESENT 블록암호 알고리듬과 운영모드에 대해 설명하고, Ⅲ장에서는 PRESENT 암·복호 프로세서 설계에 대해 설명한다. 설계된 회로의 기능 검증 및 FPGA 구현에 대해 Ⅳ장에서 기술하며, Ⅴ장에서 결론을 맺는다.

Ⅱ. PRESENT 알고리듬과 운영모드

2.1. 블록암호 알고리듬 PRESENT[4,5]

국제표준화기구 ISO(International Organization for Standardization)와 IEC(International Electrotechnical Commission)에 의해 경량 암호기술 표준인 ISO/IEC 29192-2[5]로 규정된 PRESENT 알고리듬은 64 비트의 평문(암호문) 블록을 80 비트 또는 128 비트의 마스터키로 암호(복호)화하여 64 비트의 암호문(평문)을 생성하는 대칭키 방식의 블록암호이다.

PRESENT는 SPN(substitution permutation network) 구조를 기반으로 31회의 라운드 변환을 통해 평문(암호문)을 출력한다. PRESENT의 암·복호화 과정은 그림 1과 같다. 암호화 과정의 라운드 변환은 그림 1 (a)와 같이 라운드키 가산을 하는 addRoundKey, 4 비트 비선형 변환를 수행하는 SBox, 64 비트 치환를 수행하는

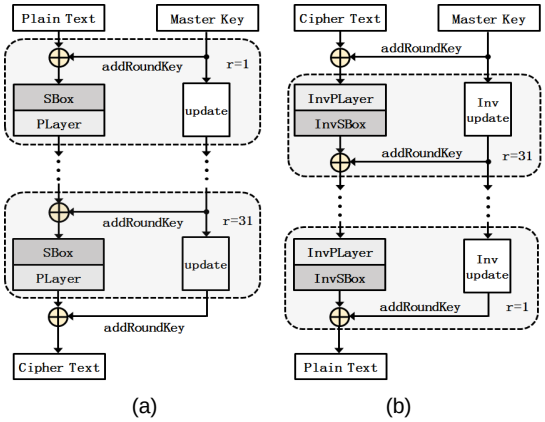


Fig. 1 Encryption and decryption of PRESENT
(a) encryption, (b) decryption

PLayer로 구성된다. 복호화 과정은 그림 1 (b)와 같이 암호화 과정의 역순으로 이루어지며, 라운드키도 역순으로 사용된다. 또한, SBox의 역변환을 위한 InvSBox와 비트 역치환을 위한 InvPLayer가 사용된다. 암·복호화 과정에서 매 라운드 마다 사용되는 라운드키는 마스터키를 바탕으로 키 스케줄러에 의해 생성된다.

2.2. 블록암호의 운영모드

블록암호의 기본 운영모드인 ECB 모드의 보안 취약성을 개선하여 기밀성을 높이기 위한 방법으로 운영모드가 사용된다. 블록암호의 운영모드는 이전 블록과 초기화 벡터(Initialization Vector; IV)의 암호(복호) 결과가 현재 블록의 암호(복호)화에 영향을 미치도록 하여 기밀성을 높이는 방법이다. 대표적인 운영모드로 CBC, OFB, CTR, CFB(Cipher Feed Back) 모드 등이 있으며, 블록간의 의존성과 오류 전파가 평문(복호문)에 미치는 영향에 따라 분류 된다[14, 15].

ECB 모드는 각 블록을 독립적으로 암호(복호)화 하는 블록암호의 기본적인 운영모드이다. 키 값이 고정된 경우에 블록의 값이 동일하면 암호문(평문)의 값도 동일하므로, 한 블록이 해독되면 나머지 블록도 해독되는 보안 취약성이 있다. CBC 모드는 이전 블록의 암호문과 현재 블록의 평문을 XOR 연산 한 후, 그 결과를 암호화한다. 첫 번째 블록의 경우 초기화 벡터(IV)와 평문의 XOR 결과를 암호화 한다. CBC 모드는 이전 블록의 암호문을 이용하여 암호화 하므로 동일한 평문의 암호화 결과로 전혀 다른 암호문이 생성되어 보안성이 높으며, 따라서 개인 정보를 다루거나 국가기관에서 사용되는 블록암호에 필수적으로 적용된다. 그러나 블록 간 병렬처리가 불가능하다는 단점이 있다.

OFB 모드는 이전 블록의 암호화 결과를 다시 암호화한 후, 현재 블록의 평문(암호문)과 XOR 연산하여 암호문(평문)으로 출력된다. 첫 번째 블록의 경우 IV를 암호화한 후, 평문(암호문)과 XOR 연산하여 암호문(평문)으로 출력된다. CTR 모드는 1씩 증가하는 계수기 값을 암호화한 후, 평문(암호문)과 XOR 연산하여 암호문(평문)으로 출력된다. CTR 모드는 이전 블록의 데이터에 대한 의존성을 갖지 않으므로, 다수의 블록에 대한 병렬처리가 가능하여 고속 동작이 가능하다. OFB 모드와 CTR 모드는 암호화와 복호화가 암호 연산만으로 처리되어 하드웨어 구현이 간단하다는 장점을 갖는다.

Ⅲ. PRESENT 암호 프로세서 설계

3.1. 전체 구조 및 운영모드 구현

4가지 운영모드(ECB, CBC, OFB, CTR)와 2가지 마스터키 길이(80 비트, 128 비트)를 지원하는 PRESENT 암호 프로세서(crypto-processor)를 설계하였다. 전체 구조는 그림 2와 같으며, PRESENT 알고리듬을 구현하는 PRESENT_Core, 4가지 운영모드 동작에 필요한 두 개의 64 비트 레지스터(iv_reg, op_reg), 16 비트 및 64 비트 XOR 게이트 그리고 제어블록으로 구성된다. 평문(암호문), IV, 마스터 키는 data_in 포트를 통해 16 비트씩 입력되고, 암호문(평문)은 data_out 포트를 통해 16 비트씩 출력된다.

ECB 모드의 암호(복호)화 과정은 다음과 같이 이루어진다. 64 비트의 평문이 16 비트씩 4 클록 주기에 걸쳐 PRESENT_Core에 입력되어 저장된 후, 암호(복호)화되어 16 비트씩 4 클록 주기에 걸쳐 출력된다.

CBC 모드의 암호화 과정은 그림 3의 동작 타이밍도로 처리된다. 먼저, 64 비트의 IV가 레지스터 iv_reg에 16 비트씩 4 클록 주기에 걸쳐 저장된다. 64 비트의 평

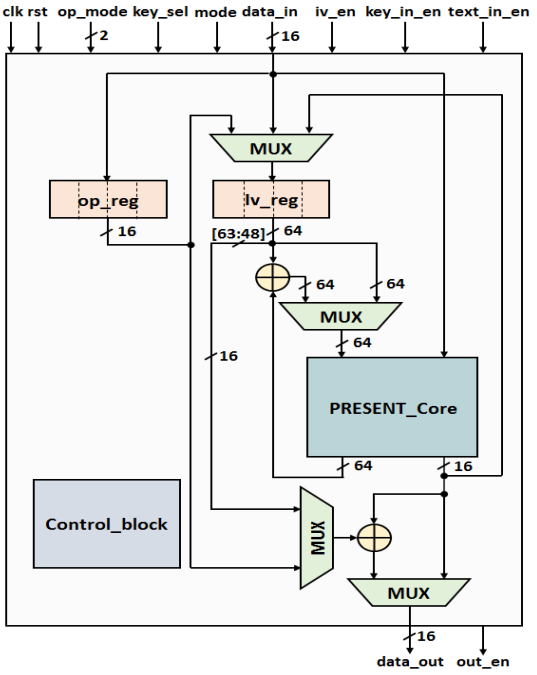


Fig. 2 Architecture of PRESENT crypto-processor supporting four modes of operation

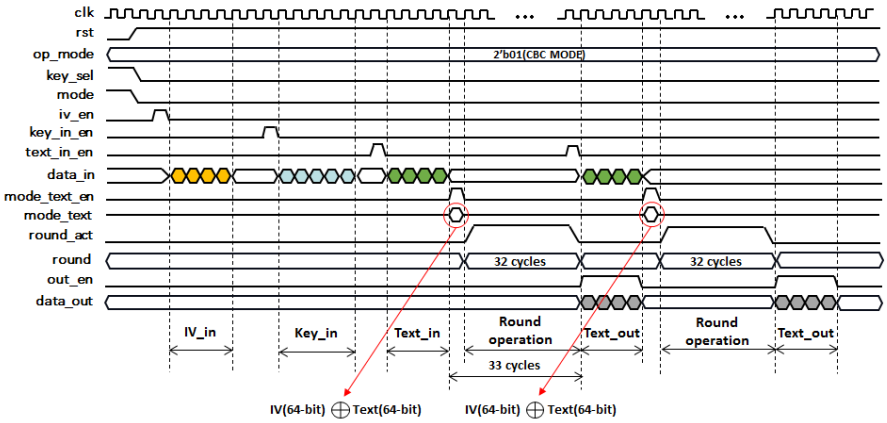


Fig. 3 Timing diagram of encryption in CBC mode

문은 16 비트씩 4 클록 주기에 걸쳐 PRESENT_Core로 입력된 후, 64 비트로 출력되어 레지스터 iv_reg에 저장된 IV와 XOR 연산되고, 그 결과가 PRESENT_Core로 입력되어 암호화 연산이 이루어진다. 암호화 결과는 out_en 신호에 의해 암호문으로 출력됨과 동시에 레지스터 iv_reg에 저장되어 다음 블록의 암호화에 사용된다. CBC 모드의 복호화 과정은 다음과 같이 이루어진다. 64 비트의 IV가 16 비트씩 4 클록 주기에 걸쳐 레지스터 iv_reg에 저장된 후, 64 비트의 암호문이 16 비트씩 4 클록 주기에 걸쳐 PRESENT_Core와 레지스터 op_reg에 저장된다. PRESENT_Core에 입력된 암호문은 복호된 후, 64 비트의 IV와 XOR 연산되어 복호문으로 출력되고, op_reg에 저장된 64 비트 암호문이 iv_reg로 옮겨 저장되어 다음 블록의 복호화에 사용된다.

OFB 모드의 암호화와 복호화는 동일한 과정으로 다음과 같이 이루어진다. 64 비트의 IV가 4 클록 주기에 걸쳐 레지스터 iv_reg 에 저장된 후, 64 비트의 평문(암호문)은 text_in_en 신호에 의해 레지스터 op_reg에 저장된다. 레지스터 iv_reg에 저장되어 있던 64 비트의 IV는 PRESENT_Core로 입력되어 암호화 연산된 후, 레지스터 op_reg에 저장되어 있는 64 비트 평문(암호문)과 XOR 연산되어 암호문(복호문)으로 출력되며, 동시에 레지스터 iv_reg에 저장되어 다음 블록의 암호(복호)화에 사용된다.

CTR 모드의 암호화와 복호화는 동일한 과정으로 다음과 같이 이루어진다. 64 비트의 IV가 4 클록 주기에 걸쳐 레지스터 iv_reg에 저장된 후, 매 블록이 처리될 때마다 1씩 증가된 계수기 값으로 갱신된다. 64 비트의 평문(암호문)은 4클록 주기에 걸쳐 레지스터 op_reg에 저장된다. 레지스터 iv_reg에 저장된 64 비트 계수기 값은 PRESENT_Core에 입력되어 암호화 연산된 후, 레지스터 op_reg에 저장 되어있던 64 비트의 평문(암호문)과 XOR연산되어 암호문(복호문)으로 출력된다.

OFB 모드와 CTR 모드의 암호화/복호화에는 복호연산 없이 암호연산만 사용되므로, core_mode 신호를 추가하여 복호화 과정에서 PRESENT_Core가 암호연산을 수행하도록 하였다.

3.2. PRESENT_Core

PRESENT_Core는 64 비트의 평문(암호문)을 80 비트 또는 128 비트의 마스터키로 암호(복호)화하여 64 비트의 암호문/평문을 생성하는 PRESENT 블록암호 알고리듬 구현 블록이다. 내부 구조는 그림 4와 같으며, 라운드 블록, 키 스케줄러 그리고 제어 블록으로 구성된다. 내부 데이터 패스를 64 비트로 구현하여 한 라운드 변환이 단일 클록에 처리되며, 64 비트 평문(암호문) 블록의 암호(복호)화에 총 32 클록이 소요된다.

라운드 블록은 64 비트 상태 레지스터(state_reg), 4 비트 비선형 변환을 수행하는 SBox 16개(SBox_64), 그 역변환을 수행하는 InvSBox 16개(InvSBox_64), 64 비트 치환을 수행하는 PLayer (PLayer_64), 그 역변환을 수행하는 InvPLayer (InvPLayer_64), 그리고 라운드키 가산을 위한 64 비트 XOR 게이트 등으로 구성된다. 상태 레지스터 state_reg는 라운드 연산의 중간결과 저장

과 CBC 암호화 모드에서 16 비트씩 입력되는 평문을 받아 64 비트로 출력하는 역할을 한다.

라운드 변환은 state_reg에 저장된 평문(암호문)이 라운드키와 가산되며, 암호화 연산에서는 SBox와 PLayer 연산이 순차적으로 수행된 결과가 state_reg에 저장되고, 복호화 연산에서는 InvPLayer 연산과 InvSBox연산이 순차적으로 수행된 결과가 state_reg에 저장된다.

키 스케줄러는 80 비트 또는 128 비트 마스터키를 받아 라운드 변환에 사용되는 64 비트 서브 라운드키를 생성하여 라운드 블록에 공급한다. 두 개의 128 비트 레지스터와 키 업데이트 모듈로 구성된다. key_state_reg는 라운드키 생성을 위한 중간키 값을 저장하는 레지스터이고, master_key_reg는 동일한 마스터키가 연속되는 평문 블록에 적용되어 암호화 되도록 마스터키를 저장한다. key_in_en 신호에 의해 새로운 마스터키가 입력될 때까지 master_key_reg에 저장된 키가 암호(복호)화 연산에 반복적으로 사용되며, 매 평문 블록마다 마스터키를 입력하지 않아도 되는 장점을 갖는다. 마스터키 길이는 key_sel 신호에 의해 결정되며, key_sel=0(80 비트 키 길이)인 경우에는 key_state_reg에 저장된 128 비트의 중간키 값에서 [79:16]이 라운드키로 출력되고, key_sel=1(128 비트 키 길이)인 경우에는 [127:64]가 라운드키로 출력된다.

key_update 블록의 내부 구조는 그림 5와 같으며, 순환이동 회로, SBox_4, InvSBox_4, 라운드 상수 가산을 위한 5 비트 XOR 게이트 등으로 구성된다. 마스터키 길이가 80 비트인 경우의 key_update 블록의 동작은 다음과 같다. 암호화의 경우에, key_state_reg에 저장된 128 비트의 중간키 값 중 하위 80비트 [79:0]을 61 비트 좌측 순환 이동시킨 결과로부터 상위 4 비트 [79:76]의 SBox_4 연산 결과, 라운드 상수와 5 비트 중간키 값 [19:15]의 XOR 연산 결과, 그리고 나머지 71 비트로 구성되는 128 비트가 key_state_reg에 저장되어 다음 라운드의 라운드키로 출력된다. 복호화 경우, key_state_reg에 저장된 중간키 값 중 4 비트 [79:76]의 SBox_4 연산 결과, 라운드 상수와 5 비트 중간키 값 [19:15]의 XOR 연산 결과, 그리고 나머지 71 비트로 구성되는 128 비트 중 하위 80 비트 [79:0]을 61 비트 우측 순환 이동시킨 결과가 key_state_reg에 저장되어 다음 라운드의 라운드키로 출력된다.

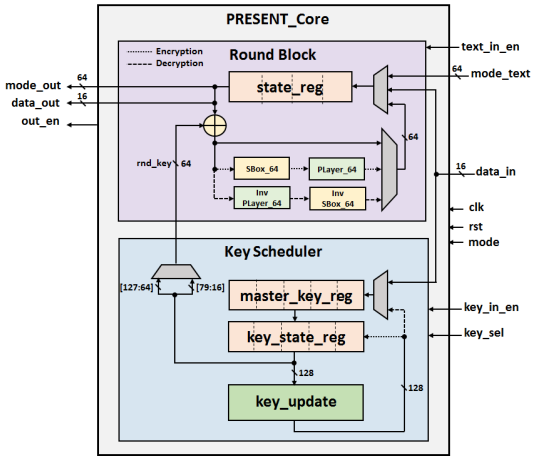


Fig. 4 PRESENT_Core block

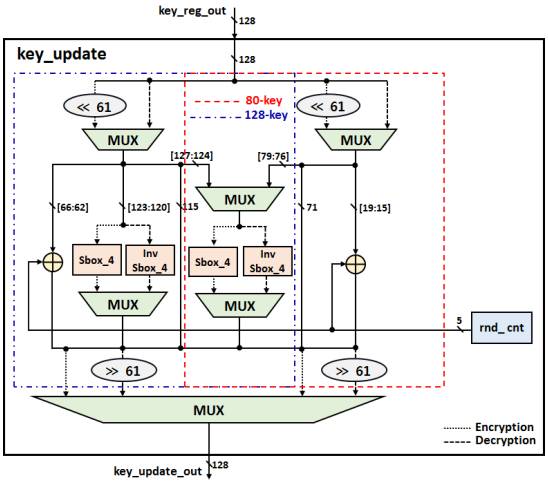


Fig. 5 Key_update block

마스터키 길이가 128 비트인 경우에도 유사하게 동작한다. 암호화의 경우, key_state_reg에 저장된 중간키 값을 61 비트 좌측 순환 이동시킨 결과로부터 상위 4 비트 [127:124]와 [123:120]의 SBox_4 연산, 라운드 상수와 5 비트 중간키 값 [66:62]의 XOR 연산 결과, 그리고 나머지 115 비트로 구성되는 128 비트 중간키 값이 key_state_reg로 저장되어 다음 라운드의 라운드키로 출력된다. 복호화의 경우, key_state_reg에 저장된 중간키 값 중 [127:124]와 [123:120]의 SBox_4 연산, 라운드 상수와 5 비트 중간키 값 [66:62]의 XOR 연산 결과, 그리고 나머지 115 비트로 구성되는 128 비트를 61 비트 우측 순환 이동시킨 결과가 key_state_reg에 저장되어 다음 라운드의 라운드키로 출력된다. 키길이가 80 비트

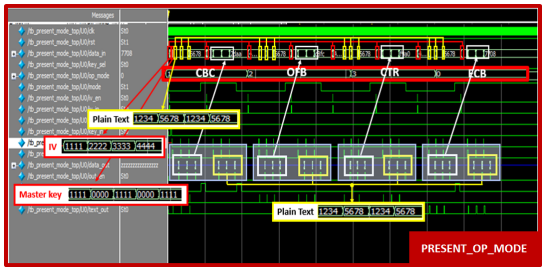
인 경우, 암호와 복호를 위해 각각 한 개의 SBox_4와 InvSBox_4가 사용되므로, 키길이 128 비트 연산에 사용되는 SBox_4와 InvSBox_4를 공유하도록 설계했다.

Ⅳ. 기능검증 및 FPGA 구현

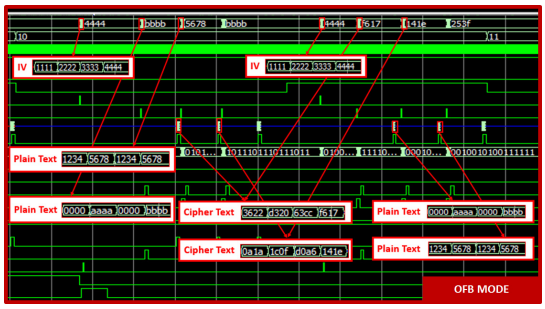
설계된 PRESENT 암호 프로세서는 RTL 시뮬레이션에 의한 기능검증과 FPGA 구현에 의한 하드웨어 동작 확인의 2단계로 검증하였다. 그림 6은 RTL 기능검증 결과의 일부를 보이고 있다. 그림 6 (a)는 64 비트의 IV "1111 2222 3333 4444", 80 비트의 마스터키 "1111 0000 1111 0000 1111", 64 비트의 평문 "1234 5678 1234 5678"을 연속적으로 입력시키면서 CBC, CTR, OFB, ECB 모드의 순서로 암호화 한 결과와 암호문을 복호화 시켜 원래의 평문이 출력됨을 확인한 시뮬레이션 결과이다. ECB 모드의 경우, 평문 "1234 5678 1234 5678"을 연속적으로 세 블록 암호화한 결과로 세 블록의 암호문이 모두 동일한 값 "d25a cca7 23dd 7708"으로 출력되었다. CBC, OFB, CTR 모드에서는 평문 "1234 5678 1234 5678"을 연속적으로 세 블록씩 암호화한 결과로 서로 다른 값을 갖는 세 블록의 암호문이 출력되었다. 각 모드의 암호화에서 얻어진 세 블록의 암호문을 복호한 결과 원래의 평문 "1234 5678 1234 5678"이 세 블록씩 출력되어 ECB, CBC, OFB, CTR 모드의 기능이 올바로 동작함을 확인하였다.

그림 6 (b)는 OFB 모드의 기능검증 결과로서, 64 비트의 IV "1111 2222 3333 4444", 64 비트의 평문 "0000 aaaa 0000 bbbb", "1234 5678 1234 5678"를 사용한 시뮬레이션 결과를 보이고 있다. 암호화 결과로 암호문 "3622 d310 63cc f617", "0a1a 1c0f d0a6 141e"가 출력되었으며, 이 암호문을 다시 복호화한 결과로 평문 "0000 aaaa 0000 bbbb", "1234 5678 1234 5678"이 출력됨을 확인할 수 있다.

설계된 PRESENT 암호 프로세서는 FPGA 보드, UART 인터페이스, PC, 구동 소프트웨어로 구성되는 검증 시스템을 통해 하드웨어 동작을 검증하였으며, Virtex5 XC5VSX-95T FPGA 디바이스가 사용되었다. PC와 FPGA 사이의 데이터 송수신은 RS232C를 통해 이루어진다. PC에서 FPGA로 전송된 평문(암호문) 데이터는 PRESENT 암호 프로세서로 입력되어 64 비트



(a)



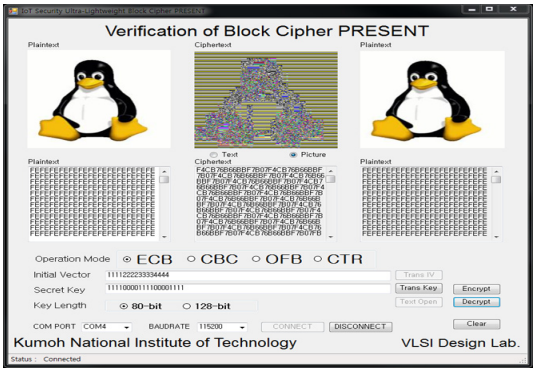
(b)

Fig. 6 Simulation results of PRESENT crypto-processor (a) Four modes (b) in OFB mode

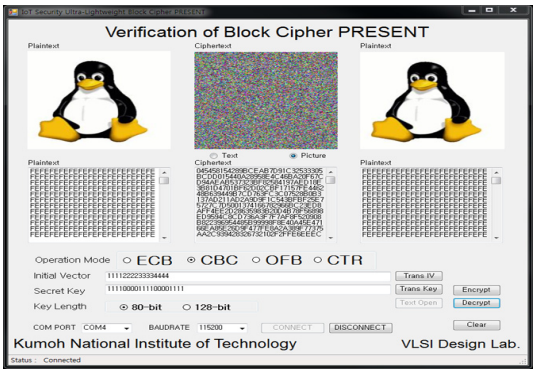
블록 단위로 암호(복호)화가 이루어진다. PRESENT 암호 프로세서에서 출력되는 암호문(평문)은 UART 통신을 통해 PC로 전송되어 화면에 표시된다.

그림 7은 FPGA 검증 결과를 보이고 있다. GUI 프로그램을 통해 PRESENT 암호 프로세서의 암호(복호)화 결과가 화면에 표시된다. 그림 7 (a),(b)에서 좌측의 원본 이미지를 FPGA로 전송하여 PRESENT 암호 프로세서에서 암호화한 결과는 중앙의 이미지와 같다. 그림 7 (a)의 중앙의 이미지는 ECB 모드로 암호화한 결과이며, 원본 이미지가 암호화 되었지만 윤곽이 드러나는 것을 볼 수 있다. 이는 ECB 모드의 경우 동일한 평문을 암호화하면 동일한 암호문이 출력되는 특성 때문이다. 그림 7 (b)의 중앙의 이미지는 동일한 원본 이미지를 CBC 모드로 암호화한 결과이다. 원본 이미지가 형체를 알아볼 수 없도록 랜덤 값으로 암호화되었음을 확인할 수 있다. CBC 모드에서는 이전 블록의 암호화 결과가 다음 블록의 암호화에 사용되므로, 동일한 평문을 암호화 하더라도 동일한 암호문이 출력되지 않기 때문이다.

그림 7 (a),(b)의 중앙의 암호화된 이미지를 다시 FPGA로 전송하여 PRESENT 암호 프로세서에서 복호화한 결과는 우측의 이미지와 같으며, 암호화에 사용된



(a)



(b)

Fig. 7 FPGA verification result of PRESENT crypto-processor (a) in ECB mode (b) in CBC mode

원본 이미지가 복원되었음을 확인할 수 있다. 그림 7의 FPGA 검증 결과에서 보는 바와 같이, 이미지를 암호화하고 암호화된 이미지를 복호하여 원래 이미지와 일치하는 결과가 출력됨으로써 설계된 PRESENT 암호 프로세서가 올바르게 동작함을 확인하였다.

Table. 1 Summary of PRESENT crypto-processor

Mode of operation	ECB, CBC, OFB, CTR	
Key size [bit]	80, 128	
Block size [bit]	64	
Cycles per block	CBC(enc), OFB(enc/dec),CTR(enc/dec)	33 cycles
	CBC(dec), ECB(enc/dec)	32 cycles
Max. clock frequency [MHz]		454 MHz
Throughput [Mbps]	@Max frequency	908
	@100 KHz	0.2
Area @100 KHz [GE]	PRESENT_Core	5,740
	Mode operation	1,950
	Total	8,100

본 논문에서 설계된 PRESENT 암호 프로세서의 특성은 표 1과 같다. 0.18㎛ CMOS 표준 셀 라이브러리로 합성한 결과, 최대 454 MHz의 클록 주파수로 동작하여 908 Mbps의 처리율을 갖는 것으로 평가되었다. 100 kHz의 주파수로 합성한 결과, 두 가지 키길이(80, 128 비트)로 암호/복호를 수행하는 PRESENT_Core는 5,740 GE로 구현되었고, 네 가지 운영모드를 지원하는 회로는 약 1,950 GE로 구현되었다. 전체 PRESENT 암호 프로세서는 8,100 GE로 구현이 되었다.

Ⅴ. 결 론

ISO/IEC 국제표준으로 승인된 64 비트 블록암호 알고리듬 PRESENT를 4가지 운영모드와 2가지 마스터키 길이를 지원하도록 하드웨어로 구현하였다. 설계된 PRESENT 암호 프로세서는 0.18㎛ CMOS 공정에서 8,100 GE로 구현되었으며, 최대 454 MHz의 클록 주파수로 동작하여 908 Mbps의 처리율을 갖는다. 설계된 PRESENT 암호 프로세서는 저면적을 특징으로 가져 IoT, RFID 환경과 같이 제한된 자원을 갖는 응용분야의 정보보호 SoC 설계에 IP로 활용이 가능하다.

ACKNOWLEDGMENTS

This work was supported by Industrial Core Technology Development Program (10049009, Development of Main IPs for IoT and Image-Based Security Low-Power SoC) funded by the Ministry of Trade, Industry & Energy.

The authors are thankful to IDEC for EDA software support.

REFERENCES

[1] C. Lu. Overview of Security and Privacy Issues in the Internet of Things [Internet]. Available: http://www.cse.wustl.edu/ ~jain/cse574-14/ftp/security.pdf

[2] FIPS-197, Advanced Encryption Standard, National Institute of Standard and Technology(NIST), November, 2001.

[3] KS X 1213:2004, 128 bit Block Encryption Algorithm

ARIA, Korean Agency for Technology and Standards (KATS), 2004.

[4] A. Bogdanov et al., "PRESENT: An Ultra-Lightweight Block Cipher," *Cryptographic Hardware and Embedded Systems (CHES 2007),* LNCS, Springer, vol. 4727, pp. 450-466, 2007.

[5] ISO/IEC Std. 29192-2, Information technology - Security techniques-Lightweight cryptography (part2): Block ciphers, International Organization for Standardization (ISO), 2012.

[6] TTA Std. TTAK.KO-12.0040/R1, 64-bit Block Cipher HIGHT, Korea Internet & Security Agency, 2008.

[7] Sony Corporation. The 128-bit Block Cipher CLEFIA: Algorithm Specification, [Internet]. Available: http://www.sony.net/Products/cryptography/clefia/download/data/clefia-spec-1.0.pdf.

[8] De Canniere, Christophe, Orr Dunkelman, and Miroslav Knežević. "KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers," *Cryptographic Hardware and Embedded Systems (CHES 2009),* Springer, pp. 272-288, 2009.

[9] TTA Std. TTAK.KO-12.0223, 128-Bit Block Cipher LEA, Telecommunications Technology Association, 2013.

[10] T. Eisenbarth, C. Paar, A. Poschmann, S. Kumar and L. Uhsadel, "A Survey of Lightweight Cryptography Implementations," *IEEE Design & Test of Computers*, vol. 24, no. 6, pp. 522-533, Nov. 2007.

[11] H.W. Park and K.W. Shin, "An efficient hardware implementation of 64-bit block cipher algorithm HIGHT," *Journal of KIICE*, vol. 15, no. 9, pp. 1933-1999, Sep. 2011.

[12] M.J. Sung and K.W. Shin, "An Efficient Hardware Implementation of Lightweight Block Cipher LEA-128/192/256 for IoT Security Applications," *Journal of KIICE*, vol. 19, no. 7, pp. 1608-1616, Jul. 2015.

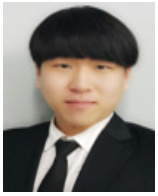
[13] G.C. Bae and K.W. Shin, "An Efficient Hardware Implementation of Lightweight Block Cipher Algorithm CLEFIA for IoT Security Applications," *Journal of KIICE*, vol. 20, no. 2, pp. 351-358, Feb. 2016.

[14] NIST Special Publication 800-38A, *Recommendation for Block Cipher Modes of Operation-Methods and Techniques*, National Institute of Standards and Technology (NIST), 2001.

[15] D.H Kim and K.W. Shin, "An Efficient Hardware Implementation of ARIA Block Cipher Algorithm Supporting Four Modes of Operation and Three Master key Lengths", *Journal of KIICE*, vol. 16, no. 11, pp. 2517-2524, Nov. 2012.

김기쁨(Ki-Bbeum Kim)

2016년 2월 금오공과대학교 전자공학부(공학사)
※관심분야 : 통신 및 신호처리용 반도체 IP 설계, 정보보호용 반도체 IP 설계



조욱래(Wook-Lae Cho)

2016년 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