JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 20, No. 2 : 351~358 Feb. 2016

IoT 보안 응용을 위한 경량 블록 암호 CLEFIA의 효율적인 하드웨어 구현

배기철 · 신경욱*

An Efficient Hardware Implementation of Lightweight Block Cipher Algorithm CLEFIA for IoT Security Applications

Gi-chur Bae · Kyung-wook Shin*

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

요 약

경량 블록 암호 알고리즘 CLEFIA의 효율적인 하드웨어 설계에 대하여 기술한다. 설계된 CLEFIA 보안 프로세서는 128/192/256-비트의 세 가지 마스터키 길이를 지원하며, 변형된 GFN(Generalized Feistel Network) 구조를 기반으로 8-비트 데이터 패스로 구현되었다. 라운드키 생성을 위한 중간키 계산용 GFN과 암호·복호 라운드 변환용 GFN을 단일 데이터 프로세싱 블록으로 구현하여 하드웨어 복잡도를 최소화하였다. 본 논문의 GFN 블록은 라운드 변환과 128-비트의 중간 라운드키 계산을 위한 4-브랜치 GFN과 256-비트의 중간 라운드키 계산을 위한 8-브랜치 GFN으로 재구성되어 동작하도록 설계되었다. Verilog HDL로 설계된 CLEFIA 보안 프로세서를 FPGA로 구현하여 정상 동작함을 확인하였다. Vertex5 XC5VSX50T FPGA에서 최대 112 MHz 클록으로 동작 가능하며, 마스터키 길이에 따라 81.5 ~ 60 Mbps의 성능을 갖는 것으로 평가되었다.

ABSTRACT

This paper describes an efficient hardware implementation of lightweight block cipher algorithm CLEFIA. The CLEFIA crypto-processor supports for three master key lengths of 128/192/256-bit, and it is based on the modified generalized Feistel network (GFN). To minimize hardware complexity, a unified processing unit with 8 bits data-path is designed for implementing GFN that computes intermediate keys to be used in round key scheduling, as well as carries out round transformation. The GFN block in our design is reconfigured not only for performing 4-branch GFN used for round transformation and intermediate round key generation of 128-bit, but also for performing 8-branch GFN used for intermediate round key generation of 256-bit. The CLEFIA crypto-processor designed in Verilog HDL was verified by using Virtex5 XC5VSX50T FPGA device. The estimated throughput is 81.5 ~ 60 Mbps with 112 MHz clock frequency.

키워드 : CLEFIA, 경량 블록 암호, 정보보안, IoT 보안, 비밀키 암호

Key word : CLEFIA, lightweight block cipher, information security, IoT security, secret key cryptography

Received 06 October 2015, **Revised** 10 November 2015, **Accepted** 24 November 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 39177, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2016.20.2.351 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

사물인터넷(Internet of Things; IoT)은 다양한 기기들이 인터넷에 연결되어 인간의 조작이나 도움 없이 서로 정보를 교환하고 공유하는 기술을 의미하며, 스마트 홈, 스마트 카, 지능형 헬스 케어와 같은 새로운 분야에 빠른 속도로 실용화되고 있다. IoT는 복잡하고 이질적인 네트워크에서 다양한 형태의 데이터가 처리되고 전송되므로, 다양한 형태의 보안 위험에 노출될 수 있다. IoT의 보안 위협은 애플리케이션, 네트워크, 단말 등 전체 구성요소에 걸쳐 존재할 수 있으며, 서버와 단말에 대한 불법 접근을 통한 가용성 침해, 정보의 조작 및 탈취를 통한 기밀성/무결성 공격, 프라이버시 침해 등이 대표적인 보안 위협이다[1-3].

IoT 보안을 위해서는 기존의 유무선 인터넷 보안과 동일한 대칭키 블록암호 방식과 공개키 암호 방식이 사용된다. 센서 네트워크, RFID 태그, 스마트카드 등 하드웨어/소프트웨어 자원이 제한되는 IoT 네트워크와 단말기의 보안을 위해서는 저전력, 저면적의 경량 블록암호(lightweight block cipher) 알고리듬이 요구된다[4, 5]. 최근에는 IoT 보안에 적합한 HIGHT (HIGh security and light weigHT)[6], LEA (Lightweight Encryption Algorithm)[7], CLEFIA[8], PRESENT[9], TEA (Tiny Encryption Algorithm)[10], mCrypton[11] 등 다양한 경량 블록암호 방식들이 제안되고 있다.

모바일 단말기를 이용한 디지털 콘텐츠 이용이 보편화됨에 따라 DRM (digital rights management) 보안 기술의 중요성이 부각되고 있다. DRM은 디지털 콘텐츠를 암호화된 데이터로 변환하여 유통하고, 인증절차를 거쳐 허가된 수신자만 암호화된 콘텐츠의 이용이 가능하도록 하는 보안기술의 한 형태이다. 소니에서는 DRM 시스템을 위한 블록 암호 알고리듬 CLEFIA를 개발하였다[8]. CLEFIA는 하드웨어/소프트웨어 자원, 동작속도, 보안성 등의 측면에서 IoT에 적합하도록 개발되었다. 또한 선형공격, 불능 차분공격 등의 보안공격에 대한 안전성이 입증되었고, 경량 구현이 가능하여 RFID, IoT 등 하드웨어/소프트웨어 자원의 최소화가 중요한 분야에 적합한 것으로 평가되고 있다.

본 논문에서는 블록암호 CLEFIA를 IoT 환경에 적합하도록 저면적으로 설계하였으며, FPGA 구현을 통해 하드웨어 동작을 검증하였다. 암호화/복호화와 중간키 생성을 위한 하드웨어 자원의 공유를 통해 설계를 최적화하였다. Ⅱ장에서는 CLEFIA 블록암호 알고리듬에 대해 간략히 설명하고, Ⅲ장에서는 CLEFIA 보안 프로세서 설계에 대해 설명한다. 설계된 회로의 기능 검증 및 FPGA 구현 결과에 대해 Ⅳ장에서 기술하며, Ⅴ장에서 결론을 맺는다.

Ⅱ. 경량 블록암호 알고리즘 CLEFIA[8]

CLEFIA는 128-비트의 블록길이와 128/192/256-비트의 세 가지 마스터키 길이를 지원하는 대칭키 방식의 블록암호이며, 다양한 경량화 기법을 적용하여 적은 하드웨어로 구현이 가능하면서 보안성능과 암호/복호 연산성능이 우수한 것으로 평가된다. 마스터키의 길이 128/192/256-비트에 따라 각각 18/22/26회의 라운드 변환을 통해 암호화/복호화가 이루어지며, 라운드 변환과 중간키 생성은 변형된 Feistel 구조인 GFN (Generalized Feistel Network)을 기반으로 한다. GFN은 4-branch와 8-branch의 두 가지 형태가 사용되며, 4-branch GFN은 라운드 변환과 128-비트의 중간키 생성에 사용되고, 8-branch GFN은 192/256-비트의 마스터키로부터 256-비트 중간키 생성에 사용된다.

그림 1은 $GFN_{d,r}$의 슈도 코드이다. $d=4,8$는 GFN의 branch 수를 나타내고, r은 라운드 변환 수를 나타낸다. X_i, Y_i $(0 \le i < d)$는 각각 d개의 32-비트 입력, 출력을 나타내며, RK_i $(0 \le i < dr/2)$는 $dr/2$개의 32-비트 라운드키를 나타낸다. $GFN_{d,r}$은 32-비트 단위로 데이터를 처리하며, 두 가지 F-함수 (F_0, F_1), XOR 연산 그리고 32-비트 단위의 순환이동으로 구성된다.

CLEFIA의 암호화/복호화 데이터 프로세싱은 그림 2와 같이 4-branch GFN인 $GFN_{4,r}$로 구성되며, 암호화와 복호화는 역순으로 이루어지며, 또한 라운드키 가산 순서와 순환이동의 방향도 반대로 이루어진다.

데이터 프로세싱 블록을 구성하는 F-함수 F_0, F_1은 그림 3과 같이 라운드키 가산을 위한 XOR, 4개의 비선형 S-box, 확산 매트릭스 M_0, M_1의 곱셈으로 구성된다. S-box S_0, S_1의 출력과 M_0, M_1의 곱셈은 기약다항식 $p(z)= z^8+z^4+z^3+z^2+1$으로 정의되는 유한체 $GF(2^8)$에서 연산된다.

$GFN_{4,r} : \begin{cases} \{\{0,1\}^{32}\}^{2r} \times \{\{0,1\}^{32}\}^4 \rightarrow \{\{0,1\}^{32}\}^4 \\ (RK_{0(32)}, \ldots, RK_{2r-1(32)}, X_{0(32)}, \ldots, X_{3(32)}) \mapsto Y_{0(32)}, \ldots, Y_{3(32)} \end{cases}$

Step 1. $T_0 \mid T_1 \mid T_2 \mid T_3 \leftarrow X_0 \mid X_1 \mid X_2 \mid X_3$
Step 2. For $i = 0$ to $r-1$ do the following:
Step 2.1 $T_1 \leftarrow T_1 \oplus F_0(RK_{2i}, T_0)$,
$T_3 \leftarrow T_3 \oplus F_1(RK_{2i+1}, T_2)$
Step 2.2 $T_0 \mid T_1 \mid T_2 \mid T_3 \leftarrow T_1 \mid T_2 \mid T_3 \mid T_0$
Step 3. $Y_0 \mid Y_1 \mid Y_2 \mid Y_3 \leftarrow T_3 \mid T_0 \mid T_1 \mid T_2$

(a)

$GFN_{8,r} : \begin{cases} \{\{0,1\}^{32}\}^{4r} \times \{\{0,1\}^{32}\}^8 \rightarrow \{\{0,1\}^{32}\}^8 \\ (RK_{0(32)}, \ldots, RK_{4r-1(32)}, X_{0(32)}, \ldots, X_{7(32)}) \mapsto Y_{0(32)}, \ldots, Y_{7(32)} \end{cases}$

Step 1. $T_0 \mid T_1 \mid \ldots \mid T_7 \leftarrow X_0 \mid X_1 \mid \ldots \mid X_7$
Step 2. For $i = 0$ to $r-1$ do the following:
Step 2.1 $T_1 \leftarrow T_1 \oplus F_0(RK_{4i}, T_0)$,
$T_3 \leftarrow T_3 \oplus F_1(RK_{4i+1}, T_2)$,
$T_5 \leftarrow T_5 \oplus F_0(RK_{4i+2}, T_4)$,
$T_7 \leftarrow T_7 \oplus F_1(RK_{4i+3}, T_6)$
Step 2.2 $T_0 \mid T_1 \mid \ldots \mid T_6 \mid T_7 \leftarrow T_1 \mid T_2 \mid \ldots \mid T_7 \mid T_0$
Step 3. $Y_0 \mid Y_1 \mid \ldots \mid Y_6 \mid Y_7 \leftarrow T_7 \mid T_0 \mid \ldots \mid T_5 \mid T_6$

(b)

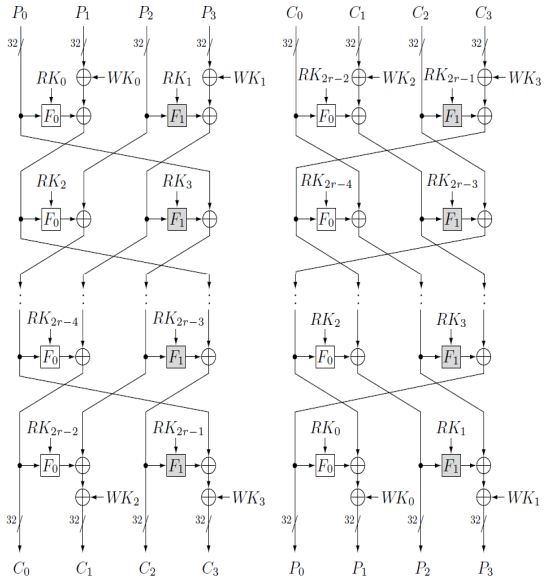
Fig. 1 Pseudo codes for GFN of CLEFIA (a) 4-branch GFN, (b) 8-branch GFN

암호화/복호화 라운드 변환의 F-함수에는 라운드키 RK_i가 사용되며, 라운드키 RK_i는 마스터키를 GFN으로 처리하여 만들어지는 중간키, 키 스케줄러 내부에서 생성되는 상수값 그리고 마스터키의 XOR 연산에 의해 생성된다. 중간키는 마스터키를 평문처럼 GFN으로 처리하여 생성되며, 이 때 F-함수에 사용되는 라운드키는 키 스케줄러 내부에서 생성되는 상수값이 사용된다.

키 스케줄러는 GFN에 의해 생성된 중간키를 저장한 뒤 더블 스왑(double swap)을 통해 갱신해가며 마스터키와 라운드 상수값의 XOR 연산으로 라운드키를 생성한다. 더블 스왑은 순환이동의 한 형태이며, 상수값 생성 블록은 마스터키 길이에 따라 정해지는 초기값을 갱신해가며 on-the-fly 방식으로 상수 값을 생성한다.

Ⅲ. CLEFIA 보안 프로세서 설계

128-비트의 평문/암호문 블록을 암호화/복호화하여 128-비트의 암호문/평문을 생성하는 CLEFIA 보안 프로세서를 설계하였다. 설계된 CLEFIA 보안 프로세서는 128/192/256-비트의 세 가지 마스터키를 지원하며, 하드웨어 자원의 최소화를 위한 다양한 방법을 적용하

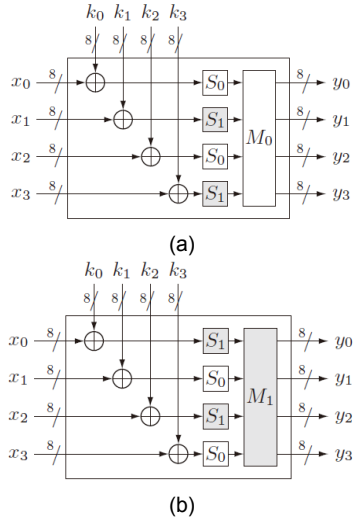


(a) (b)

Fig. 2 Structures for encryption / decryption data processing of CLEFIA (a) encryption, (b) decryption

였다. 전체 구조는 그림 4와 같으며, 데이터 프로세싱 블록, 키 스케줄러, 제어 블록으로 구성된다.

데이터 프로세싱 블록은 128-비트의 데이터 레지스터, 256-비트의 마스터키 레지스터 그리고 라운드 변환과 중간키 생성에 사용되는 GFN 연산회로로 구성된다.



(a)

(b)

Fig. 3 F-function (a) F_0, (b) F_1

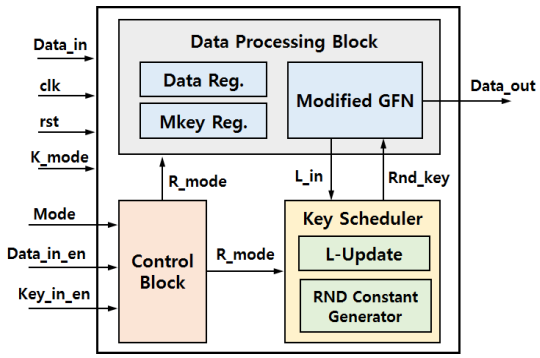


Fig. 4 Architecture of CLEFIA crypto-processor

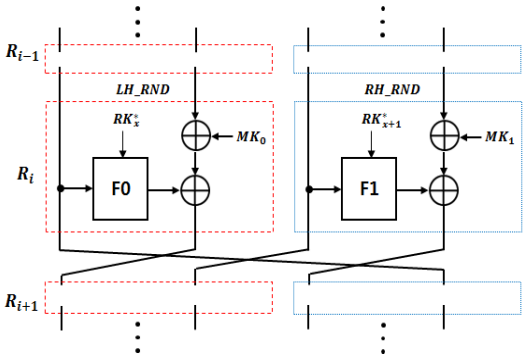


Fig. 5 Modified structure of data processing for encryption/decryption

키 스케줄러 블록은 데이터 프로세싱 블록의 GFN 연산으로 생성된 중간키 값을 저장한 후, 더블 스왑 연산을 통해 업데이트 하면서 세미 라운드키를 생성한다.

3.1. 데이터 프로세싱 블록

데이터 프로세싱 블록은 암호화/복호화 과정의 라운드 변환과 128-비트 중간키 생성을 위한 4-branch GFN과 256-비트 중간키 생성을 위한 8-branch GFN 기능을 수행한다. 하드웨어 복잡도를 최소화하기 위해 다음과 같은 방법을 적용하여 설계하였다. 첫째, GFN의 핵심 연산블록인 F-함수 F_0, F_1은 그림 3에서 보는 바와 같이 4개의 S-box와 확산 매트릭스 곱셈으로 구성되므로 가장 많은 하드웨어를 사용하는 블록이다. 따라서 데이터 프로세싱 블록의 하드웨어를 최소화하기 위해서는 F-함수의 최적화가 중요하다.

그림 2의 GFN 구조는 암호화와 복호화의 순환이동이 반대방향으로 이루어지며, 이는 하드웨어 복잡도를 증가시키는 요인이 된다. F-함수의 적용 순서와 라운드키 가산 순서를 조정하여 암호화와 복호화의 순환이동 방향이 동일해지도록 만들고, 키 스케줄러 블록의 마스터키 가산을 데이터 프로세싱 블록으로 이동시킴으로써 XOR 개수를 줄이는 방법이 제안되었다[12]. 본 논문에서는 문헌 [12]의 방법을 적용하여 그림 5와 같은 변형된 데이터 프로세싱 구조를 적용하여 설계하였다. 그림 5의 변형된 구조를 적용하면 암호화와 복호화의 순환이동 방향이 동일해지지만, F_0, F_1 함수의 적용 순서와 세미 라운드키 및 마스터키의 적용 순서가 달라진다. 그림 5의 변형된 데이터 프로세싱 구조에서 라운드 변환의 좌측 반쪽 라운드 LH_RND와 우측 반쪽 라운드 RH_RND는 동일한 구조를 가지며, F_0, F_1 함수만 다르다. 그림 3에서 볼 수 있듯이, F_0, F_1 함수는 확산 매트릭스 M_0, M_1와 S-box S_0, S_1의 적용 순서만 다르므로, 그림 6-(a)와 같이 F_0와 F_1 함수를 단일 회로로 통합한 구조로 구현할 수 있다. 그림 6-(a)의 통합 F-함수 구조에서는 라운드 r과 좌측 또는 우측 반쪽 라운드에 따라 S-box S_0와 S_1, 그리고 확산 매트릭스 M_0와 M_1이 선택적으로 수행된다.

본 논문에서는 그림 6-(a)의 통합 F-함수 구조를 그림 6-(b)와 같이 8-비트 데이터 패스로 축소하여 설계함으

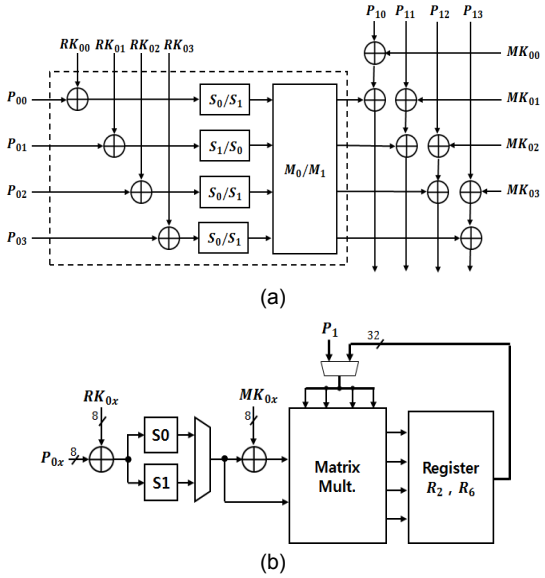


Fig. 6 Unified F-function structure (a) 32-bit datapath, (b) 8-bit datapath

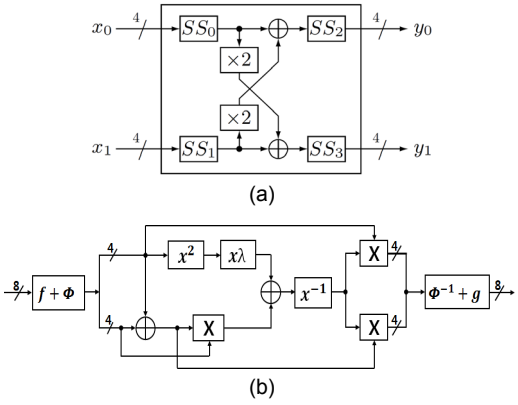


Fig. 7 S-box (a) S_0 , (b) S_1

로써 확산 매트릭스 M_0, M_1 곱셈과 S-box에 의한 하드웨어를 최소화하였다. 각각의 반쪽 라운드가 4 사이클 처리되므로, 한 라운드가 8 사이클에 처리된다. 따라서 암호화/복호화에 소요되는 사이클 수는 늘어나지만, GFN에 필요한 F-함수의 개수, F-함수에 필요한 S-box, XTIME, XOR 등의 하드웨어가 최소화된다.

하드웨어 최소화를 위해서는 F-함수 구현에 필요한 곱셈기와 S-box를 효율적으로 구현해야 한다. 확산 매트릭스 곱셈을 위해 x2, x4, x6, x8, x10의 상수 곱셈이 필요하지만, 본 논문에서는 하드웨어 최소화를 위해 x2를 수행하는 XTIME 블록 3개와 8-비트 XOR만을 사용하여 상수 곱셈들을 구현하였다.

8비트 데이터 치환을 수행하는 S-box는 256-바이트의 LUT로 구현할 수 있으나, 이는 너무 많은 하드웨어 자원을 소모하게 된다. 본 논문에서는 조합회로를 기반으로 S-box를 설계하였다. S-box S_0는 4-비트 비선형 S-box SS_x 4개와 x2를 수행하는 XTIME 블록 2개 그리고 XOR를 사용하여 그림 7-(a)와 같이 구현하였다. 이와 관련된 모든 연산은 기약다항식 $q_0(z)=z^4+z+1$로 정의되는 유한체 $GF(2^4)$에서 연산된다. 한편, S-box S_1은 유한체 $GF(2^8)$ 기반의 연산 대신에 $GF((2^4)^2)$의 복합체(composite field) 연산을 이용하여 그림 7-(b)와 같이 설계함으로써 하드웨어가 최소화되도록 하였다[13]. S-box S_1의 연산은 기약다항식 $q_0(z)=z^4+z+1$으로 정의되는 유한체 $GF(2^4)$와 $q_1(z)=z^2+z+\lambda$로 정의되는 유한체 $GF((2^4)^2)$에서 연산되며, 본 논문에서는 $\lambda=8$을 적용하였다.

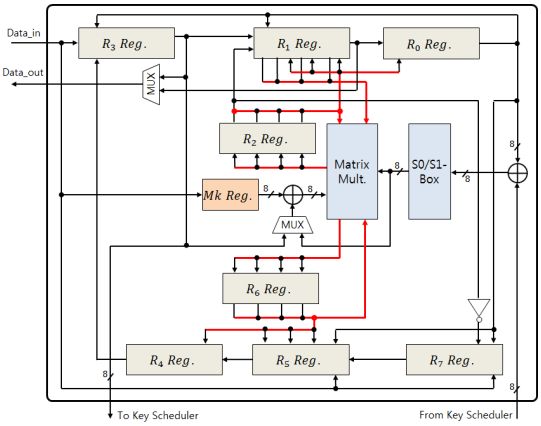


Fig. 8 Data processing block

지금까지 서술된 방법을 적용하여 설계된 데이터 프로세싱 블록은 그림 8과 같은 구조를 갖는다. 8개의 32-비트 레지스터 $R_0 \sim R_7$와 256-비트의 마스터키 레지스터 $MKReg$, 매트릭스 곱셈기, S0/S1-box 그리고 마스터키 및 라운드키 가산을 위한 XOR 게이트 등으로 구성된다.

3.2. 키 스케줄러 블록

CLEFIA의 세미 라운드키는 마스터키로부터 생성되는 중간키와 라운드 상수값을 XOR 연산하여 생성된다. 중간키는 마스터키를 GFN으로 처리하여 생성되며, 마스터키가 128-비트인 경우에는 4-branch GFN로부터 128-비트 중간키가 생성되고, 마스터키가 192/256-비트인 경우에는 8-branch GFN에 의해 256-비트 중간키가 생성된다.

본 논문에서는 GFN 공유 개념을 적용하여 데이터 프로세싱 블록의 GFN이 중간키 생성에도 사용되도록 함으로써 하드웨어 최적화를 이루었다. 설계된 키 스케줄러는 그림 9와 같으며, 중간키 레지스터 $L_0 \sim L_7$, 더블 스왑 블록, 라운드에 따라 상수값을 생성하는 상수 생성기 그리고 중간키와 라운드 상수값을 연산하는 XOR 게이트 등으로 구성된다. 중간키 레지스터는 데이터 프로세싱 블록의 GFN 연산을 통해 생성된 중간키 값을 저장하고 더블 스왑을 통해 갱신하는 기능을 수행한다. 중간키 갱신값과 라운드 상수의 XOR 연산 결과인 세미 라운드키는 그림 8의 데이터 프로세싱 블록으로 입력되어 가산된다.

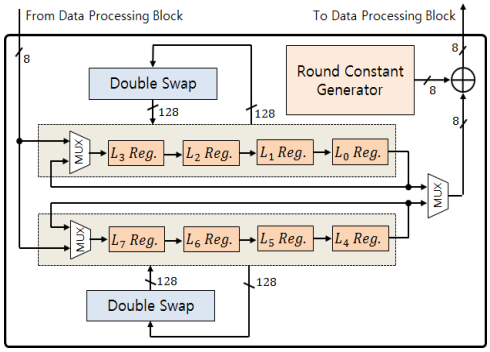


Fig. 9 Key scheduling block

설계된 라운드 상수 생성 블록은 그림 10과 같으며, 초기값 레지스터(IV Reg.), $GF(2^{16})$상에서 곱하기 2(x2) 또는 곱하기 1/2(x2^{-1})을 연산하는 유한체 곱셈기, 1-비트 순환이동 블록(SFT1b), 8-비트 순환이동 블록(SFT8b), 그리고 XOR 게이트 등으로 구성된다. 마스터키 길이에 따른 중간키 생성과 암호화/복호화를 위한 라운드 상수 생성에 9개의 라운드 상수 초기값이 선택적으로 사용되며, 라운드키는 매 라운드 마다 on-the-fly 방식으로 생성된다.

Ⅳ. 기능 검증

Verilog HDL로 설계된 CLEFIA 보안 프로세서의 기능검증 결과는 그림 11과 같다. 128비트의 평문 “00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F”와 256비트의 마스터키 “FF EE DD CC BB AA 99 88 77 66 55 44 33 22 11 00 F0 E0 D0 C0 B0 A0 90 80 70 60 50 40

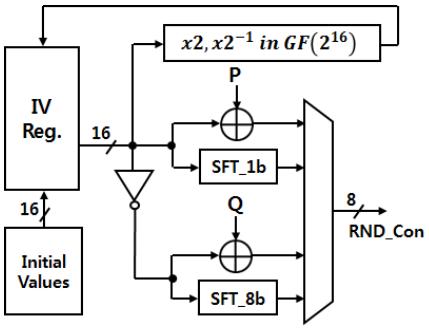


Fig. 10 Round constant generation block

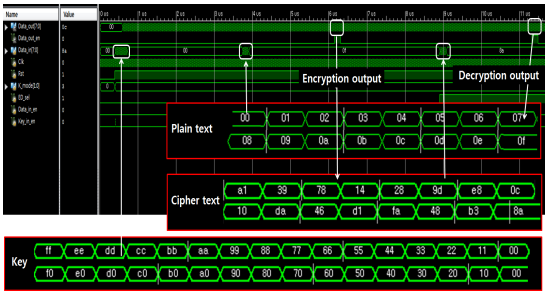


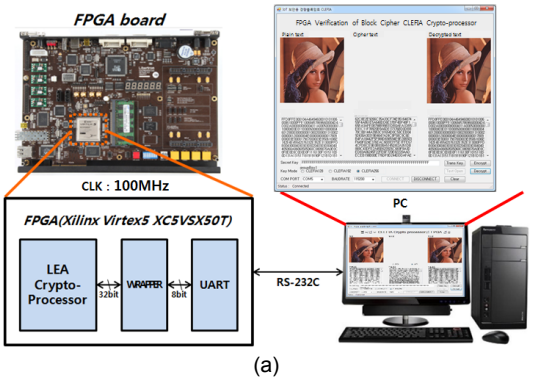
Fig. 11 Simulation result of CLEFIA crypto-processor

30 20 10 00”를 사용하여 시뮬레이션한 결과를 보이고 있다. 암호화 결과로 암호문 “A1 39 78 14 28 9D E8 0C 10 DA 46 D1 FA 48 B3 8A”이 출력되었고, 이를 동일한 마스터키로 다시 복호화한 결과로 평문 “00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F”이 출력되었으며, 설계된 CLEFIA 보안 프로세서의 논리기능이 올바로 동작함을 확인하였다.

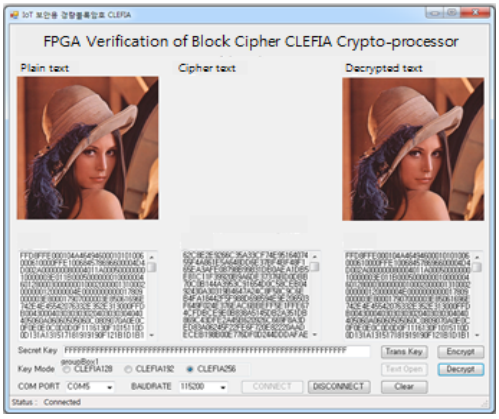
기능검증이 완료된 CLEFIA 코어는 FPGA 구현을 통해 하드웨어 동작을 검증하였다. FPGA 보드, UART 인터페이스, 구동 소프트웨어로 구성된 검증시스템은 그림 12-(a)와 같으며, Xilinx Virtex5 XC5VSX50T FPGA 디바이스가 사용되었다. PC에서 입력된 비밀키와 평문/암호문 데이터가 RS232C 통신을 통해 FPGA로 입력되고, FPGA에서 출력되는 암호문/평문 데이터가 표시된다. 그림 12-(b)는 FPGA 검증 결과를 보이고 있다. 평문을 암호화하고, 암호문을 복호화하여 원래의 평문과 일치하는 복호결과가 출력되어 FPGA에 구현된 CLEFIA 프로세서가 올바르게 동작함을 확인하였다.

설계된 CLEFIA 보안 프로세서는 FPGA 합성결과 1,563개의 LUT-FF pair로 구현되었다. 최대 112 MHz 클록 주파수로 동작이 가능하며, 128/192/256비트의 3가지 마스터키 길이에 따라 81.5 ~ 60 Mbps 성능을 갖는 것으로 평가되었다.

표 1은 본 논문에서 설계된 CLEFIA 보안 프로세서와 문헌에 발표된 사례의 비교를 보이고 있다. 문헌 [13]과 [14]의 CLEFIA 프로세서는 128-비트의 마스터키 길이만 지원하며, 128-비트 데이터 패스로 설계된 사례이다. 본 논문의 CLEFIA 프로세서는 세 가지 마스터키 길이(128/192/256-비트)를 지원하면서 문헌 [13]과 [14]의 사례와 비슷한 등가 게이트로 구현되었다. 본 논문의 설계에서는 데이터 프로세싱 블록을 8-비트 데이터 패



(a)



(b)

Fig. 12 FPGA verification of CLEFIA crypto- processor (a) FPGA verification system, (b) FPGA verification result

스로 축소하여 설계함으로써 회로 복잡도를 최소화시켰으며, 이에 의해 암호/복호에 소요되는 사이클 수가 8배 정도 증가하였다. 본 논문의 설계에 비해 문헌 [13]과 [14]의 최대 동작 주파수가 2.5 ~ 3배 빠른 것은 공정 기술의 차이에 의한 것으로 평가된다.

V. 결 론

소니에서 개발되고 ISO IEC 국제표준으로 승인된 128-비트 블록암호 알고리즘 CLEFIA를 FPGA로 구현하여 동작을 확인하였다. 암호화/복호화 라운드 연산과 중간키 생성이 하나의 데이터 프로세싱 블록에서 처리되도록 설계하여 하드웨어가 최소화되도록 하였다. 설계된 CLEFIA 보안 프로세서는 Xilinx FPGA 디바이스

Table. 1 Comparison of CLEFIA crypto-processors

	Ref [13]	Ref [14]	our design
Key length [bit]	128	128	128/192/256
Data-path [bit]	128	128	8
Area [GE]	12,010	13,830	13,530
# of cycles	18	18	176/208/240
Max. Freq. [MHz]	422	565	169
Technology [nm]	90	90	180

에서 1,563개의 LUT-FF pair로 구현되었으며, IoT 및 RFID 환경 등과 같이 경량화가 요구되는 응용분야의 정보보호 코어로 활용이 가능하다.

ACKNOWLEDGMENTS

- This paper was supported by Research Fund, Kumoh National Institute of Technology
- The authors are thankful to IDEC for supporting EDA softwares

REFERENCES

[1] D.H. Kim, S.W Yoon and Y.P. Lee, "Security for IoT Services," *Information and Communications Magazine,* vol. 30, no. 8, pp. 53-59, Jul. 2013.

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

[3] B.I Jang and C.S. Kim, "A study on the security technology for the internet of things," *Journal of Security Engineering*, vol.11, no.5, pp.429-438, 2014.

[4] 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, 2007.

[5] M.J. Sung and K.Y. Shin, "An Efficient Hardware Implementation of Lightweight Block Cipher LEA-128/192/ 256 for IoT Security Applications," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 19, no. 7, pp, 1608~1616, Jul. 2015.

[6] *HIGHT Algorithm Specification*, Korea Internet and

Security Agency, Jul. 2009.

[7] TTA Standard TTAK.KO-12, *128-bit Lightweight Block Cipher LEA*, Telecommunication Technology Association, Dec. 2013.

[8] *The 128-bit Block Cipher CLEFIA : Algorithm Specification*, Sony Corporation, 2007.

[9] A. Bogdanov et al., "PRESENT: An Ultra-Lightweight Block Cipher," *Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES 07)*, LNCS 4727, Springer, pp. 450-466, 2007.

[10] D. Wheeler and R. Needham, "TEA, a Tiny Encryption Algorithm," *Proc. of the Second International Workshop on Fast Software Encryption*, pp. 97-110, 1995.

[11] C.H. Lim and T. Korkishko, "mCrypton - A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors," *Proc. of Information Security Applications*, LNCS 3786, pp. 243-258, Aug. 2005.

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

[13] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "Hardware Implementations of the 128-bit Blockcipher CLEFIA," Technical report of IEICE, vol. 107, no. 141, ISEC2007-49, pp. 29-36, Jul. 2007 (in Japanese)

[14] T. Sugawara, N. Homma, T. Aoki and A. Satoh, "High-Performance ASIC Implementation of the 128-bit Block Cipher CLEFIA," *Proc. of 2008 IEEE International Symposium on Circuits and Systems(ISCAS 2008)*, pp. 2925-2928, May, 2008.

배기철(Gi-Chur Bae)

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