ISSN 1229-800X The Transactions of the Korean Institute of Electrical Engineers Vol. 67P, No. 1, pp. 9~14, 2018
http://doi.org/10.5370/KIEEP.2018.67.1.009

IoT 애플리케이션을 위한 AES 기반 보안 칩 설계

A Design of an AES-based Security Chip for IoT Applications using Verilog HDL

박 현 근* · 이 광 재†
(Hyeon-Keun Park · Kwangjae Lee)

Abstract - In this paper, we introduce an AES-based security chip for the embedded system of Internet of Things(IoT). We used Verilog HDL to implement the AES algorithm in FPGA. The designed AES module creates 128-bit cipher by encrypting 128-bit plain text and vice versa. RTL simulations are performed to verify the AES function and the theory is compared to the results. An FPGA emulation was also performed with 40 types of test sequences using two Altera DE0-Nano-SoC boards. To evaluate the performance of security algorithms, we compared them with AES implemented by software. The processing cycle per data unit of hardware implementation is 3.9 to 7.7 times faster than software implementation. However, there is a possibility that the processing speed grow slower due to the feature of the hardware design. This can be solved by using a pipelined scheme that divides the propagation delay time or by using an ASIC design method. In addition to the AES algorithm designed in this paper, various algorithms such as IPSec can be implemented in hardware. If hardware IP design is set in advance, future IoT applications will be able to improve security strength without time difficulties.

Key Words : Security chip, Hardware AES, Field programmable Gate array, Verilog HDL

1. 서 론

사물인터넷(Internet of Things; IoT)이란 사물 혹은 인간이 임베디드 통신시스템을 통해 긴밀하게 상호작용할 수 있도록 네트워크로 연결된 상태를 의미한다[1]. 사물인터넷 기술은 산업의 여러 분야에서 보다 효율적이고 편리하게 인간의 삶을 변화시킬 것으로 전망된다. 특히 에너지, 의료, 제조업 등의 분야는 이 기술을 적용하기 위해 많은 연구를 진행중이며, 이와 관련한 사물인터넷 시장은 꾸준히 증가하고 있다[2]. 그러나 많은 관심과 빠른 발전 속도로 인해 이러한 사물인터넷과 관련한 사이버 범죄 또한 증가하고 있다. 이런 현상은 사물인터넷 기기의 임베디드 시스템 특유의 적은 자원으로 인한 보안적용의 어려움이 있다[3]. 임베디드 시스템은 몇 가지 특징을 갖는데, 소형, 경량, 저전력, 실시간 수행(Real-time Operation) 등이다. 이들 중 소형과 저 전력의 특성으로 인하여 임베디드 시스템은 상용 PC에 적용되는 고성능의 중앙처리장치(Central Processing Unit; CPU)를 장착하지 않고 저전력, 저사양의 CPU 또는 마이크로프로세서(Micro Processer Unit; MPU)를 탑재한다. 때문에 임베디드 시스템 상에서의 보안 알고리즘의 적용은 기존 수행하던 프로세스(Process)나 태스크(Task)의 성능을 저하시킬 수 있으며, 이는 실시간 수행능력을 크게 저하시킬 수 있다. 이를 보완하기 위해 높은 성능의 CPU를 사용하면 저전력 특징을 만족할 수 없기 때문에 임베디드 시스템 환경에서 소프트웨어로 구현된 보안 알고리즘은 적용에 한계가 있다. 그럼에도 불구하고 최근 IP카메라 해킹, 차량 인포테인먼트(In-Vehicle Infotainment; IVI) 해킹과 같은 지속적으로 발생하는 보안사고로 인하여 사물인터넷에 대한 보안 시스템 구축이 다수 진행되고 있다[4]. 이에 따라 사물인터넷 관련 기기들은 자체적으로 보안강도(Security Strength)를 가질 수 있도록 성능저하를 감수하고 보안 알고리즘 라이브러리를 내장하여 소프트웨어적으로 구현하거나, CPU 연산능력에 지장을 주지 않으면서 보안능력을 가질 수 있도록 보안기능을 가진 칩을 사용하여 보안 알고리즘을 구현하고 있다[5].

본 논문에서는 여러 보안 알고리즘 중, 고급 암호화 표준(Advanced Encryption Standard; AES) 알고리즘을 탑재한 하드웨어 보안 칩을 FPGA(Field Programmable Gate Array)상에 구현을 소개한다. 하드웨어 구현은 저전력과 CPU 연산능력 저하 문제를 해결할 수 있는 방식이다. 보안 알고리즘의 성능평가를 위해 이를 소프트웨어로 구현된 AES와 비교한다.

2. 배경지식

2.1 고급 암호화 표준 알고리즘

본 논문에서 보안 칩 상에 구현하는 보안 알고리즘은 AES이다. AES는 2002년부터 미 국가안보국(NSA)에 의해 미국의 1급 비밀 암호화에 사용되는 알고리즘이다[6]. AES는 128-bit의 평문과 128-bit의 암호키를 가지고 128-bit의 암호문을 생성하는 블록암호(Block Cipher)방식의 보안 알

† Corresponding Author : Dept. of Information Security Engineering, Sang Myung University, Korea
E-mail : begleam@smu.ac.kr
* Dept. of Information Security Engineering, Sang Myung University, Korea
접수일자 : 2018년 2월 13일
최종완료 : 2018년 2월 26일

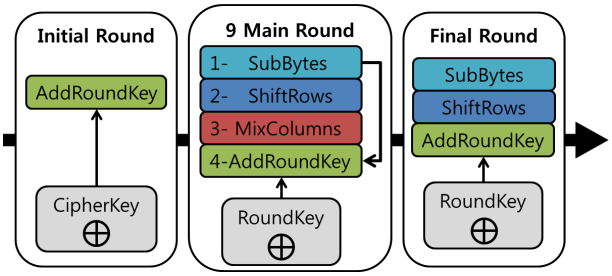

그림 1 고급 암호화 알고리즘 개념도
Fig. 1 A conceptual diagram of the AES Algorithm

고리즘이다. AES의 암호화 방식은 그림 1과 같다. AES는 1번의 초기 라운드, 9번의 메인 라운드와 1번의 마지막 라운드를 수행하며 총 11번의 암호화 라운드를 거쳐 평문을 암호문으로 암호화한다. 초기 라운드를 제외한 10개 라운드는 각각의 라운드 키에 의해 암호화가 이루어진다. AES의 암호화 라운드는 SubBytes, ShiftRows, MixColumn, AddRoundKey의 총 4개의 부분으로 이루어져 있다. 초기 라운드를 제외한 10라운드의 암호화에 사용되는 라운드 키는 RotWord, SubBytes, Xor의 3개 과정을 수행하여 생성된다. 특이 사항으로는, 초기라운드는 암호키와 평문의 AddRoundKey 과정만 수행하며, 다음 9번의 메인 라운드는 SubBytes, ShiftRows, MixColumn, AddRoundKey 과정이 차례로 수행하고, 마지막 라운드는 MixColumn 과정을 제외한 SubBytes, ShiftRows, AddRoundKey 과정을 수행한다.

AES는 파일의 암호화, 통신의 암호화 등 다방면에서 사용된다. 예를 들어, IEEE 802.11i에서 정의한 WPA2는 AES의 한 종류인 AES-CCM(Counter with CBC-MAC) 방식을 사용한다[7]. 이처럼 AES는 다양한 영역에 걸쳐 전반적으로 사용되고 있으며, 최근에는 암호강도를 높이기 위해 256bits의 암호키를 이용하여 14회의 암호화 라운드를 수행하는 AES-256도 사용되고 있다. 특히나 AES는 하드웨어 구현에 최적화된 암호화 알고리즘이기 때문에, 이를 FPGA상에 구현하는 것이 바람직하다[8].

2.2 FPGA

FPGA는 내부에 수많은 논리 게이트와 플립플롭, 메모리를 포함하는 논리 블록(Logic Block)과 프로그래밍으로 각 논리블록 사이를 연결하거나 입/출력단자와 연결할 수 있는 배선 영역(Interconnection Resources) 등으로 구성된 반도체이며, 개략적인 내부 구조는 그림 2와 같다[9]. 하드웨어 기술 언어인 HDL(Hardware Description Language)로 FPGA를 프로그래밍 함으로써 설계할 수 있다. 그리고 FPGA로 설계된 모듈을 주문형 반도체(Application Specific Integrated Circuit; ASIC)로 구현하면 더욱 성능을 향상시킬 수 있다. FPGA는 ASIC보다 느리고, 복잡한 설계를 할 수 없으나, 개발시간이 짧고, 오류를 바로 재수정 할 수 있으며, 초기 개발비가 저렴하다는 장점이 있다. 그 이유로 FPGA는 ASIC의 초기버전 개발 등에 사용된다[10].

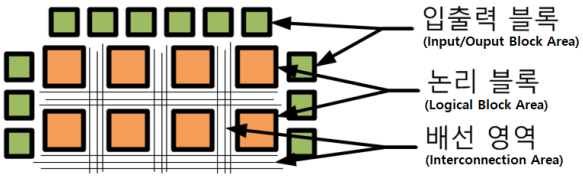

그림 2 FPGA의 내부 구조
Fig. 2 The internal structure of FPGA

3. AES 알고리즘의 하드웨어 구현

본 논문에서 AES 알고리즘을 FPGA에 구현하기 위해 Verilog HDL을 사용하였고, 하위 모듈들부터 구현하여 상위 모듈을 구현하는 Bottom-Up 설계 방식을 이용하였다. 그림 3은 본 논문에서 구현한 AES 알고리즘의 블록도이다. AES의 최상위 모듈은 AES_TOP으로 정의하였다. 입력은 동작주파수 50MHz를 갖는 클럭 동기신호 Clk, 동작신호 En, 암/복호화 선택신호 E/D, 128-bit 암호키 데이터버스 K, 128-bit 평문 데이터 버스 P이며, 출력은 128-bit 암호문 데이터 버스 C이다.

AES_TOP 모듈의 내부는 라운드 암호키를 생성하기 위한 KEY_GEN 모듈과 평문의 암호화를 위한 AES_CORE 모듈로 구성된다. AES_CORE 모듈의 내부에는 S_Box모듈을 파생해 S-Box로 각각의 바이트를 치환하는 SubBytes 기능을 구현 하였다. ShiftRows 기능은 S_Box의 출력을 wire로 출력과 동시에 바이트들의 위치를 바꾸게 함으로서 ShiftRows 기능을 구현하였다. MixColumn 기능은 유한체(Galois Field; GF)를 이용한 배열공식을 사용하여 해당 비

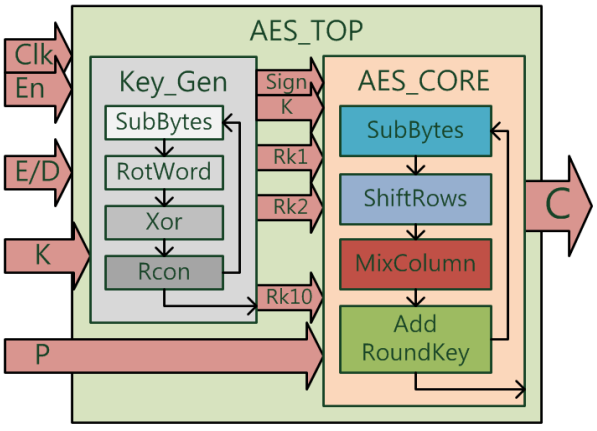

그림 3 구현된 AES 알고리즘의 블록도
Fig. 3 A block diagram of the implemented AES Algorithm

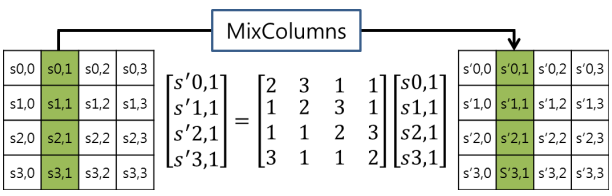

그림 4 MixColumns 기능 개념도
Fig. 4 A conceptual diagram of MixColums function

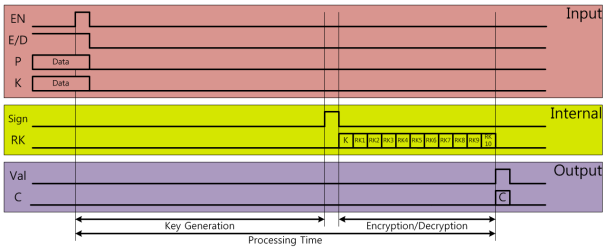

그림 5 구현된 AES 모듈의 타이밍 차트
Fig. 5 A timing chart of an implemented AES module

트의 곱을 각각의 모듈로써 파생하였다. GF를 이용한 배열 곱 공식은 그림 4와 같이 표현된다. AddRoundKey 기능은 KEY_GEN 모듈에서 생성된 라운드 암호키를 입력받아 Xor하는 것으로 구현하였다. KEY_GEN 모듈의 내부에는 S_Box 모듈을 파생해 각각의 바이트를 치환하는 SubBytes 기능을 구현 하였으며, 고정된 값인 Rcon은 레지스터를 이용해 구현함으로써 AES 알고리즘의 기능들을 구현하였다.

그림 5는 구현된 AES 모듈의 개략적인 타이밍 차트이다. 암호화는 En이 활성화 되자마자 시작되기 때문에, 그 전에 앞서서 암호키, 평문 그리고 암/복호화를 선택 신호들이 미리 준비되어 있어야 한다. En이 활성화되면, KEY_GEN 모듈이 먼저 입력된 암호키를 기반으로 한 암/복호화의 초기 라운드를 제외한 10라운드에 적용될 라운드 암호키를 생성한다. 전체 라운드 키를 생성된 후, KEY_GEN 모듈은 AES_CORE 모듈에 준비 완료 신호(Sign)를 보내고, 순차적으로 초기라운드 부터 10라운드까지 라운드 암호키들을 순차적으로 전달한다. 완료 신호 이후에 AES_CORE 모듈은 KEY_GEN 모듈로부터 전달받은 라운드 암호키와 평문을 이용하여 암호화를 수행하게 된다. 평문의 암호화가 완료되면 AES_CORE모듈은 출력 유효신호(Val)와 함께 암호문을 출력으로 내보낸다.

설계한 AES 모듈을 FPGA에 적용하고 기능 확인을 위해 Altera사의 DE0-Nano-SoC 보드 2개를 사용하여 실험하였다. 이 보드는 Cyclone V 계열의 FPGA를 사용한다. 실험에 사용한 동작주파수는 50MHz이고, 추가적인 구현 및 실험 환경의 구성을 위하여 10개의 평문과 4개의 암호문을 갖는 실험용 상위 모듈 AES_TEST을 구성하였다. 또한 상호 FPGA간의 통신을 통한 암/복호화 기능 확인을 위해 시리얼 통신모듈을 내장하였다. 추가로 실험의 편리성과 결과의 시각화를 통해 결과를 확인할 수 있도록 시각화 보드와 컨트롤 보드를 제작하여 실험 보드들의 GPIO핀에 연결하였다. 시각화 보드에는 도트 매트릭스 보드를 연결하여 평문과 암호문을 을 도트 매트릭스 상에 숫자로 보일 수 있도록 하였다. 컨트롤 보드에는 암호화키 선택과 En과 E/D와 같은 여러 신호 또한 조작할 수 있어 여러 가지 입력 상황들을 실험할 수 있도록 구성하였다. 완성된 실험 환경은 그림 6과 같다. 이는 2개의 FPGA 보드에서 작동하는 AES 알고리즘의 암복호화 기능 확인 실험한 모습이다. 좌상단의 도트매트릭스에 3이라고 이미지화 된 평문의 2진 데이터가 왼쪽의 FPGA 보드에서 암호화되어 디스플레이 된 뒤, 시리얼 통신을 통해 오른쪽의 FPGA에서 동일한 암호화키를 통하여 복호화되어 다시 평문으로써 우하단 도트메트릭스에 출력되는

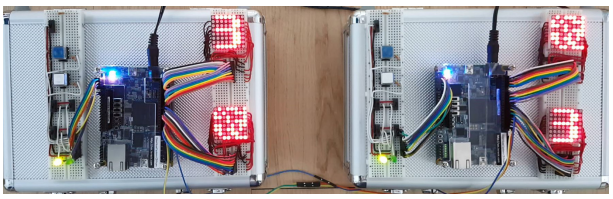

그림 6 AES 기능을 확인을 위한 실험 환경
Fig. 6 The test environment to verify AES functions

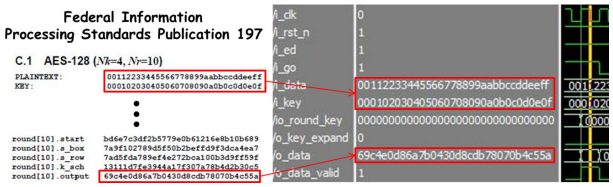

그림 7 AES 기능 확인을 위한 RTL 시뮬레이션
Fig. 7 An RTL simulation to verity AES functions

정상적으로 암호화와 복호화가 이루어지는 모습을 확인할 수 있다.

4. AES 기반 보안 칩의 성능 분석 및 평가

FPGA에 구현된 AES 기능을 확인을 위해 ModelSim 10.5b 프로그램으로 RTL(Resigter Transfer Level) 시뮬레이션을 수행하였다. 그림 7에서 볼 수 있듯이, FPGA에 구현한 AES는 FIPS-197상의 예시와 동일한 평문과 암호화키를 가지고 동일한 출력을 내고 있다. 이를 통하여 이론과 구현된 AES가 일치하며, 구현한 AES가 올바르게 작동하고 있음을 알 수 있다.

암호화 처리 시간은 암호화키 입력부터 암호화 완료 신호의 출력 간의 시간이다. 이 시간은 암호화 모듈의 처리주기(processing cycle)와 동작 클록(operation clock)과의 비로도 표현 가능하다. 이것은 암호화 모듈의 처리주기가 일정할 때, 동작 클록에 따라 암호화 처리 시간이 단축됨을 의미하며 이론상으로 고속으로 동작하는 시스템에서 더 좋은 성능을 낼 수 있음을 의미한다. 그림 8은 구현된 AES 모듈의 시뮬레이션 결과이다. 이 시뮬레이션은 동작 주파수가 100MHz이며, 출력 파형에서 클록 주기가 10ps임을 확인 할 수 있다. 그리고 암호화 시간은 시뮬레이션 상에서 600ps이며, 이 시간은 라운드 암호키 생성시간(470ps)과 평문을 암호화하는 시간(130ps)을 합한 시간이다. 즉 암호화에는 60clk이 소요된다. 이 수치를 단위 Byte당 Cycle 수로 표현하자면, 데이터가 16-Byte(=128-bit)이므로 3.75 Cycles/Byte를 갖는다. 따라서 구현된 AES 모듈이 시뮬레이션에서 올바르게 동작하고 있음을 확인하였고, FPGA 에뮬레이션에서도 동일하게 작동함을 그림 6을 통해서 알 수 있다.

테스트에 사용한 FPGA 보드의 내부 클록은 50MHz이다. 이 보드를 통해 FPGA 에뮬레이션하여 1초당 처리하는 데이터 크기(Throughput)을 계산하면 식 (1)과 같이 106.7Mbps를 갖는다.

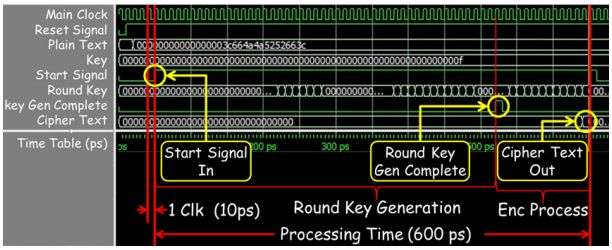

그림 8 구현한 AES 모듈의 타이밍 분석

Fig. 8 A timing analysis of the implemented AES

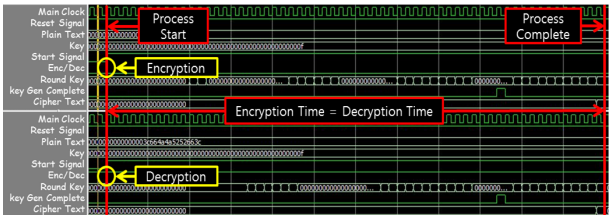

그림 9 구현한 AES 모듈의 암복호화 속도 비교

Fig. 9 A comparison of AES encryption and decryption time

$$Throughtput = \frac{\text{단위시간처리데이터량}}{\text{처리시간}} = \frac{128bit}{1200ps} = 106.7Mbps \quad (1)$$

즉 FPGA상에 구현된 AES는 50MHz 환경에서 초당 약 106.7Mbit의 속력으로 데이터를 처리함을 알 수 있다. 그리고 그림 9에서 보는 바와 같이, 구현된 AES 모듈은 암호화 속도와 복호화 속도가 동일하다. 이를 통상 PC의 속도인 2GHz에서도 동작한다고 가정한다면, 106.7Mbps의 40배의 속도로 암복호화 과정을 진행할 수 있게 된다.

그러나 고속의 시스템 환경에서는 클록 당 여유 시간(timing margin)이 점점 줄어들게 되므로, 각 내부 신호들의 전달 지연(propagation delay)이 여유 시간보다도 커 질 수 있으므로 구현된 AES 모듈이 정상적으로 동작하는지 장담할 수 없다. 따라서 구현한 AES가 최대 어느 정도의 클록에서 분석하기 위하여 Altera Quarters II 7.0을 이용한 타이밍 시뮬레이션(timing simulation)을 수행하였다. 이 때 사용한 디바이스는 Cyclone II EP2C70F896C8이다. 설계된 AES 모듈은 그림 10에서 볼 수 있듯이 13,517개의 조합논리회로와 2,330개의 플립플롭을 레지스터로써 합성되었다. 또한 9비트 곱셈기 등이 사용되지 않는 간단한 방법으로 구현되었음을 알 수 있다. 그림 11은 완료된 타이밍 시뮬레이션 결과를 보여준다. 타이밍 시뮬레이션에 따르면 가장 시간이 많이 걸릴 수 있는 최악의 케이스는 약 29.295ns의 전달지연시간이 걸림을 알 수 있다. 통상적으로 걸리는 시간은 약 22ns이며, 이 의미는 시뮬레이션에서 45MHz의 동작 주파수를 갖도록 권장하고 있다는 것이다.

즉, 이 AES는 Cyclone II EP2C70F896C8상에서 최대 45MHz까지만 동작할 수 있으며, 그 이상의 동작 주파수를 갖는 시스템에서는 정상 작동을 보장할 수 없다. 최악의 케이스까지 고려한다면 최대 허용 가능한 주파수는 더욱 낮아

Flow Status	Successful - Wed Dec 06 15:51:04 2017
Quartus II Version	7.1 Build 156 04/30/2007 SJ Web Edition
Revision Name	AES_SIM
Top-level Entity Name	AES_SIM
Family	Cyclone II
Device	EP2C70F896C8
Timing Models	Final
Met timing requirements	Yes
Total logic elements	13,517 / 68,416 (20 %)
Total combinational functions	13,372 / 68,416 (20 %)
Dedicated logic registers	2,330 / 68,416 (3 %)
Total registers	2330
Total pins	517 / 622 (83 %)
Total virtual pins	0
Total memory bits	0 / 1,152,000 (0 %)
Embedded Multiplier 9-bit elements	0 / 300 (0 %)
Total PLLs	0 / 4 (0 %)

그림 10 FPGA 컴파일 결과

Fig. 10 An FPGA compilation result of the AES module

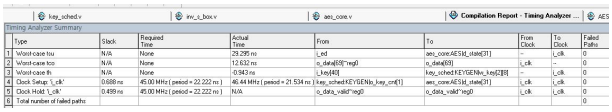

그림 11 타이밍 시뮬레이션 결과

Fig. 11 A result timing simulation of the AES module

질 것이다. 그러나 실제 50MHz 동작 클록인 FPGA상에서 AES가 올바르게 구현하는 것을 테스트 보드에서 확인 하였다. 이러한 시뮬레이션과의 차이는 Cyclone II와 Cyclone V 사이의 세대차이 및 그 동안의 기술의 발전에 의한 전파지연시간의 감소로 생각된다. 결론적으로, 구현한 AES는 Cyclone II EP2C70F896C8상에서 통상 최대 45MHz clk까지만 동작할 수 있으며, Cyclone V 5CSEMA4U23C6에서는 50MHz까지는 동작이 확인되었으나, 그 보다 빠른 클록에서는 정상동작을 보증할 수 없다.

한편, 표 1에서 볼 수 있듯이, PC에서 소프트웨어로 구현한 AES 모듈의 경우에는 단위 데이터(byte)당 소요되는 처리주기(cycles)는 짧은 경우라도 14.57 Cycles/Byte를 갖는다[11]. 본 논문에서 구현된 AES 모듈은 3.75 Cycles/Byte이므로 하드웨어로 구현된 결과가 소프트웨어로 구현한 결과보다 3.9에서 7.7배 빠른 성능을 갖는다. 이를 통하여 하드웨어로 구현한 AES가 소프트웨어로 구현한 AES보다 더 빠른 처리속도를 보임을 알 수 있다. 구현한 AES 모듈의 동작 주파수인 50MHz는 최근 사용되고 있는 사물인터넷 시스템에서 사용되는 MPU에는 충분하나 현재의 고성능 CPU의 시스템 클럭에 비하여 현저히 낮은 속도이다. 예를 들어, 2GHz의 동작 주파수를 갖는 PC상에서 소프트웨어기반 AES 모듈을 처리하면 227.7MByte/sec의 처리속도를 갖는다. 이때 사이클 당 명령어 처리 횟수를 16으로 가정한다[12]. 그러나 하드웨어기반 AES 모듈로 처리하게 된다면 23.4MByte/sec가 된다. 이러한 동작 차이는 소프트웨어기반 AES 모듈의 속도가 하드웨어 방식보다 빠르게 되는 역전현상을 일으킬 수 있다. 이에 고속의 임베디드 시스템에 적용시키기 위하여 전달 지연 시간을 줄일 수 있는 파이프라인(Pipeline)을 적용할 수 있을 것이다. 60개의 사이클 중 몇 개의 사이클이 전파지연이 큰 최악의 경우를 갖느냐에 따라 다르겠지만, 이론적으로 이러한 파이프라인 구현을 통해 획기적인 속도의 향상을 이룰 수 있을 것이다[13]. 그러므로

보안 칩 제작을 위해 구현된 하드웨어기반 AES 모듈은 소프트웨어 방식보다 단위 데이터당 소요되는 처리 주기가 적어 동일한 환경에서는 제안한 하드웨어 방식이 3.9에서 7.7배 빠른 성능을 보여주나, 전파지연의 문제로 인하여 통상 PC보다 속도가 뒤쳐진다. 그러나 이는 파이프라인의 적용을 통해 극복할 수 있다.

표 1 구현된 AES 모듈의 처리속도 비교

Table 1 A Comparison of processing speed of the implemented AES module

AES module	Measurement	Cycles/byte
This Paper	eSTREAM	3.75
Bernstein	eSTREAM	14.57
Gladman	eSTREAM	17.84
Wu	eSTREAM	26.74
OpenSSL 0.9.7e	openssl speed aes	29

5. 결 론

본 논문은 하드웨어 보안 칩을 FPGA상에 구현하는 방법을 소개하였다. 대표적인 보안 알고리즘인 AES를 Verilog HDL 언어로 설계하고 FPGA 합성 및 타이밍 시뮬레이션을 수행하였다. 그리고 보안 알고리즘의 성능평가를 위해 이를 소프트웨어로 구현된 AES와 비교하였다. 그 결과, 하드웨어 기반 AES 모듈이 소프트웨어로 구현된 방식보다 단위 데이터당 소요되는 처리주기가 3.9배 이상 빠르다는 것을 확인하였다. 그러나 이 차이는 하드웨어 설계의 특징 때문에 고속으로 동작하는 시스템에서는 처리속도의 역전이 발생할 수는 있다는 문제점이 있다. 이를 개선하기 위해서 전달 지연시간을 나눠서 처리하는 파이프라인 방식을 이용하여 극복이 가능하며, 병렬처리 또는 ASIC을 통한 구현과 같은 방법으로 그 차이를 극복할 수 있을 것이다. 뿐만 아니라, 소프트웨어로 보안 알고리즘을 구현하는 경우에는 다른 프로세스에게 CPU자원을 할당하는 시간이 있기 때문에 최대의 속도로 암호화를 진행할 수는 없으며, 이에 따라 암호화 속도의 저하 또한 불가피할 것이다. 하지만 하드웨어로 구현한 AES는 입력만 지속된다면 CPU의 작업과는 상관없이 최대한의 속도를 유지한 채 암호화를 지속할 수 있다. 결론적으로 하드웨어를 통하여 구현한 AES보안 알고리즘이 소프트웨어로 구현한 AES보다 안정적인 고속으로 암호화를 할 수 있음을 알 수 있다.

이 논문을 통하여, 자원이 제약되는 사물인터넷, 임베디드 시스템과 같은 환경에서는 하드웨어로 구현한 보안 시스템이 속도와 효율적인 측면에서 더 효과적임을 알 수 있었다. 최근 들어 유명 회사의 전자제품이 보안 칩을 내장하여 암호화 기능을 갖는 것은 하드웨어를 통해 보안 기능을 구현하는 보안 칩 시장이 더욱 성장할 것임을 암시하는 것일 것이다. FPGA, ASIC과 같은 하드웨어로 구현된 보안 알고리즘은 앞서 언급한 파이프라인과 병렬처리 등을 통해 더욱 빠른 속도로 암호화가 가능할 것으로 예상된다. 그리고 AES 알고리즘이 아닌 IPSec과 같은 다양한 알고리즘들도 하드웨어에도 적용 가능하므로 설계를 IP(Intellectual property)화하여 적용해 놓는다면 미래의 사물 인터넷관련 제품들은 시간적 어려움 없이 보안 강도를 한층 향상 시킬 수 있을 것이다.

감사의 글

본 연구는 2018년도 상명대학교 교내연구비를 지원받아 수행하였음.

References

[1] E. Brown, “Who Needs the Internet of Things?,” linux.com, Oct. 2016.

[2] Verizon, “State of the Market : Internet of Things 2017,” Verizon, New York, 2017.

[3] M. Hossain et al., “Towards an analysis of security issues, challenges, and open problems in the internet of things,” *in 2015 IEEE World Congr. on Services*, 2015.

[4] M. Popa et al., “Privacy and Security in Connected Vehicles Ecosystems,” *Informatica Economica 21.4*, pp. 29-40, 2017.

[5] C. Wootton, “Samsung ARTIK Reference: The Definitive Developers Guide,” *Apress*, 2016.

[6] Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, Nov. 2001.

[7] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 6: Medium Access Control (MAC) Security Enhancements, IEEE Std 802.11i-2004, 2004.

[8] J. Daemen and V. Rijmen, “AES proposal: Rijndael,” 1999.

[9] J. Daintith and E. Wright, “A Dictionary of Computing (6ed),” Oxford University Press, 2008.

[10] I. Kuon and J. Rose, “Measuring the gap between FPGAs and ASICs,” *in Proc. 2006 ACM/SIGDA 14th Int. Symp. on Field Programmable Gate Arrays*, Monterey, California, 2006.

[11] D. J. Bernstein and P. Schwabe, “New AES Software Speed Records,” *in 9th Int. Conf. on Cryptology*, Kharagpur, India, pp. 322-336, 2008.

[12] D. A. Patterson and J. L. Hennessy, “Computer Organization and Design MIPS Edition: The Hardware/Software Interface,” Newnes, 2013.

[13] T. Rahman et al., “Design of a High Throughput 128-bit AES (Rijndael Block Cipher),” *in Proc. Int. Multi Conf. Engineers and Comput. Scientists*, Hong Kong, 2010.

저 자 소 개

박 현 근 (Hyeon-Keun Park)

2018년 상명대학교 정보보호공학과 졸업(학사).
주관심분야 : 정보보안, 임베디드시스템, 사물인터넷
E-mail : gudrmd2@naver.com

이 광 재 (Kwangjae Lee)

2014년 고려대학교 전자컴퓨터공학과 졸업(공박). 2012~2017년 전자부품연구원 정보통신미디어본부 연구원. 현재 상명대학교 정보보안공학과 조교수.
E-mail : begleam@smu.ac.kr