AES-128 크립토 코어의 경량화 구현

배기철* · 신경욱

*금오공과대학교

A Lightweight Implementation of AES-128 Crypto-Core

Gi-Chur Bae* · Kyung-Wook Shin

*Kumoh National Institute of Technology

E-mail : bae921216@kumoh.ac.kr

요 약

128-비트의 마스터 키를 지원하는 블록암호 AES-128을 IoT 보안에 적합하도록 경량화하여 구현하였다. 키 스케줄러와 라운드 블록을 8 비트 데이터 패스로 구현하고, 다양한 최적화 방법을 적용함으로써 하드웨어를 최소화시켰으며, 100 MHz 클록 주파수에서 4,400 GE의 작은 게이트로 구현되었다. Verilog HDL로 설계된 AES 크립토 코어를 Vertex5 XC5VSX50T FPGA 디바이스에 구현하여 올바로 동작함을 확인하였다.

키워드

AES, Lightweight Cryptography, Block Cipher, Security

Ⅰ. 서 론

사물인터넷(Internet of Things)이란 센서가 부착된 다양한 기기들이 실시간으로 정보를 인터넷을 통하여 공유하며, 공유된 정보를 활용하여 스스로 판단, 작동하는 시스템을 말한다. 사물인터넷은 시대의 아이콘으로 부상하였으며, 이러한 흐름을 증명하듯 여러 분야의 첨단기기에는 사물인터넷이 적용되고 있다. 그 중에는 스마트 자동차, 스마트 카드 등과 같이 보안위협에 노출될 시 큰 손실을 야기할 수 있는 분야들도 포함되어 있다. 사물인터넷에 가해지는 보안위협은 비인가 사용자가 불법으로 정보에 접근하는 가용성 침해와 접근한 정보를 조작 및 탈취하는 기밀성/무결성 공격, 사생활 침해 등이 대표적이다[1-2].

사물인터넷의 보안을 위해서는 대칭키 블록암호 방식 또는 공개키 암호 방식이 사용되며, 하드웨어와 소프트웨어 자원이 제한되는 RFID 태그, 스마트카드, 센서 노드 및 네트워크 등 단말과 네트워크의 보안을 위해서는 경량화 구현이 가능한 블록암호 알고리즘이 요구된다[3-4].

본 논문에서는 다양한 분야의 정보보안을 위해 폭넓게 사용되고 있는 대표적인 블록암호 알고리즘 AES(Advanced Encryption Standard)를 저면적으로 구현하였다. 이를 위해 알고리즘의 연산순서 변경, 데이터 패스 축소, 저면적 구조의 연산블록 채택 등 여러 가지 기법들을 적용하였다. Ⅱ장에서는 블록암호 알고리즘 AES에 대해 간략히 소개하며, Ⅲ장에서는 AES-128 크립토 코어의 경량화 구현에 대해 설명한다. 설계된 AES 코어의 기능검증 및 FPGA 구현 결과를 Ⅳ장에 기술하며, Ⅴ장에서 결론을 맺는다.

Ⅱ. 블록암호 알고리즘 AES [5]

AES는 미국 상무부 기술표준국(NIST)에 의해 선정된 블록암호 알고리즘이다. 128 비트의 블록 길이와 128/192/256 비트의 마스터 키 길이를 지원하는 AES는 non-Feistel 구조를 기반으로한 대칭키 방식의 블록암호이며, 마스터 키의 길이에 따라서 10/12/14회의 라운드 변환을 수행한다. AES의 라운드 변환은 초기 라운드 키 가산, 반복 라운드 변환, 최종 라운드 변환의 과정으로 이루어지며, 암호화와 복호화에 동일하게 적용된다.

라운드변환은 SubByte, ShiftRow, MixColumn, KeyAdd의 연산으로 구성된다. SubByte 연산은 state를 구성하는 각 바이트에 대해 서로 독립적인 선형치환을 수행한다. 치환에 사용되는 테이블을 S-Box라고 하며, S-Box는 유한체 $GF(2^8)$에서 곱셈의 역원을 취하는 $X \rightarrow X^{-1}$ 매핑과 $GF(2)$에서의 Affine 변환으로 이루어진다. ShiftRow는 별도의 연산과정 없이 state를 구성하는 데이터의 각 행을 정해진 오프셋만큼 바이트 단위로 좌측 순

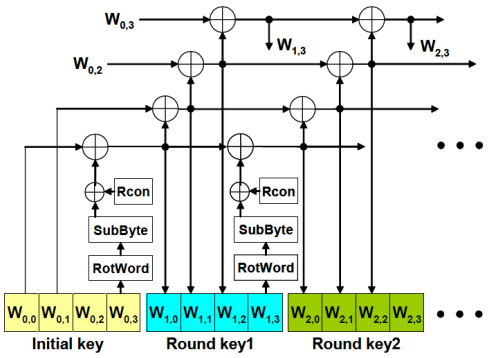

Fig. 1. Round Key Expansion of AES

환이동 시키는 연산이다. MixColumn은 state의 열을 정해진 행렬 테이블에 대입하여 행렬 곱셈을 수행하는 연산이며, 행렬 곱셈 연산은 유한체 $GF(2^8)$에서 이루어진다.

AES의 라운드 변환에서 사용되는 라운드 키는 그림 1의 키 확장에 의해 생성되며, 키 확장은 RotWord, SubByte, RconAdd와 XOR 연산으로 이루어진다. RotWord는 32 비트의 워드를 1 바이트 좌측 순환이동시키는 연산이다. RconAdd는 라운드에 따라 정해진 상수 값을 첫 번째 바이트에 가산하는 연산이다. $i-1$번째 라운드 키로 부터 i번째 라운드 키를 생성하기 위해서는 $i-1$번째 라운드 키의 네 번째 워드인 $W_{i-1,3}$에 대해 RotWord, SubByte, RconAdd 연산을 수행한 후, $i-1$번째 라운드 키의 첫 번째 워드 $W_{i-1,0}$와 XOR 연산을 진행하여 i번째 라운드 키의 첫 번째 워드 $W_{i,0}$가 생성되며, 이를 토대로 $W_{i,x+1}$와 $W_{i-1,x}$를 XOR 연산하면 i번째 라운드 키의 나머지 워드를 생성할 수 있다.

III. AES-128 크립토 코어 설계

128 비트의 마스터 키를 지원하며, 128 비트의 평문/암호문을 암호화/복호화하여 암호문/복호문을 생성하는 AES-128 크립토 코어를 설계하였다. 하드웨어 복잡도를 줄이기 위하여 라운드 블록과 키 확장 블록의 데이터 패스를 8-비트로 설계하였으며, 전체적인 구조는 그림 2와 같다. 암호화와 복호화의 라운드 연산순서 차이에 의해 유발되는 하드웨어 구현의 비효율성을 개선할 수 있도록 ShiftRow와 SubByte 연산의 순서를 바꾸어 최적화하였다.

라운드 변환을 구성하는 SubByte, MixColumn 연산 회로를 저면적 구현에 적합하도록 최적화하여 설계하였다. SubByte와 InvSubByte 블록을 LUT로 구현하는 경우에는 하드웨어 크기가 커지게 되고, 암호화와 복호화의 하드웨어 공유가 불가능하다. 본 논문에서는 유한체 $GF(2^8)$에서 곱셈의 역원(multiplicative inverse) 회로를 사용하여

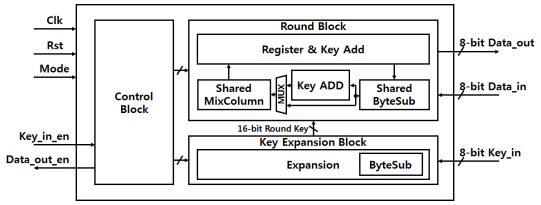

Fig. 2. Architecture of AES-128 Crypto-Core

암호화와 복호화 과정에서 하드웨어가 공유되도록 하였으며, 곱셈의 역원은 유한체 $GF(((2^2)^2)^2)$로 분해하여 구현함으로써 회로 복잡도가 최소화되도록 설계하였다.

MixColumn은 4×4 바이트의 매트릭스와 4 바이트 state의 곱셈이므로 다수의 상수 곱셈기와 XOR 게이트가 필요하다. 본 논문에서는 경량화 구현을 위해 MixColumn 연산을 8 비트 단위로 분할하여 연산되도록 설계하였으며, 이를 위해서 32 비트 레지스터가 추가로 필요하지만 전체적으로 저면적으로 구현되었다.

라운드 블록이 8 비트 데이터 패스로 구현되었으므로, 라운드 키도 8 비트 단위로 확장하면 키 스케줄러의 하드웨어 최소화할 수 있다. 본 논문에서는 키 스케줄러를 8 비트 데이터 패스로 구현하였으며, 라운드 블록과 키 확장 연산, 그리고 데이터 흐름에 대한 면밀한 분석을 통해 키 확장 연산이 최소의 회로로 구현되도록 하였다. 그 중 한 가지를 예로 든다면, 복호화 키 확장을 8 비트 데이터 패스로 처리하기 위해서는 추가적인 32 비트의 레지스터가 필요하지만, 키 확장 연산의 과 동작 타이밍 특성을 반영하여 24 비트 XOR 게이트로 구현하였으며, 하드웨어 복잡도를 최소화하였다.

IV. 기능 검증 및 FPGA 구현 결과

Verilog HDL로 설계된 AES-128 크립토 코어의 기능검증 결과는 그림 3과 같다. 128 비트의 평문(암호문)과 마스터 키가 8 비트 단위로 16 클록 주기에 걸쳐 입력된 후, 암호화(복호화) 연산에 의해 암호문(복호문)이 출력되며, 설계된 AES-128 크립토 코어의 논리 기능이 올바로 작동함을 확인하였다.

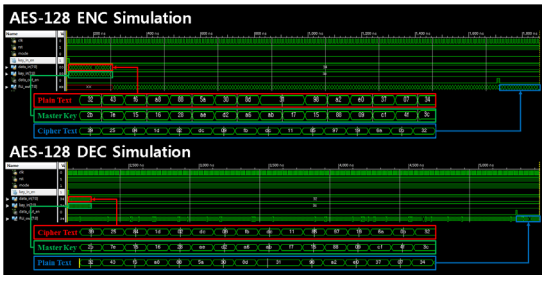

Fig. 3. Simulation results of AES-128 Crypto-Core.

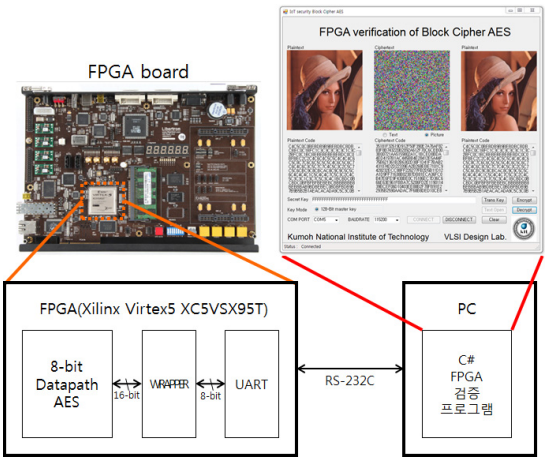

Fig. 4. FPGA verification of AES-128 Crypto-Core.

기능검증이 완료된 코어는 FPGA 구현을 통하여 하드웨어 동작을 검증하였다. FPGA 보드, UART 인터페이스, GUI 소프트웨어로 구성된 검증 시스템은 그림 4와 같으며, FPGA 디바이스는 Xilinx Virtex5 XC5VSX50T가 사용되었다. PC에서 입력된 평문(암호문)과 마스터 키는 RS232C 통신을 통해 FPGA로 전달되고, 암호화(복호화)된 결과는 다시 PC로 전송되어 GUI 소프트웨어에 의해 화면에 표시된다. Lena 이미지를 암호화한 결과로 랜덤 값과 유사한 암호화된 이미지가 출력되고, 이를 다시 복호화 하면 최초의 Lena 이미지가 출력되어 FPGA에 구현된 AES-128 크립토 코어가 올바로 동작함을 확인하였다.

설계된 AES-128 크립토 코어는 180nm CMOS Cell 라이브러리를 활용하여 합성한 결과, 최대 137 MHz 클록 주파수에서 동작이 가능하며, 최대 동작 주파수에서 5,700 GE, 100 MHz 클록 주파수에서 4,400 GE로 구현되었다.

V. 결 론

정보보안을 위해 가장 널리 사용되고 있는 블록암호 알고리즘 AES를 8 비트 데이터 패스의 경량화 하드웨어로 구현하고, FPGA 검증을 통해 정상 동작함을 확인하였다. 설계된 AES 크립토 코어는 100 MHz 클록 주파수에서 4,400 GE로 구현되어 경량화가 요구되는 IoT 보안에 매우 적합한 것으로 평가된다.

감사의 글

· 산업통상자원부 출연금으로 수행한 산업핵심기술개발사업(1004 9009, 사물인터넷 기반 영상보안용 초저전력 SoC 핵심 IP 기술 개발)의 지원을 받았음.

· 반도체설계교육센터(IDEC)의 CAD Tool 지원에 감사드립니다.

참고문헌

[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/cse57414/ftp/security/

[3] 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.

[4] 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.

[5] Rijmen, Vincent and Joan Daemen, Advanced encryption standard, *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology (2001)*, 19-22, 2001.