ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 19-02-07

j.inst.Korean.electr.electron.eng.Vol.23,No.2,388~394,June 2019
http://dx.doi.org/10.7471/ikeee.2019.23.2.388

블록암호와 해시 함수 IP가 내장된 Cortex-M0 기반의 보안 시스템 온 칩

A Cortex-M0 based Security System-on-Chip Embedded with Block Ciphers and Hash Function IP

최 준 영*, 최 준 백* 신 경 욱*★

Jun-Yeong Choe*, Jun-Baek Choi*, Kyung-Wook Shin*★

Abstract

This paper describes a design of security system-on-chip (SoC) that integrates a Cortex-M0 CPU with an AAW (ARIA-AES- Whirlpool) crypto-core which implements two block cipher algorithms of ARIA and AES and a hash function Whirlpool into an unified hardware architecture. The AAW crypto-core was implemented in a small area through hardware sharing based on algorithmic characteristics of ARIA, AES and Whirlpool, and it supports key sizes of 128-bit and 256-bit. The designed security SoC was implemented on FPGA device and verified by hardware-software co-operation. The AAW crypto-core occupied 5,911 slices, and the AHB_Slave including the AAW crypto-core was implemented with 6,366 slices. The maximum clock frequency of the AHB_Slave was estimated at 36 MHz, the estimated throughputs of the ARIA-128 and the AES-128 was 83 Mbps and 78 Mbps respectively, and the throughput of the Whirlpool hash function of 512-bit block was 156 Mbps.

요 약

블록암호 알고리듬 ARIA와 AES 그리고 해시 함수 Whirlpool을 단일 하드웨어로 통합 구현한 AAW(ARIA- AES-Whirlpool) 크립토 코어를 Cortex-M0 CPU에 슬레이브로 인터페이스한 보안 SoC(System-on-Chip) 설계에 대해 기술한다. AAW 크립토 코어는 ARIA, AES, Whirlpool의 알고리듬 특성을 이용한 하드웨어 공유를 통해 저면적으로 구현되었으며, 128-비트와 256-비트의 키 길이를 지원한다. 설계된 보안 SoC 프로토타입을 FPGA 디바이스에 구현하고, 하드웨어-소프트웨어 통합 검증을 하였다. AAW 크립토 코어는 5,911 슬라이스로 구현이 되었으며, AAW 크립토 코어가 포함된 AHB_Slave는 6,366 슬라이스로 구현되었다. AHB_Slave의 최대 동작 주파수는 36 MHz로 예측되었으며, ARIA-128, AES-128의 데이터 처리율은 각각 83 Mbps, 78 Mbps이고, Whirlpool 해시 함수의 512-비트 블록의 처리율은 156 Mbps로 평가되었다.

Key words : Security SoC, Cortex-M0, ARIA, AES, Whirlpool hash function

---

* School of Electronic Engineering, Kumoh National Institute of Technology
★ Corresponding author
E-mail : kwshin@kumoh.ac.kr, Tel : +82-54-478-7427
※ Acknowledgment
・This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2017R1D1A3B03031677)
・This work was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korean government (Ministry of Trade, Industry & Energy, HRD Program for Software- SoC convergence) (No. N0001883)
・Authors are thankful to IDEC for supporting EDA software.
Manuscript received Jun. 16, 2019; revised Mar. 12, 2019; accepted Jun. 17, 2019.

Ⅰ. 서론

정보통신기술의 급속한 발전에 힘입어 다양한 사물들이 사물인터넷(Internet of Things)을 통한 네트워크로 연결되어 사람과 사물, 사물과 사물 간에 정보를 교류하는 지능형 인프라 및 서비스가 보편화되고 있다. 이와 같은 초연결 사회에서는 네트워크에 연결된 장치에 저장되고, 네트워크를 통해 유통되는 정보를 불법적인 유출, 위조 및 변조로부터 보호하고, 또한 정당한 사용자를 인증하는 등의 정보보안이 매우 중요한 요소가 된다[1]. 정보보안은 데이터 암호화, 인증, 전자서명, 키 관리 등을 포함하는 다양한 기술들을 기반으로 하며, 소프트웨어나 전용 하드웨어 또는 하드웨어와 소프트웨어의 혼합 방식으로 구현된다. 물리적인 안전성과 저전력 소모가 중요한 응용분야 또는 대량의 데이터에 대한 고속 실시간 처리가 필요한 경우에서는 보안 알고리듬을 전용 하드웨어로 구현하는 방법이 사용된다.

특히, IoT 네트워크 및 단말, 무선 스마트 단말기, 자율주행 자동차, 드론 등과 같이 제한된 하드웨어 및 소프트웨어 자원을 가지면서 다양한 보안 프로토콜의 구현이 필요한 분야에서는 보안 하드웨어 IP와 소프트웨어를 결합하여 구현할 수 있는 보안 SoC(System-on- Chip)가 핵심 컴포넌트로 부각되고 있으며, 이에 대한 연구와 개발이 활발하게 이루어지고 있다. 일반적으로, 보안 SoC는 대칭키(symmetric-key) 암호 코어, 공개키(public-key) 암호 코어, 해시(hash) 함수 코어, TRNG(True Random Number Generator) 등의 하드웨어 IP(intellectual property)가 CPU에 버스로 인터페이스 되며, 데이터의 암호・복호, 전자서명, 키관리, 인증 및 무결성 검증 등의 다양한 보안 프로토콜이 하드웨어-소프트웨어 통합으로 구현된다[2-5].

본 논문에서는 블록암호와 경량 해시 함수가 통합 구현된 AAW(ARIA-AES-Whirlpool) 크립토 코어 IP를 Cortex-M0에 슬레이브로 인터페이스된 보안 SoC 프로토타입 구현에 대해 기술한다. Ⅱ장에서는 블록암호 국내 및 국제 표준인 ARIA, AES 알고리듬과 경량 해시 함수 Whirlpool에 대해 소개하고, Ⅲ장에서는 Cortex-M0 기반의 보안 SoC 설계에 대해 설명한다. Ⅳ장에서는 설계된 보안 SoC의 BFM 시뮬레이션 및 FPGA 검증에 대해 기술하고, Ⅴ장에서 결론을 맺는다.

Ⅱ. ARIA, AES 블록암호 및 Whirlpool 해시 함수

1. ARIA 블록암호[6]

ARIA(Academy, Research Institute, Agency)는 128 비트의 평문/암호문을 암호화/복호화 하여 128 비트의 암호문/복호문을 만드는 대칭키 블록암호이다. 128/192/256 비트의 세 가지 키 길이를 지원하며, 키 길이에 따라 12/14/16회의 라운드 변환이 진행된다. 라운드 변환은 라운드 키 가산, 치환(substitution) 계층, 확산 (diffusion) 계층의 연산으로 구성된다. 홀수 라운드의 변환함수와 짝수 라운드의 변환함수에 각기 다른 치환계층이 사용되며, 최종 라운드의 변환함수에는 확산계층이 라운드 키 가산으로 대체된다. ARIA는 ISPN(Involution Substitution-Permutation Network) 구조를 가지므로 암호화와 복호화 과정이 동일하며, 단지 라운드 키만 다르다. 키 스케줄러는 키 초기화 과정과 라운드 키 생성 과정으로 구성된다. 키 초기화 과정에서는 3라운드의 Feistel 구조를 이용하여 마스터키로부터 4개의 128 비트 초기화키 값을 생성하고, 이 초기화키 값들은 라운드키 생성 과정에 사용된다. 키 길이에 따라 라운드 변환이 12/14/16회 만큼 진행되고 최종 라운드에서 키 가산이 두 번 이루어지므로, 총 13/15/17개의 라운드 키가 생성된다.

2. AES 블록 암호[7]

AES(Advanced Encryption Standard) 알고리듬은 128 비트의 평문/암호문을 암호화/복호화 하여 128 비트의 암호문/복호문을 만드는 대칭키 블록암호이다. 128/192/256 비트의 세 가지 키 길이를 지원하며, 키 길이에 따라 10/12/14회의 라운드 변환이 진행된다. 암호화 라운드 변환은 초기 라운드 키 가산(AddRoundKey) 후, SubByte, ShiftRow, MixColumn, AddRoundKey 연산으로 구성되며, 마지막 라운드에는 MixColumn 연산이 생략된다. 복호화는 암호화에 사용된 함수의 역변환인 InvSubByte, InvShiftRow, InvMixColumn이 사용된다.

3. Whirlpool 해시 함수[8]

Whirlpool은 ISO/IEC 10118-3 표준으로 채택된

경량 해시 함수이며, 임의의 길이(최대 2^{256} 비트)의 메시지를 512 비트의 메시지 다이제스트(message digest)로 변환한다. AES와 유사한 non-Feistel SPN 구조의 블록암호가 압축함수로 사용되며, 입력 메시지를 256 비트의 홀수 배가 되도록 만드는 메시지 패딩(message padding) 전처리가 필요하다. 라운드 변환은 초기 키 가산 후, SubBytes, ShiftColumn, MixRows, KeyAdd 연산으로 구성되는 라운드 변환이 9회 반복되며, 마지막 라운드 변환은 SubBytes, ShiftColumn, KeyAdd 연산으로 구성된다. 각 데이터 블록의 라운드 변환 결과 값과 해당 블록의 입력 데이터 그리고 직전 블록의 암호키가 XOR 연산되어 다음 데이터 블록의 암호키로 사용되며, 이와 같은 연산이 모든 데이터 블록들에 대해 반복되어 메시지 다이제스트가 생성된다.

III. Cortex-M0 기반의 보안 SoC 설계

Cortex-M0는 2만 게이트 정도로 구현이 가능하여 저면적과 저성능을 필요로 하는 IoT(internet of things)와 모바일 응용분야에 적합한 경량 MCU 코어이다[9]. IoT, 무선 센서 네트워크, 모바일 분야의 보안을 위해서는 데이터의 기밀성과 함께 인증, 전자서명 등 다양한 보안 프로토콜의 구현이 필요하며, 이를 위해서는 MCU와 보안 IP가 단일 칩에 집적된 보안 SoC의 경량 하드웨어 구현이 필요하다. 본 논문에서는 IoT와 같이 요구되는 성능은 낮으면서 저면적 구현이 필요한 응용분야에 적합하도록 Cortex-M0 기반의 보안 SoC 프로토타입을 구현하였다. 설계된 보안 SoC는 그림 1과 같이 AAW_Slave가 AHB를 통해 Cortex-M0에 연결된 구조이며, AAW_Slave는 AHB 버스 프로토콜을 통해 Cortex-M0와 데이터를 송・수신한다.

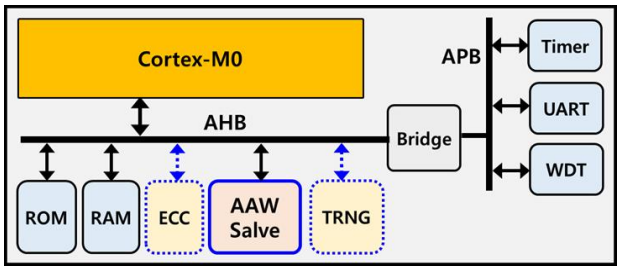


Fig. 1. Architecture of the Cortex-M0 based SoC.
그림 1. Cortex-M0 기반의 SoC 구조

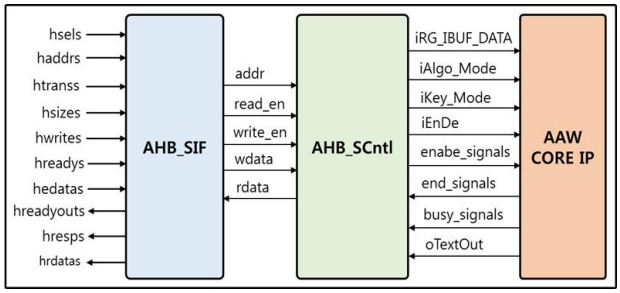


Fig. 2. AAW_Slave module.
그림 2. AAW_Slave 모듈

1. AAW_Slave

AAW_Slave는 그림 2와 같이 AHB 슬레이브 인터페이스 모듈(AHB_SIF), AHB 슬레이브 컨트롤 모듈(AHB_SCntl) 그리고 AAW 코어 IP로 구성되며, AHB 버스에 인터페이스 되어 Cortex- M0와 데이터를 주고받는다. AHB_SIF는 AHB 프로토콜을 받아 동일한 사이클에서 주소와 데이터를 활성화시키며, AHB_SCntl로 입력되는 읽기/쓰기 신호를 만들어 낸다. AHB_SCntl은 AHB_SIF로부터 받은 데이터를 AAW 코어 IP의 입력 형식에 맞게 변환하여 보내준다. AAW IP의 출력은 AHB_SCntl 내부의 버퍼에 저장된 후, AHB_SIF를 거쳐 Cortex-M0로 출력된다.

AHB_SCntl의 내부 구성도는 그림 3과 같으며, AAW 코어의 동작을 제어하는 AAW_Cntl, 읽기 레지스터 RD_Reg, 쓰기 레지스터 WR_Reg로 구성된다. AAW_Cntl 블록은 컨트롤 레지스터 값을 해석하여 AAW 코어 IP의 동작에 필요한 제어신호를 생성하며, AHB-lite 버스 프로토콜을 통해 입력되는 데이터를 키 길이, 메시지 길이, 평문 또는 암호문으로 구별하여 AAW 코어 IP에 입력하는 역할을 한다. Reg_map은 544-비트(32-b x 17)의

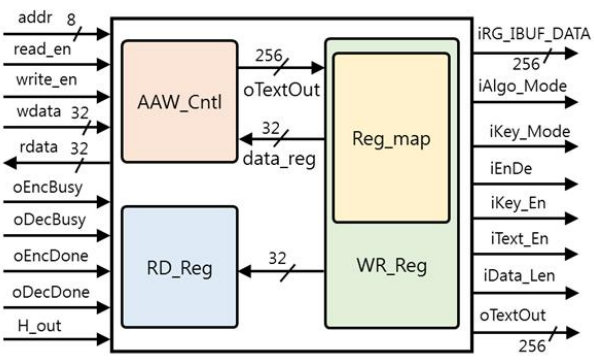


Fig. 3. AHB_SCntl block.
그림 3. AHB_SCntl 블록

레지스터로 구성되며, 주소 0번지는 제어 레지스터 ctrl_reg로 사용되고, 주소 1번지~8번지는 ARIA 또는 AES의 암호/복호를 위한 키, 평문/암호문 데이터의 버퍼로 사용되며, 주소 1번지~16번지는 Whirlpool 해시함수를 위한 메시지, 메시지 길이, 메시지 다이제스트 데이터의 버퍼로 사용된다. ARIA, AES 및 Whirlpool의 동작에 사용되는 데이터의 저장 공간이 공유되도록 설계하여 하드웨어 면적을 최소화 하였다.

WR_Reg는 Cortex-M0로부터 쓰기신호가 활성화되었을 때 입력되는 32-비트의 데이터, 그리고 AAW 코어로부터 출력되는 데이터를 Reg_map의 버퍼에 저장하는 역할을 한다. RD_Reg는 Cortex-M0로부터 읽기신호가 활성화되었을 때 AAW 코어의 출력을 AHB_SIF를 거쳐 Cortex-M0로 보낸다. 출력되는 데이터는 레지스터 ctrl_reg의 값에 따라 ARIA 또는 AES의 암호문/복호문이거나 Whirlpool의 메시지 다이제스트이다.

AAW 코어 IP의 동작모드와 그에 따른 제어신호의 생성 및 입력되는 데이터의 구별을 위해 컨트롤 레지스터 ctrl_reg는 표 1과 같이 설정된다. 키, 평문/암호문, 키 길이에 관한 제어신호를 AAW 코어 IP로 보낼 때 동작모드 신호가 입력되더라도 연산 중에는 동작모드가 변경되지 않도록 설계하여 오동작을 방지하였다.

2. AAW(ARIA-AES-Whirlpool) 코어 IP

ARIA, AES, Whirlpool 알고리듬의 특성을 이용하여 하드웨어 자원 공유가 극대화 되도록 설계하였다. 전체 구조는 그림 4와 같으며, Whirlpool 해시함수를 위한 패딩 블록(padder), ARIA, AES, Whirlpool 알고리듬의 통합 라운드 블록(round), 통합 라운드 키 생성블록(key_gen), 제어 블록으로 구성된다. 패딩 블록은 가변 길이의 메시지를 Whirlpool 해시함수에 적합하도록 512-비트의 고정 길이로 분할하고, 패딩하는 블록이며, 이때 패딩은 메시지 길이 데이터에 따라 그 방식이 변경된다. 통합 라운드 블록은 라운드 연산 중간 결과 값을 저장하는 512-비트 상태 레지스터 State_reg, 알고리듬 및 암호화/복호화 선택신호에 따라 치환계층 연산이 수행되는 통합 치환계층 블록(AAW-Sbox), 알고리듬에 따른 확산계층 연산이 수행되는 통합 확산계층 블록(AAW- Diff), 스테이트의 행과 열을 바이트 단위로 순환 시프트 하는 Shift-Row, Shift-Col 등으로 구성되며, 내부 데이터 패스는 128 비트로 설계되었다. 통합 치환계층 블록은 ARIA와 AES의 치환계층에 사용되는 S-box가 유한체에 기반을 두고 있기 때문에 S-box 연산 수행 시 이 부분을 공유하도록 설계되었다. 통합 확산계층은 XOR 블록(common XORs), ARIA 확산 연산 블록, AES 확산연산 블록, 그리고 Whirlpool 확산연산 블록으로 이루어져 있다. XOR 블록에서 ARIA, AES 및 Whirlpool의 확산함수에 공통으로 사용되는 XOR 항을 공유함으로서 하드웨어를 최소화하였다. 통합

Table 1. Control register setting for operation modes.
표 1. 동작 모드를 위한 컨트롤 레지스터 설정

ctrl_reg[31:0]	Operation modes	Enable signals generated		
0000_0000	ARIA-128 Enc	iAlgo_Mode = 0	iKey_Mode = 0	iEnDe = 0
0000_0010	ARIA-128 Dec			iEnDe = 1
0000_0008	ARIA-256 Enc		iKey_Mode = 1	iEnDe = 0
0000_0018	ARIA-256 Dec			iEnDe = 1
0000_0002	AES-128 Enc	iAlgo_Mode = 1	iKey_Mode = 0	iEnDe = 0
0000_0012	AES-128 Dec			iEnDe = 1
0000_000a	AES-256 Enc		iKey_Mode = 1	iEnDe = 0
0000_001a	AES-256 Dec			iEnDe = 1
0000_0004	WHIRLPOOL Hash	iAlgo_MODE = 2		
0000_0020	Key input enable	iKey_En		
0000_0040	Text input enable	iText_En		
0000_0080	Data length for WHIRLPOOL hash	iData_Len		

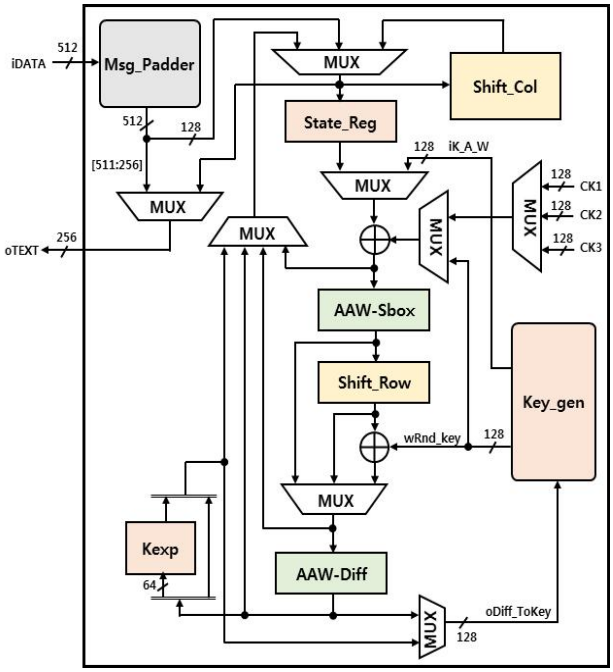
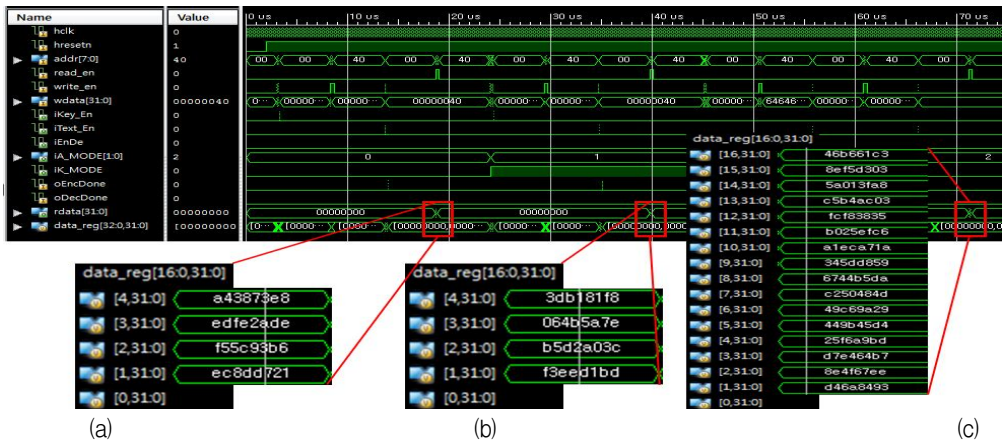


Fig. 4. Architecture of AAW core.
그림 4. AAW 코어의 구조



(a) (b) (c)

Fig. 5. BFM simulation results of AAW_Slave, (a) encryption mode of ARIA-128, (b) encryption mode of AES-256, (c) Whirlpool hash mode.

그림 5. AAW_Slave의 BFM 시뮬레이션 결과 (a) ARIA-128의 암호화 모드, (b) AES-256의 암호화 모드, (c) Whirlpool hash 모드

라운드키 생성 블록은 ARIA와 AES의 경우 128-비트와 256-비트 키 길이에 대한 키 초기화 과정과 라운드 키를 생성하며, Whirlpool의 경우 키 생성 과정과 라운드 연산과정이 동일하므로 라운드 함수 재사용 방식을 적용하여 라운드 연산과 키 생성이 시분할 방식으로 처리되도록 설계하여 하드웨어를 간소화하였다[10].

Ⅳ. BFM 시뮬레이션 및 FPGA 검증

1. BFM 시뮬레이션 검증

AAW 코어가 AHB 버스로 인터페이스 되어 AHB 프로토콜에 따라 정상동작 하는지 검증하기 위한 BFM(Bus Functional Model) 시뮬레이션을 진행하였다. 그림 5-(a)는 키 길이가 128-비트인 ARIA의 암호화 동작의 검증결과이며, 128-비트의 평문 “55555555 cccccccc 55555555 dddddddd”을 암호화한 결과로 “ec8dd721 f55c93b6 edfe2ade a43873e8”의 암호문이 출력되어 ARIA-128 암호 모드가 정상동작함을 확인하였다. 그림 5-(b)는 키 길이 256-비트인 AES의 암호화 동작을 검증한 결과이며, 평문 “6bc1bee2 2e409f96 e93d7e11 7393172a”을 암호화한 결과로 암호문 “f3eed1bd b5d2a03c 064b5a7e 3db181f8”이 출력되어 AES- 256 암호 모드가 올바로 동작함을 확인하였다. 그림 5-(c)는 Whirlpool 해시함수의 동작을 검증하기 위해 “aaaabbbb ccccdddd aaaabbbb ccccdddd aaaabbbb ccccdddd aaaabbbb ccccdddc aaaa”의 544-비트 메시지를 입력하여 512-비트의 메시지 다이제스트 “d46a8493 8e4f67ee d7e464b7 25f6a9bd 449b45d4 49c69a29 c250484d 6744b5da 345dd859 a1eca71a b025efc6 fcf83835 c5b4ac03 5a013fa8 8ef5d303 46b661c3”가 얻어져 Whirlpool 해시함수가 올바로 동작함을 확인하였다.

2. FPGA 검증

FPGA 검증을 위해 그림 6과 같이 Cyclone-V FPGA가 탑재된 V2M-MPS2 보드를 이용하여 UART 통신을 통해 PC와 데이터를 송·수신하는 FPGA 검증 플랫폼을 구성했다. Cortex-M0와 AAW_Slave로 구성된 보안 SoC를 Altera Quartus Prime를 이용하여 합성한 후, Tcl script 기능을 통해 V2M-MPS2 보드로 다운로드 하였다. FPGA에 구현된 AAW_Slave의 동작을 제어하기 위한 소프트웨어는 Keil uVision을 사용하여 크로스컴파일하였다. V2M-MPS2 보드의 JTAG 포트와 연결된 ULINK2를 uVision의 디버그 기능을 이용하여

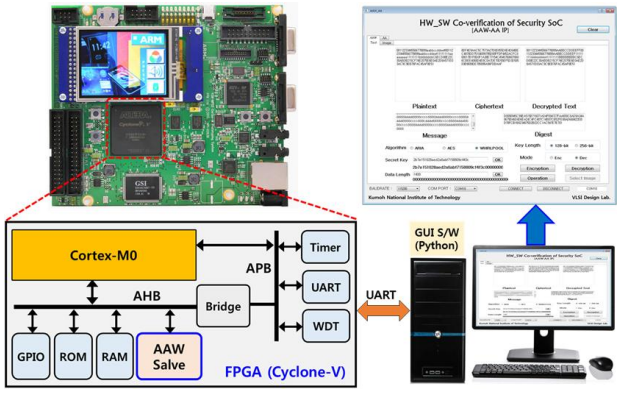
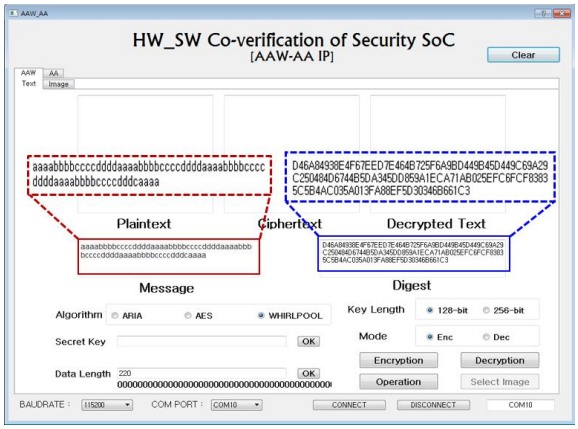


Fig. 6. FPGA verification setup

그림 6. FPGA 검증 시스템 구성

AAW_Slave의 입·출력을 확인하였다.

그림 7은 FPGA에 구현된 Cortex-M0 기반의 보안 SoC 동작을 검증한 결과이다. FPGA 검증을 위해 Python을 이용한 GUI 소프트웨어를 작성하여 사용하였다. 그림 7-(a)는 Whirlpool 동작모드의 FPGA 검증 결과이며, 데이터 길이와 임의의 길이의 메시지를 입력하여 AAW_Slave에서 얻어진 512-비트의 메시지 다이제스트를 GUI 화면에 출력한 결과를 보이고 있다. 그림 7-(b)는 128-비트의 키 길이에 대한 ARIA 암호, 복호 동작모드의 FPGA 검증 결과이다. 화면 좌측에 표시된 이미지의 픽셀 값들을 평문으로 하여 암호화한 결과가 화면 중앙에 암호화된 이미지로 표시되었다. 암호화된 이미지를 다시 복호화한 결과로, 화면 우측의 이미지가 얻어졌으며, 화면 좌측의 이미지와 동일하게 복원되었다. 이와 같은 FPGA 구현 검증을 통해, 본 논문에서 설계된 보안 SoC가 Cortex-M0에 구현된 소프트웨어와 연동하여 올바로 동작함을 확인하였다.



(a)

(b)

Fig. 7. FPGA verification results of the security SoC
(a) Whirlpool hash mode (b) ARIA-128 mode
그림 7. 설계된 보안 SoC의 FPGA 검증 결과
(a) Whirlpool hash 모드 (b) ARIA-128 모드

Table 2. Performance of the AHB_Slave.
표 2. AHB_Slave의 성능

Design	AHB_Slave Supporting AAW	AAW crypto IP
Algorithms supported	ARIA, AES, Whirlpool	
Key length supported [bits]	128, 256	
Modes of operation	ECB	
Cycles for processing a block	Data transfer via AHB BUS Interface: 38	ARIA(128/256): 17/21 AES(128/256): 21/29 Whirlpool: 80
Number of Slices [Spartan6 XC6SLX45]	6,366	5,911
Maximum frequency [MHz]	36	37
Throughput [Mbps] (@Max. Frequency)	ARIA(128/256): 83/78 AES(128/256): 78/68 Whirlpool: 156	ARIA(128/256): 278/225 AES(128/256): 225/163 Whirlpool: 236

표 2는 본 논문의 보안 SoC에 IP로 사용된 AAW 크립토 코어와 AHB_Slave를 Spartan-6 XC6SLX45 FPGA 디바이스로 합성한 결과이다. AAW 크립토 코어는 5,911 슬라이스로 구현이 되었으며, AAW 코어 IP가 포함된 AHB_Slave는 6,366 슬라이스로 구현되어 AAW 코어 IP의 AHB 인터페이스에 약 7.7%의 슬라이스가 추가로 사용되었다. AHB 버스 인터페이스를 통한 데이터 전송에 38 클록 사이클이 소요되며, AHB_ Slave의 최대 동작 주파수는 36 MHz로 예측되어 AAW 코어의 최대 동작 주파수 37 MHz와 비슷하다. AHB 인터페이스의 데이터 송수신에 소요되는 38 클록 사이클을 포함한 데이터 처리율은 ARIA-128, AES-128의 경우 각각 83 Mbps, 78 Mbps이고, Whirlpool 해시 함수의 512-비트 블록 처리율은 156 Mbps로 평가되었다.

V. 결론

본 논문에서는 ARIA, AES 대칭키 블록암호와 Whirlpool 해시 함수가 통합 구현된 AAW 크립토 코어를 AHB 버스를 통해 Cortex-M0에 슬레이브 인터페이스 하여 보안 SoC 프로토타입을 설계하였다. 설계된 보안 SoC를 Cyclone-V FPGA 디바이스에 구현하고, Cortex-M0에 구현된 소프트웨어와 연동하여 하드웨어-소프트웨어 통합 검증을 하였다. 설계된 보안 SoC 프로토타입은 6,366 슬라이스로 구현되었으며, 약 36 MHz의 최대 동작 주파수

를 갖는다. 본 논문의 보안 SoC 프로토타입에 ECC 공개키 암호 IP와 무작위 난수 발생기 IP를 추가하면, EC-DSA, ECIES 등 다양한 보안 프로토콜 구현에 활용될 수 있다.

References

[1] Ali Ismail Awad, "Introduction to information security foundations and applications," In book: Information Security: Foundations, Technologies and Applications. Chapter: 1, The Institution of Engineering and Technology (IET), *Editors: Ali Ismail Awad and Michael Fairhurst*, 2018.

[2] Neowine developed security SoC DORCA -3 supporting asymmetric-key encryption, https://news.v.daum.net/v/20180109133504243.

[3] MS500: Low Power, Advanced Security Features for IoT, http://kr.ewbm.com/page/sub2_1

[4] P. Choi, Design and Implementation of High-Performance and Low-Complexity Security System on Chip (SoC), Ph. D. Dissertation, Hanyang University, 2017.

[5] A. P. Deb Nath, S. Ray, A. Basak and S. Bhunia, "System-on-chip security architecture and CAD framework for hardware patch," *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jeju, pp.733-738, 2018. DOI: 10.1109/ASPDAC.2018.8297409

[6] KS X 1213:2004, 128 bit Block Encryption Algorithm ARIA, Korean Agency for Technology and Standards (KATS), 2004.

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

[8] Paulo S. L. M. Barreto and Vincent Rijmen, "The WHIRLPOOL Hashing Function," pp.1-20, 2003. DOI: 10.1.1.529.3184

[9] ARM Cortex-M0, https://developer.arm.com /products/processors/cortex-m/

[10] K. B. Kim and K. W. Shin, "An Integrated Cryptographic Processor Supporting ARIA/AES Block Ciphers and Whirlpool Hash Function," *Journal of Institute of Korean Electrical and Electronics Engineers*, vol. 22, no. 1, pp. 38~45, 2018. DOI: 10.7471/ikeee.2018.22.1.38

BIOGRAPHY

Jun-Yeong Choe (Member)

2019 : BS degree in Electronic Engineering, Kumoh National Institute of Technology.
2019~ : Graduate student, Kumoh National Institute of Technology

Jun-Baek Choi (Member)

2019 : BS degree in Electronic Engineering, Kumoh National Institute of Technology.
2019~ : Graduate student, Kumoh National Institute of Technology

Kyung-Wook Shin (Member)

1984 : BS degree in Electronic Engineering, Korea Aerospace University
1986 : MS degree in Electronic Engineering, Yonsei University
1990 : Ph.D. degree in Electronic Engineering, Yonsei University

1990~1991 : Senior Researcher, Semiconductor Research Center, Electronics and Telecommunications Research Institute (ETRI)
1991~ : Professor in School of Electronic Engineering, Kumoh National Institute of Technology
1995~1996 : University of Illinois at Urbana- Champaign (Visiting Professor)
2003~2004 : University of California at San Diego (Visiting Professor)
2013~2014 : Georgia Institute of Technology (Visiting Professor)