Cortex-M0를 이용한 Whirlpool 해시함수의 하드웨어 구현

김동성 · 신경욱

금오공과대학교

A Hardware Implementation of Whirlpool Hash Function using Cortex-M0

Dong-seong Kim · Kyung-wook Shin

Kumoh National Institute of Technology

E-mail : kdsung322@kumoh.ac.kr

요 약

본 논문에서는 Whirlpool 해시 코어가 Cortex-M0의 슬레이브로 인터페이스된 보안 SoC 프로토타입 구현에 대해 기술한다. ISO/IEC에서 표준으로 채택된 경량 해시 알고리듬인 Whirlpool 해시 함수를 64-비트의 데이터 패스로 구현하였으며, 키 확장 연산과 암호화 연산을 수행하는 하드웨어를 공유하여 면적이 최소화되도록 설계하였다. 설계된 보안 SoC 프로토타입을 Cyclone-V FPGA에 구현한 후, ULINK2 어댑터와 Cortex 내부 디버거를 통해 Whirlpool 해시 코어에서 연산된 해시값을 확인함으로써 SoC 프로토타입의 동작을 확인했다.

키워드

Whirlpool, Security SoC, Cortex-M0, Hash Function

I. 서 론

최근 모바일 장치 보급의 확산과 함께 IoT 응용분야가 확대되면서 다양한 보안 위협에 대응하기 위한 정보보안 기술의 중요성이 강조되고 있다[1]. 정보보안은 대칭키 암호, 공개키 암호, 해시함수 등을 통해 이루어지며, 해시함수는 데이터 무결성(integrity)과 전자서명(digital signature) 등을 위해 사용된다. 대표적인 해시함수로는 MD 알고리듬, SHA 알고리듬, Whirlpool[2] 등이 있다. 최근에는 대칭키 및 공개키 암호, 해시함수, 난수발생기 등을 CPU와 통합하여 단일 칩으로 구현하는 보안 SoC에 관한 연구가 활발히 이루어지고 있다. 보안 SoC는 제품의 소형화, 고기능화가 가능하기 때문에 다양한 분야에 응용이 가능할 것으로 판단하고 있다.

본 논문에서는 경량화를 위해 64-비트 데이터 패스로 구현된 Whirlpool 해시 코어와 Cortex-M0를 통해 Whirlpool 해시 슬레이브를 제어하는 보안 SoC 프로토타입 구현에 대해 기술한다. II장에서는 Whirlpool 해시 알고리듬에 대해 간략히 기술하고, III장에서는 Cortex-M0 기반 보안 SoC에 Whirlpool 해시 코어를 슬레이브로 구현한 설계에 대해 설명한다. IV장에서는 FPGA 검증 결과에 대해 기술한 후, 결론을 맺는다.

II. Whirlpool 해시 알고리듬[2]

Whirlpool 해시함수는 블록암호 알고리듬인 AES (Advanced Encryption Standard)를 기반으로 하며, Miyaguchi-Preneel 구조로 이루어진다. 입력받을 수 있는 평문의 최대 길이는 2^{256} - 비트이다. 해시 연산이 수행되기 전에 평문이 블록 단위로 연산이 수행될 수 있도록 패딩된다. 평문 이후에 하나의 비트만 1로 채워지고 이어지는 비트는 0으로 채워져 256-비트의 홀수배가 되며, 256-비트의 평문 길이필드가 추가되어 전체 메시지 길이는 512-비트의 배수가 된다. 하나의 블록단위는 512-비트이며 연산이 수행될 때 총 10 라운드를 거치게 된다. 각 라운드마다 사용되는 라운드 키는 키 확장을 통해 생성되며 암호화와 동일한 과정으로 진행된다. 하나의 라운드는 SubBytes, ShiftColumn, MixRows, AddRoundKey 순서로 진행된다. 블록단위의 연산으로 출력되는 결과 값은 해당 블록연산을 수행할 때 입력되었던 평문 부분과 직전 블록에서 사용되었던 암호키와 함께 XOR 연산되어 중간 값으로 변환된다. 위 과정의 결과로 생성된 중간 값은 다음 블록의 암호키로 사용된다. 마지막 블록에서 생성된 결과 값은 Whirlpool 해시 함수의 최종 해시 값으로 정의된다.

III. Whirlpool 해시 코어의 Cortex-M0 기반 보안 SoC 구현

그림 1은 Cortex-M0 기반의 보안 SoC에 Whirlpool 해시 코어가 슬레이브로 인터페이스된 시스템 구조를 보이고 있다. Cortex-M0는 AHB 프로토콜에 따라 Whirlpool 해시 코어 슬레이브에 데이터를 쓰거나 읽기 동작을 수행한다. Whirlpool 해시 코어 슬레이브의 구조는 그림 2와 같이 Slave Interface 모듈, Wrapper 모듈 그리고 Whirlpool 해시 코어로 구성된다. Slave Interface 모듈은 Cortex-M0에서 사용하는 AHB- Lite 프로토콜에 맞춰 데이터를 전달하도록 설계되었다.

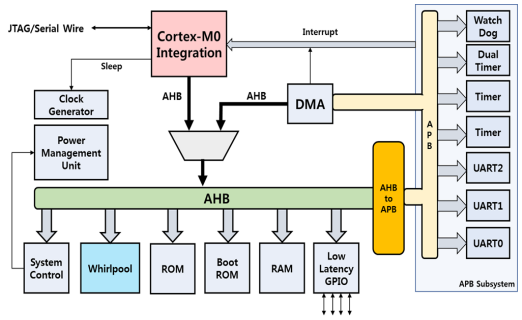

그림 1. Cortex-M0 기반의 보안 SoC 아키텍처.

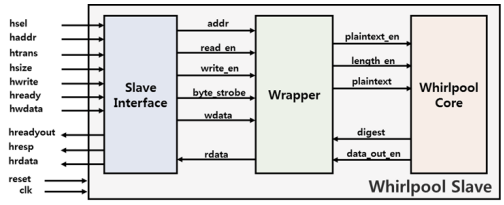

그림 2. Whirlpool 해시 코어 슬레이브의 구조.

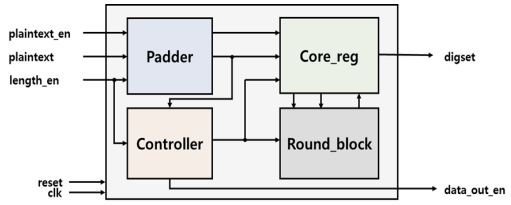

그림 3. Whirlpool 해시 코어의 내부 구조.

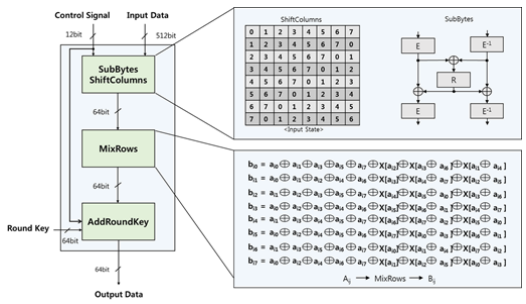

그림 4. Round_block의 구조.

Wrapper 모듈을 Slave Interface로부터 전달받은 데이터를 내부 레지스터에 저장한 후, 내부 레지스터의 Ctrl_reg에 데이터가 써지면 해당 비트에 대응하는 제어신호를 생성하고, Whirlpool 해시 코어가 연산을 수행할 수 있도록 데이터를 변환한다.

Whirlpool 해시 코어는 키 확장과 암호화 알고리듬이 동일하므로, 하드웨어 자원을 공유하도록 설계되었다.[3-4] 내부 구조는 그림 3과 같이 Padder, Controller, Core_reg, Round_block으로 구성된다. 내부 데이터 패스는 64-비트로 설계되었으며, 평문은 블록단위로 연산된다. Padder를 통해 평문이 패딩되며, Controller로부터 연산을 수행하기 위한 제어신호가 생성된다. Core_reg에는 연산의 중간결과와 라운드 연산에 필요한 라운드키가 저장된다. Round_block은 실질적인 연산이 수행되는 블록이며, 그림 4와 같은 구조로 설계되었다. SubBytes_ShiftColumns을 통해 64-비트 단위로 SubBytes와 ShiftColumns의 연산이 수행되며 MixRows는 그림 4에 표현된 수식을 통해 연산된다. AddRoundKey는 XOR 연산을 수행하며, 키 확장 단계에서는 라운드 상수와 입력 데이터를 XOR 연산하고, 암호화 과정에서는 키 확장을 통해 생성된 라운드키와 XOR 연산한다.

Whirlpool 해시 슬레이브가 AHB 프로토콜에 따라서 정상적으로 동작하는 지 확인하기 위해서 BFM (Bus Function Model) 시뮬레이션을 수행했으며 그 결과는 그림 5와 같다. 224-비트 평문 “4f5a6df1ad5f1d2f6a2d1fd88d4f”를 암호환 결과로 해시 값 “be4ef988eb9f9f35f77112b63dadc7d1d866a59eeba4f6caf74ac25fdf4b18d1e8590881d275d6aa6e726f36d1e2f375887755095b762a54999ac209a9ea420e”가 출력되어 정상 동작함을 확인했다.

그림 5. Whirlpool 해시 코어 슬레이브의 BFM 시뮬레이션 결과.

IV. FPGA 검증

FPGA 검증 시스템의 구성은 그림 6과 같다. Cyclone-V FPGA가 탑재된 V2M-MPS2 보드는 ULINK2 어댑터를 통해 Cortex 내부의 디버거로 연결되며, 이를 통해 Wrapper 내부에 있는 레지스터로 엑세스하여 Whirlpool 해시 코어에서 연산된 해시 값을 확인하였다. Cyclone-V FPGA에 Cortex-M0와 Whirlpool 해시 코어 슬레이브가 구현되고, CMSDK의 버스 매트릭스에 연결하여 동작을 검증하였다. FPGA에 구현된 SoC의 동작을 제어하기 위한 소프트웨어는 Keil의 uVision으로 크로스컴파일 되었으며, Whirlpool 해시 코어 슬레이브의 동작이 제어된다.

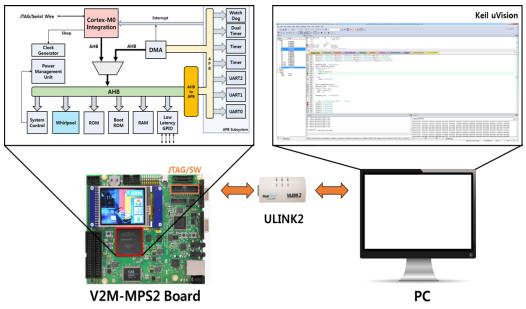

그림 6. FPGA 검증 환경.

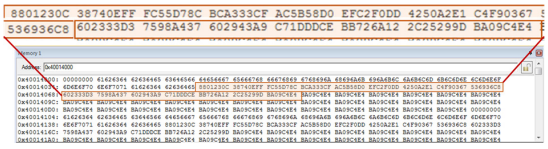

그림 7. Whirlpool 슬레이브의 FPGA 검증 결과.

그림 7은 ULINK2를 통해 얻어진 FPGA 검증 결과를 보이고 있으며, 512-비트의 평문 “abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopqabcdbcde”을 암호화한 결과로 해시 값 “8801230C38740EFFFC55D78CBCA333CFAC5B58D0EFC2F0DD4250A2E1C4F90367536936C8602333D37598A437602943A9C71DDDCEBB726A122C25299DBA09C4E4”가 출력되어 FPGA에 구현된 Cortex-M0와 Whirlpool 해시 코어가 정상적으로 동작함을 확인하였다.

ACKNOWLEDGMENTS

- 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)
- This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (No. 2017R1D1A3B03031677)
- Authors are thankful to IDEC for supporting EDA S/W

V. 결 론

본 논문에서는 Whirlpool 해시 코어를 AHB를 통해 Cortex-M0에 슬레이브로 연결하여 FPGA에 구현한 후, Slave Interface 모듈과 Wrapper 모듈을 통해 버스 매트릭스와 연동되고 Cortex-M0에 의해 제어되어 동작하는 것을 검증하였다. 향후 대칭키 및 비대칭키 암호 코어들을 추가하여 다양한 기능을 갖는 보안 SoC 구현으로 확장할 예정이다.

참고문헌

[1] A. Whitmore, A. Anurag, and L.D. Xu, “The Internet of Things - A Survey of Topics and Trends,” *Information Systems Frontiers,* vol. 17, no. 2, pp. 261-274, Apr. 2015.

[2] P. Barreto, and V. Rijmen, “The Whirlpool Hashing Function,” May 2003.

[3] P. Kitsos, and O. Koufopavlou, “Efficient Architecture and Hardware Implementation of the Whirlpool Hash Function,” *IEEE Transactions on Consumer Electronics*, vol. 50, Issue 1, Feb. 2004.

[4] Y.J. Kwon, D.S. Kim, and K.W. Shin, “A Hardware Implementation of Whirlpool Hash Function using 64-bit datapath,” *Journal of the Korea Institute of Information and Communication Engineering*, vol. 21, no. 2, pp. 485-487, Oct. 2017.