64-비트 데이터패스를 이용한 Whirlpool 해시 함수의 하드웨어 구현

권영진* · 김동성* · 신경욱*

*금오공과대학교

A Hardware Implementation of Whirlpool Hash Function using 64-bit datapath

Young-Jin Kwon* · Dong-Seong Kim* · Kyung-Wook Shin*

*Kumoh National Institute of Technology

E-mail : kwon8489@kumoh.ac.kr

요 약

국제 표준화 기구인 ISO/IEC에서 10118-3 표준으로 채택된 Whirlpool 해시 함수는 AES 블록 암호와 유사한 SPN(Substitution Permutation Network) 구조를 기반으로 하여 메시지의 무결성을 제공하는 알고리듬이다. 본 논문에서는 Whirlpool 해시 함수의 하드웨어 구현에 대해서 기술한다. 라운드 블록은 64-비트 데이터 패스로 설계하였으며, 10회의 라운드에 걸쳐서 암호화가 진행된다. 면적을 최소화하기 위해 키 확장과 암호화 알고리듬은 동일한 하드웨어를 사용한다. Verilog HDL을 이용해 Whirlpool 해시 함수를 모델링하였고, ModelSim으로 시뮬레이션을 수행하여 정상 동작을 확인하였다.

ABSTRACT

The whirlpool hash function adopted as an ISO / IEC standard 10118-3 by the international standardization organization is an algorithm that provides message integrity based on an SPN (Substitution Permutation Network) structure similar to AES block cipher. In this paper, we describe the hardware implementation of the Whirlpool hash function. The round block is designed with a 64-bit data path and encryption is performed over 10 rounds. To minimize area, key expansion and encryption algorithms use the same hardware. The Whirlpool hash function was modeled using Verilog HDL, and simulation was performed with ModelSim to verify normal operation.

키워드

Whirlpool, hash, internet of things, cryptography, information security

Ⅰ. 서 론

IoT(Internet of Things) 기술이 발전함에 따라 인터넷에 연결되는 기기와 서비스 범주가 확장되고 있다. 다양한 네트워크를 통해 정보를 공유하는 만큼 주변의 모든 것이 공격 대상이자 도구가 될 수 있으며 이를 방지하기 위해 적절한 보안서비스를 제공하는 것은 매우 중요하다.

국제 표준화 기구인 ISO/IEC에서 10118-3 표준으로 채택된 Whirlpool 해시 함수[1]는 블록암호를 기반으로 메시지에 대한 무결성을 보장하는 암호화 기법이며 디지털 서명, 전자상거래 등에 응용될 수 있다. 본 논문에서는 Whirlpool 해시 함수를 Verilog-HDL로 모델링 하고, ModelSim으로 시뮬레이션을 수행하여 기능을 검증하였다.

Ⅱ. Whirlpool 해시 알고리듬

Whirlpool은 수정된 AES(Advanced Encryption Standard)를 기반으로 한 Miyaguchi-Preneel 구조

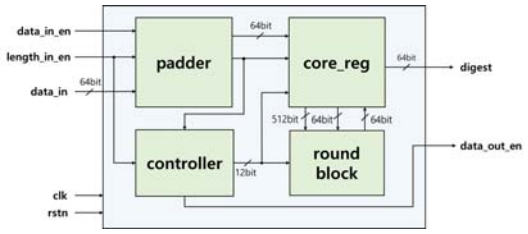


Fig. 1. Architecture of Whirlpool Hash Core

를 가진다. 2^{256}-비트보다 작은 길이의 메시지로 입력이 제한된다. 512-비트 블록단위로 10개의 라운드에 걸쳐 해싱 동작이 수행되며 하나의 512-비트 메시지 다이제스트를 생성해낸다.

해싱 동작은 512-비트 블록 단위로 수행되기 전에 메시지는 1로 패딩 되고 이어지는 비트들은 모두 0으로 채워진다. 패딩 된 메시지의 길이는 256-비트의 홀수 배수가 되며, 평문 길이를 정의하는 256-비트의 길이 필드가 추가됨으로써 메시지의 총 길이는 512-비트의 배수가 된다.

첫 번째 라운드의 암호 키는 모든 비트가 0인 값이며 각 블록을 암호화하여 생성된 값은 직전의 암호 키, 평문 블록과 함께 XOR 연산된 후 다음 블록의 암호 키로 사용된다. 마지막 블록에서 10개의 라운드가 수행된 후 출력되는 512-비트 암호문을 Hash 값으로 정의한다.

Whirlpool 해시 함수는 라운드키를 생성하는 키 확장 라운드 연산과 평문을 암호화하는 암호화 라운드 연산이 동일하다. 키 확장 라운드 연산은 이전 라운드에서 생성된 라운드키를 입력으로 받아 처리되며, 각 라운드마다 정의된 라운드 상수를 라운드키로 사용함으로써 암호화 라운드 연산에서 사용되는 라운드키를 생성한다.

Ⅲ. Whirlpool 해시 코어 설계

설계된 Whirlpool 해시 코어의 구조는 그림 1과 같으며 round block, padder, controller, core_reg 구성된다. 키 확장과 암호화는 연산이 동일하므로 round block을 재사용함으로써 수행된다[2]. data_in 포트를 통해 평문 메시지가 512-비트 블록단위로 8 클록 사이클에 걸쳐 64-비트씩 입력

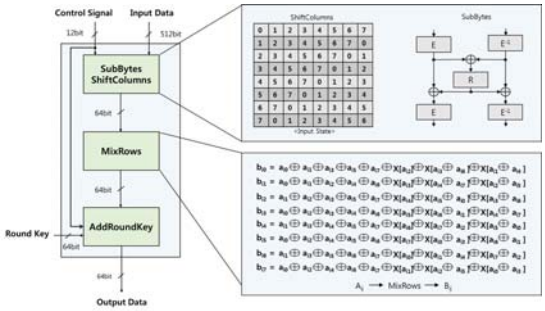


Fig. 2. Architecture of round block

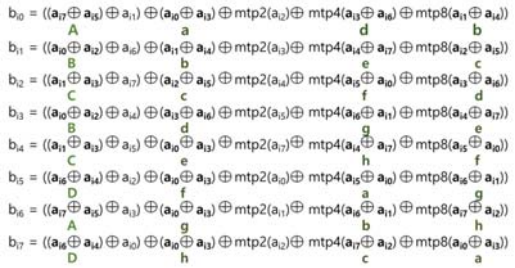


Fig. 3. Function of MixRows

된다. 입력된 평문 메시지는 padder에서 패딩된 후 core_reg에 512-비트로 저장된다. 첫 번째 라운드의 키 확장 연산이 수행된 후 출력되는 라운드키는 암호화 연산 수행을 위해 core_reg에 저장된다. 두 번째 라운드부터는 core_reg에 저장된 이전 단계 라운드키를 입력으로 키 확장 연산이 수행되며, 암호화는 이전 단계 라운드로부터 계산된 중간 값을 입력으로 사용하게 된다. 한 블록당 10 라운드를 수행하게 된다. core_reg에서 round block로 입력되는 데이터를 제외한 모든 과정에서 데이터는 64-비트 데이터 패스로 입·출력 및 저장이 이루어진다.

round block의 구조는 그림 2와 같다. SubBytes와 ShiftColumns은 하나의 블록에서 동시에 수행되고 MixRows와 AddRoundKey 순서로 각 라운드가 진행되며 각 모듈간 데이터 패스는 64-비트이다. SubBytes부터 AddroundKey까지 8 클록 사이클이 소모되며, 매 라운드마다 키 확장과 암호화가 각각 한 번씩 실행되므로 16 클록 사이클이 소모된다. 하나의 블록을 암호화하는데 10회의 라운드가 실행되므로 총 160 클록 사이클이 소모된다.

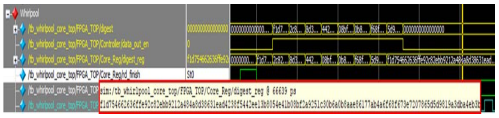
SubBytes_ShiftColumns 블록에 입력되는 데이터는 그림 2와 같이 8-비트 데이터를 원소로 하는 8x8 정방행렬이라고 가정한다. Input State 행렬의 번호 순서대로 64-비트씩 입력되며 이는 ShiftColumns 연산이 수행된 행렬의 행과 동일하다. SubBytes는 비선형변환을 하는 블록으로 E, E^{-1}, R-Box의 Look-up Table을 이용해서 연산이 진행된다.

MixRows 블록은 높은 확산(diffusion)을 제공하기 위해 행 단위로 혼합(mixing)하는 역할을 하며 그림 2의 MixRows 모듈에 표시된 식을 통해 연산된다[3]. 게이트 수를 줄이기 위해 그림 3 과 같이 공통항을 이용하여 구현하였다.

AddRoundKey 블록은 입력되는 데이터와 라운드 키의 XOR연산을 수행한다. 키 확장 연산에서는 라운드 상수와 입력 데이터를 XOR 연산하고, 암호화 연산과정에서는 키 확장단계에서 생성된 라운드키와 입력 데이터를 XOR 연산한다.

Ⅳ. 기능검증

Verilog HDL로 설계된 Whirlpool 해시 코어의 동작은 시뮬레이션으로 검증했으며, 검증결과는 그림 4와 같다. 그림 4-(a) 의 결과는 208-비트 평문 메시지 “abcdefghijklmnopqrstuvwxyz”을 입력으로 사용하여 암호화한 결과로 메시지 다이제스트 “f1d754662636ffe92c82ebb9212a484a8d38631ead4238f5442ee13b8054e41b08bf2a9251c30b6a0b8aae86177ab4a6f68f673e7207865d5d9819a3dba4eb3b”가 출력되고, 그림 4-(b) 의 결과는 112-비트 평문 메시지 “message digest”를 입력으로 사용하여 암호화한 결과로 메시지 다이제스트 “378c84a4126e2dc6e56dcc7458377aac838d00032230f53ce1f5700c0ffb4d3b8421557659ef55c106b4b52ac5a4aaa692ed920052838f3362e86dbd37a8903e”가 출력되어 논리기능이 정상적으로 동작함을 확인하였다.



(a)

(b)

Fig. 4. Simulation Result

Ⅴ. 결 론

본 논문에서는 Verilog HDL을 이용해 Whirlpool 해시 함수를 모델링하였고, ModelSim으로 시뮬레이션을 수행하여 정상 동작을 확인하였다. 설계된 프로세서는 배포한 소프트웨어의 불법적인 변경 여부를 점검하거나 대용량 자료의 송수신 무결성 검증을 요구하는 환경에 응용될 수 있을 것으로 예상된다.

참고문헌

[1] Barreto, P. and V. Rijmen. The Whirlpool Hashing Function. Submitted to NESSIE, May. 2003.

[2] P. Kitsos and O. Koufopavlou, Member, IEEE , Efficient Architecture and Hardware Implementation of the Whirlpool Hash Function, Vol. 50, Issue 1, February. 2004.

[3] Akashi Satoh, ASIC Hardware Implementations for 512-bit Hash Function Whirlpool. Added to IEEE, June. 2008.