SHA-3 해쉬함수 소비전력 특성 분석 및 저전력 구조 기법

김성호* · 조성호*

Analysis on Power Consumption Characteristics of SHA-3 Candidates and Low-Power Architecture

Sung-ho Kim* · Sung-ho Cho*

요 약

해쉬함수는 데이터와 명령에 대한 위변조를 방지와 같은 무결성 제공하거나 서명이나 키 분배 등 다양한 보안 프로토콜에서 서명 및 인증, 키 분배 목적으로 많이 사용되는 일방향성 함수(one-way function)다. 2005년 Wang에 의해 암호학적 취약성이 발견되기까지 해쉬함수로는 SHA-1이 많이 사용 되었다. SHA-1의 안전성에 문제가 생기게 되자 NIST(National Institute of Standards and Technology)에서는 암호학적으로 안전한 새로운 해쉬함수 개발 필요성을 느껴 2007년 11월에 공개적으로 새로운 해쉬함수에 대한 공모를 시작했으며, SHA-3로 명명된 새로운 해쉬함수는 2012년 최종 선정될 예정이다. 현재 제안된 SHA-3 함수들에 대한 암호학적인 특성과 하드웨어로 구현했을 때의 하드웨어 복잡도, 소프트웨어로 구현했을 때의 성능 등에 대한 평가가 이뤄지고 있다. 하지만 하드웨어로 구현된 해쉬함수의 중요한 특성 평가 척도(metrics)인 소비 전력 특성에 대한 연구는 활발히 이뤄지지 않고 있다. 본 논문에서는 제안된 SHA-3 해쉬함수를 하드웨어로 구현했을 경우의 소비 전력 특성을 분석하고 소비전력 특성 분석 결과를 토대로 SHA-3 해쉬함수 중에서 새로운 SHA-3 해쉬함수로 선정될 확률이 높은 Luffa 함수에 대한 저전력 구조를 제안한다. 제안된 저전력 구조는 기존의 Luffa 하드웨어보다 약 10% 정도 적은 전력을 소비함을 보인다.

ABSTRACT

Cryptographic hash functions are also called one-way functions and they ensure the integrity of communication data and command by detecting or blocking forgery. Also hash functions can be used with other security protocols for signature, authentication, and key distribution. The SHA-1 was widely used until it was found to be cryptographically broken by Wang, et. al, 2005. For this reason, NIST launched the SHA-3 competition in November 2007 to develop new secure hash function by 2012. Many SHA-3 hash functions were proposed and currently in review process. To choose new SHA-3 hash function among the proposed hash functions, there have been many efforts to analyze the cryptographic secureness, hardware/software characteristics on each proposed one. However there are few research efforts on the SHA-3 from the point of power consumption, which is a crucial metric on hardware module. In this paper, we analyze the power consumption characteristics of the SHA-3 hash functions when they are made in the form of ASIC hardware module. Also we propose power efficient hardware architecture on Luffa, which is strong candidate as a new SHA-3 hash function. Our proposed low power architecture for Luffa achieves 10% less power consumption than previous Luffa hardware architecture.

키워드

해쉬함수, 보안, SHA-3, 하드웨어 복잡도, 소비전력

Key word

Hash function, security, SHA-3, hardware complexity, power consumption

* 정회원 : 한양대학교 공과대학 융합전자공학부 (zoozang@gmail.com) 접수일자 : 2010. 10. 22
심사완료일자 : 2010. 11. 16

Ⅰ. 서 론

정보통신 기술의 급격한 발전은 사물이 지능화 및 네트워크화 되는 유비쿼터스(혹은 M2M:Machine To Machine) 환경 실현 가능성을 더욱 높이고 있다. 특히 최근 스마트폰 시장이 급격히 확대되고 스마트폰이 모바일 RFID 리더와 센서네트워크 게이트웨이 등으로 사용되어 사물과의 중간 통신 매개 수단 역할을 하게 됨에 따라, 유비쿼터스 환경은 먼 미래의 기술이 아닌 조만간에 실현 가능한 것으로 볼 수 있게 되었다. 현재 국내외에서 많은 연구 개발이 진행되고 있는 유비쿼터스 환경 실현 기술로는 이미 언급한 모바일 RFID 기술이나 센서네트워크 기술뿐만 아니라 사물 통신 기술과 스마트그리드 기술, 공간정보지능화 기술 등이 있다.

한편, 유비쿼터스 기술은 각 응용 환경에 적합한 보안 기술을 적절히 갖춰야만 사람에게 편리하고 신뢰할 수 있는 서비스를 제공할 수 있다. 만약 적절한 보안 기술을 갖추지 못한다면 사물의 지능화 및 네트워크화는 개인에 대한 프라이버시 침해, 물리적인 공격, 행동 통제, 정보 침해 등, 인터넷상의 보안 침해와는 비교 되지 않을 정도로 큰 피해를 서비스 사용자에게 가져다 줄 것이다. 이에 많은 보안 연구자들은 안전하고 신뢰할 수 있는 유비쿼터스 실현 기술 개발을 위해 암호학적인 수단 혹은 실용적인 수단을 사용하여 보안 기술에 대한 연구/개발을 하고 있다. 이러한 보안 기술의 핵심에는 해쉬함수(hash function)가 존재하는데, 해쉬함수는 데이터와 명령 등에 대한 위변조 공격을 방지하는 무결성(integrity) 보안 서비스를 제공할 뿐만 아니라, 다른 보안 기법 및 프로토콜과 함께 같이 사용되어 정보 원천(source)에 대한 인증(authentication), 디지털 서명(digital signature), 키 분배(key distribution) 등을 제공한다.

해쉬함수로는 2005년 Wang 등에 의해 암호학적 취약성이 발견되기까지 SHA-1 혹은 SHA-2 함수를 많이 사용했다. 암호학적인 취약성이 발견됨에 따라 NIST (National Institute of Standards and Technology)는 암호학적으로 안전한 새로운 해쉬함수에 대한 개발 필요성을 느껴 2007년 11월에 공개적으로 새로운 해쉬함수에 대한 공모를 시작했으며, 2012년에 기존의 해쉬함수보다 암호학적으로 안전한 새로운 해쉬함수인 SHA-3를 최종 선정할 예정이다[1].

이에, SHA-3 해쉬함수 후보로서 많은 알고리즘이 제안되었다. 많은 연구자들은 제안된 SHA-3 후보 함수에 대한 암호학적인 안전성과 하드웨어로 구현했을 때의 하드웨어 복잡도, 소프트웨어로 구현했을 때의 성능 등에 대한 평가를 수행하여 적합한 SHA-3 해쉬함수 선정을 위한 노력을 하고 있다. 하지만 하드웨어로 구현된 해쉬함수의 중요한 특성 평가 척도(metrics)인 소비 전력 특성에 대한 연구는 활발히 이뤄지지 않고 있다. 즉, 유비쿼터스 응용 중에서 RFID나 센서네트워크, 스마트그리드, 사물 통신 등에서는 저전력으로 보안 기술을 구현하는 것이 가장 중요한 요소 및 성능 지표(metrics) 중의 하나임에도 불구하고 국내외적으로 SHA-3에 대한 소비전력 특성에 대한 연구는 거의 이뤄지지 않고 있는 상황이다.

이에, 본 논문에서는 SHA-3 후보 해쉬함수를 하드웨어로 구현했을 경우의 소비 전력 값을 추정하며, 이를 토대로 SHA-3 후보 해쉬함수의 구조적 특성을 분석했다. 분석 결과는 SHA-3 해쉬함수 선정 및 새로운 저전력 해쉬함수 프리미티브 설계에 활용될 수 있다. 또한, SHA-3 후보 함수 중에서 안전성 및 하드웨어/소프트웨어 구현시 좋은 특성을 가지는 것으로 알려진 Luffa 해쉬함수에 대한 저전력 구조를 제안한다. 논문의 구성은 2장에서는 SHA-3 해쉬함수 중에서 소비전력 분석 대상에 대한 간략한 소개와 함께 SHA-3 해쉬함수에 대한 소비전력 추정 방법론을 설명한다. 3장에서는 SHA-3 해쉬함수 하드웨어 구조와 소비전력 특성 분석 결과를 보인다. 4장에서는 저전력 Luffa 해쉬함수 구조를 제안하고 이의 특성을 분석한 후, 5장에서 본 논문의 결론을 낸다.

Ⅱ. SHA-3 해쉬함수에 대한 소비전력 분석 기법

2.1 소비전력 분석 대상인 SHA-3 해쉬함수

이 절에서는 소비전력 분석 대상인 Luffa, Keccak, Fugue, Grøstl 해쉬함수 구조를 간략히 살펴본다. 2차 라운드를 통과한 총 14개의 SHA-3 후보 함수(BLAKE, Blue Midnight Wish, CubeHash, ECHO, Fugue, Grøstl, Hamsi, JH, Keccak, Luffa, Shabal, SHAvite-3, SIMD,

Skein) 중에서 아래의 네 개를 선택하여 이에 대한 소비 전력을 분석했다. 네 가지 해쉬함수를 선택한 이유는 본 논문에서 선택한 해쉬함수를 하드웨어로 구현했을 때의 성능(특히 처리율: throughput)과 소프트웨어 구현시의 성능, 안전성이 좋은 것으로 알려져 있기 때문이다 [2]. 즉, 위 네가지 해쉬함수 중에서 하나 혹은 복수개가 최종 SHA-3 함수로 선정될 확률이 높기 때문에 위 네가지 해쉬함수 각각에 대한 소비 전력 특성을 분석했다. 각 해쉬함수에 대한 소비 전력 특성을 분석해봄으로서 해쉬함수를 수동형 RFID 태그와 같은 저전력 응용에 적용할 경우의 적용 가능성과 적용시의 장단점을 알 수 있을 것이다.

2.1.1 Luffa

Luffa는 Sponge 구조의 해쉬함수다. 이는 메시지와 이전 해쉬의 출력을 입력 받아 XOR 연산과 갈로아 필드 곱셈(×2) 연산을 통해 값을 섞어 주는 Message Injection(MI)블록을 가진다. 또한, 상태 값을 순환시켜주는 Tweak 블록, 그리고 반복 구조로 다시 한 번 상태 값을 섞어 주는 Step 블록으로 구성된다. 이 중 Step 블록은 반복 되는 부분이며, 이는 다시 SubCrumb이라고 불리는 S-BOX구조와 블록 순환(rotate) 및 XOR로 구성된 MixWord, 그리고 LFSR(Linear Feedback Shift Register)을 통해 루프가 반복될 때 마다 생성되는 상수를 더해주는 AddConstant 블록으로 구성된다. Luffa에서 상태 값은 해쉬 출력 길이에 따라 그 크기가 달라지며, 256비트 출력을 갖는 Luffa의 경우 256비트 블록 3개를 상태 값으로 갖는다. Step 블록은 256 비트의 입출력을 가지므로 일반적인 경우 3개의 Step 블록이 필요하다.

2.1.2 Keccak

Keccak 역시 sponge 구조의 해쉬함수로서, 간단한 라운드 함수의 반복으로 구성되어 있고, 키 스케줄이 없는 블록 암호방식과 유사하다. Keccak의 라운드 함수는 7개의 Keccak-f permutation을 가지며, 1600bit의 상태 값을 가진다. Keccak-f는 XOR, AND, NOT, 순환의 4가지 기본연산으로 구성되어 있다. Keccak-256의 입력은 1088비트 이지만, 1600비트의 상태 값으로 만들기 위해 0을 확장시켜준다. 이 상태 값은 64개의 5x5의 행렬 형태로 나타낼 수 있다. 입력은 이전의 해쉬 출력과 XOR 되며, 라운드 함수 내부에서 24번의 라운드 반복을 거치게 된다. 최종 출력은 1600비트의 상태 값 중 일부를 사용한다.

2.1.3 Fugue

Fugue 역시 Sponge 구조이며, 주요구조는 AES의 라운드 함수와 유사한 형태를 갖는 SMIX라는 구조이다. 256비트 출력을 갖는 Fugue는 960비트의 메시지를 입력으로 받으며, 30 컬럼의 4바이트 상태 값을 유지한다. Fugue는 메시지의 입력이 있을 때 반복되는 Round 구조와 마지막 해쉬값 출력을 위한 Final Round 구조를 가진다. 반복 Round 구조에서는 TIX(I), ROR3, CMIX, SMIX, ROR3, CMIX, SMIX가 연속적으로 실행되는 구조이다. TIX(I)는 XOR, Truncate, Insert, XOR 연산을 수행하며, ROR3은 상태를 오른쪽으로 3컬럼 순환 시킨다. CMIX는 Column mix로서 컬럼 간의 XOR 연산으로 구성된다. ROR3, CMIX, SIX를 subround라 부르며, TIX(I)에 이어 2개의 subround가 수행된다고 볼 수 있다. Final Round에서는 RORn, CMIX, SMIX, XOR연산이 반복하여 수행되며, 상태 값 중 일부 값을 해쉬값으로 출력한다.

2.1.4 Grøstl

grøstl은 Wide-Pipe구조의 해쉬함수로서, AES의 구조를 기반으로 하고 있다. AES 구조와의 차이점은 grøstl은 출력 길이에 따라 8x8 혹은 8x16의 상태 행렬을 가지며, AddRoundKey 대신 AddRoundConstant가 사용된다는 것이다. grøstl은 P와 Q 두 개의 함수를 가지는데, 바로 앞에서 설명한 AES 기반의 라운드 함수이다. AddRoundConstant단계에서 더해지는 상수가 다르다는 차이를 빼면 동일하다. 출력되는 해쉬를 h, 메시지를 m이라고 한다면 메시지에 대한 중간 해쉬값 출력은 $f(h,m) = P(h \oplus m) \oplus Q(m) \oplus h$ 의 형태로 출력되며, 최종 해쉬값 출력은 $\Omega(x) = trunc_n(P(x) \oplus x)$로 n 비트만큼 가변 길이로 출력할 수 있도록 되어 있다.

2.2 SHA-3 해쉬함수에 대한 소비전력 분석 기법

SHA-3 해쉬함수에 대해 소비전력을 분석하기 위해 본 논문에서는 아래 그림 1에 나타나 있는 소비전력 추정 절차를 사용했다. 이 단계를 보면 다음과 같다.

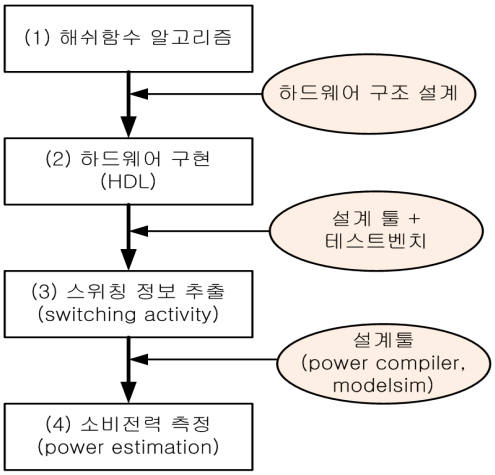

그림 1. 해쉬함수 소비전력 추정을 위한 절차
Fig.1. Procedure for power consumption estimation for hash functions

- 단계 1: 먼저 소비전력 추정 대상이 되는 해쉬함수에 대한 구조적인 분석을 수행한다.
- 단계 2: 해쉬함수의 구조 분석이 끝나면 이에 대한 하드웨어 구조를 설계한 후, 이를 VHDL 혹은 Verilog와 같은 HDL(Hardware Description Language)로 구현한다.
- 단계 3: HDL로 구현한 분석 대상 하드웨어 블록은 Synopsys사의 Design Compiler로 합성(synthesis)한다. 또한, HDL로 구현된 하드웨어 블록을 modelsim으로 시뮬레이션 하여, 스위칭 정보를 얻는다.
- 단계 4: 얻은 switching 정보는 Synopsys사의 Power Compiler (Design Compiler에 통합되어 있음)툴을 사용하여 해당 해쉬함수 하드웨어 모듈에 대한 소비전력 추정치 값을 얻는다.

그림 1의 단계 (3)을 보면 SHA-3 하드웨어 모듈에 대한 소비전력 값을 얻기 위해서는 스위칭 정보를 추출해야 한다. 하드웨어 회로의 스위칭 동작과 소비전력은 매우 높은 상관관계를 가진다. 즉, 그림 2를 보면, CMOS 인버터 회로에 0에서 1로 바뀌는 입력 값을 입력하면 출력 값은 1에서 0으로 바뀐다. CMOS 인버터 회로에서 소비되는 전력을 분석하면, 먼저 입력 값의 변화에 의해 출력 단에는 I_{sw} 전류가 흐른다. 해당 전류는 부하 커패시턴스(C_{load})에 공급된다. 이를 스위칭 전력(switching power)라고 하며 $C_{load} \times V_d^2 \times f_{switch}$으로 정의된다. f_{switch} 값은 스위칭 주파수로서 스위칭이 빨리 일어날수록 소비전력은 많아진다. CMOS 인버터 회로가 동작할 때, 두 개의 트랜지스터(pMOS, nMOS)가 모두 ON이 되는 시점이 발생하며 이 때 전류 I_{sc}가 흐르게 된다. 이를 내부 전력(internal power)이라고 한다. 이 두 가지 소비전력 요소는 모두 회로가 동작할 때 발생하는 소비전력이므로 동적 전력(dynamic power)이라고 한다. 하드웨어 모듈에서 동적 전력 요소가 전체 소비 전력의 약 70%에서 90% 정도를 차지한다고 알려져 있다[3]. 이 때문에 본 논문에서는 동적 전력을 중심으로 SHA-3의 소비전력 특성을 분석했다.

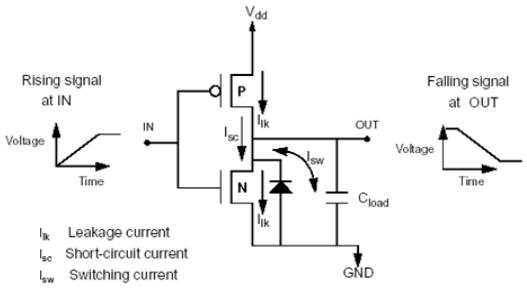

그림 2. CMOS 인버터 회로에 대한 스위칭 동작과 소비전력 상관관계 설명
Fig. 2. Relationship between the switching activity and power consumption on CMOS inverter

한편 그림 2를 보면, I_{lk} 요소가 있는데. 이는 트랜지스터의 소스와 드레인간의 누설 전류 등에 의한 값이다. 이 값은 회로가 동작하지 않아도 발생하는 소비전력이므로 정적 전력(static power)이라고 한다. 본 논문에서는 SHA-3의 소비 전력 요소를 고려할 때, 정적 전력 요소는 고려하지 않고 동적 소비 전력 요소만 고려했다.

그림 1에 기술된 해쉬함수 하드웨어에 대한 소비 전력 추정 절차는 그림 3처럼 HDL 형태의 RTL 레벨 하드웨어 모듈에 대한 소비전력을 추정하는 상세 단계(그림 1의 단계 2,3,4)로 표현할 수 있다[4,5]. 각 단계를 설명하면 다음과 같다.

- 단계 1: HDL로 기술된 하드웨어 블록을 Synopsys사의 Design Compiler를 사용하여 analyze와 elaborate

작업을 수행한다. 이 단계를 거치면 Design Compiler 툴이 해당 HDL 코드를 이해할 수 있는 형태로 변환된다.

- 단계 2: 이 단계에서는 Synopsys 툴로 읽은 설계 데이터베이스를 SAIF(Switching Activity Interchange Format) 파일로 변환한다. SAIF 파일은 로직이 동작할 때 내부 스위칭 활동을 기록할 수 있는 포맷이며, Modelsim과 같은 시뮬레이터 프로그램이 회로에 대한 시뮬레이션을 수행할 때 스위칭 정보를 기록한다.

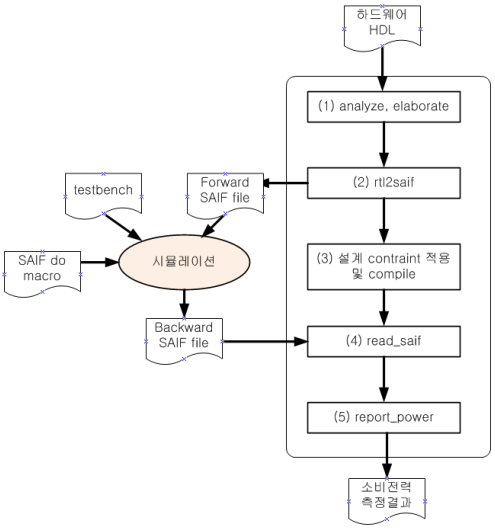

그림 3. Synopsys 툴을 사용한 RTL 레벨 하드웨어 모듈에 대한 소비 전력 추정 절차
Fig. 3. Procedure for power consumption estimation for RTL level hardware hash block

- 단계 3: 단계 3에서는constraint 정보를 사용하여 설계 데이터베이스에 대한 합성을 수행한다.
- 단계 4: 이 단계에서는 Modelsim 툴을 사용하여 RTL 레벨의 하드웨어 블록에 대해테스트벤치 입력 값과 SAIF 파일을 이용하여 시뮬레이션해서 얻은 Backward SAIF 파일을 읽는다.
- 단계 5: Backward SAIF 파일을 읽은 Design Compiler 툴(Power Compiler)은 소비전력 추정 값을 출력한다.

본 절에서 기술한 소비전력 추정 방법을 사용하여 SHA-3 후보 해쉬함수에 대한 소비전력 분석 결과를 다음절에 제시한다. 사용한 반도체 공정은 TSMC사의 0.13um CMOS 공정이었으며 하드웨어 설계를 위해선 verilog HDL을 사용했다. 즉, 본 논문에서는 현재 반도체 칩 설계에서 많이 사용하고 있는 검증된 설계 방법(Verilog HDL 사용)과 시뮬레이션 기법(Modelsim 사용), Front-end/Back-end 합성 기법(Synopsys)을 사용하여 해쉬 함수에 대한 실제 하드웨어 설계를 수행하였다. 또한, 설계된 SHA-3 하드웨어 칩에 대하여 그림 3에 제시된 소비전력 추정 기법을 사용하여 소비전력 값을 추정하였다. 이와 같은 소비전력 추정 기법은 이미 타연구 결과에서 실제 칩으로 구현했을 경우와 비교할 때 정확도가 높다는 것이 증명된 방법이며, 본 논문에서는 이와 같은 방법을 사용했다[4,5,7].

Ⅲ. SHA-3 소비전력 특성 분석

3.1 SHA-3 공통 하드웨어 플랫폼 구조

본 논문의 소비전력 분석 대상인 Luffa와 Keccak, Fugue, Grøstl 해쉬함수는 2.1절에서 기술한 것처럼 서로 다른 구조적인 특성을 가진다. 이 때문에 하드웨어로 구현했을 때, 서로 다른 구조를 가지며, 이러한 서로 다른 하드웨어 구조는 해쉬함수에 대한 공정한 소비 전력 평가를 어렵게 만든다. 이 때문에 소비전력 특성 평가를 위한 공통 하드웨어 플랫폼 구조(common hardware platform architecture)를 제시한다.

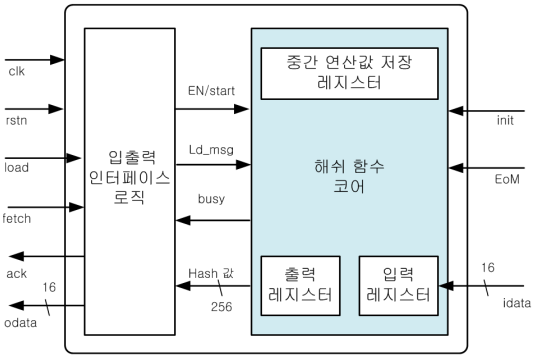

그림 4. SHA-3 해쉬함수 소비전력 추정을 위한 공통 하드웨어 플랫폼 구조
Fig. 4. Common hardware platform architecture for power consumption estimation of hash functions

본 논문에서 사용한 공통 하드웨어 플랫폼 구조는 K.Kobayashi [6]가 하드웨어 성능(처리율: throughput) 평가를 위해 사용한 구조를 참고하여 설계 되었다. 그림 4에 제시된 해쉬함수 소비전력 분석을 위한 공통 하드웨어 플랫폼 구조는 크게 입출력 인터페이스 로직과 해쉬함수 코어로 구성된다. 입출력 인터페이스 로직은 대부분의 하드웨어 블록이 입출력 통신 오버헤드(communication overhead)에 크게 의존하기 때문에 서로 다른 해쉬함수에 대해서도 이를 동일하게 구현함으로서 공정한 비교를 가능하게 한다.

즉, 어떤 해쉬함수는 32비트 단위로 해쉬 대상 입력 메시지를 받고 또 다른 해쉬함수는 한꺼번에 수 백 바이트 길이의 메시지를 받을 수 있다. 또 다른 경우에는 이미 메시지를 받았다는 가정 하에 해쉬함수를 동작시킬 수도 있다. 이 경우 모두 서로 다른 레지스터 크기와 입출력을 위한 클록 사이클 수가 다르기 때문에 공정한 하드웨어의 소비전력 특성 비교가 불가능해진다. 이 때문에 여러 해쉬함수에 대한 공정한 소비전력 분석을 위해선 입출력 인터페이스를 먼저 동일하게 구현해야 한다.

입출력 인터페이스를 동일하게 하기 위해 그림 4처럼 입출력 레지스터의 크기와 구조, 중간 연산값 저장 레지스터의 크기와 구조, 입출력 인터페이스 로직의 동작 방법, 해쉬함수 동작 제어 방법 및 이를 위한 입출력 포트를 동일하게 했다. 각 해쉬함수는 이러한 입출력 구조에 정합(matching) 되도록 설계 된 후, 공정한 소비전력 분석이 이뤄졌다.

3.2 SHA-3 소비전력 특성 분석 결과

Luffa와 Keccak, Fugue, Grøstl 해쉬함수에 대한 소비전력이 본 논문에서 제시한 소비전력 추정 방법과 공통 하드웨어 플랫폼 구조에 따라 이뤄졌다. 표 1은 소비전력 추정 값을 보여주고 있다. 표 1을 보면, Luffa 해쉬함수의 소비전력이 가장 적은 0.56uW 이며, Keccak은 이보다 약 3.5배 더 많은 1.96uW, Fugue와 Grøstl은 Luffa보다 각각 약 5.6배, 5.8배 정도 더 많은 전력을 소비한다.

표 1. 네 가지 SHA-3 후보군의 소비전력 분석 결과
Table 1. Power consumption estimation results for four SHA-3 candidates

SHA-3 해쉬함수	동적 전력(dynamic power) (uW)
Luffa	0.56
Keccak	1.96
Fugue	3.17
Grøstl	3.24

모두 출력 해쉬값을 256 비트를 출력하고 공통된 하드웨어 플랫폼 구조에 따라 설계했음에도 소비전력에 있어서 매우 큰 차이를 보이는 것은 해쉬함수의 선택에 있어서 성능만큼이나 중요한 요소임을 알 수 있게 한다. 즉, 해쉬함수를 수동형 RFID 태그와 같이 태그의 동작 에너지를 RFID 리더로부터 수신한 RF 신호로부터 얻는 응용인 경우, 해쉬함수에서 소비되는 전력이 해쉬함수에 따라 수 배 이상 차이 난다고 하면 당연히 저전력 해쉬함수를 사용하는 것이 올바른 선택일 것이다. 표 1의 네 가지 해쉬함수 소비전력 추정 값이 서로 큰 차이를 가지는 이유에 대해 각 해쉬함수의 구조 및 특성을 분석함으로서 알아본다. 먼저 본 연구 결과에 의해 하드웨어 블록에서 전력을 많이 소비하는 부분을 표 2와 같이 정의했다.

표 2. 전력 소비 주요 원인
Table 2. Principal reason for consuming power

전력 소비 요인 인덱스	전력 소비 요인
C1	많은 레지스터 혹은 메모리를 사용하는 경우
C2	레지스터 혹은 메모리 저장 값의 변화가 많은 경우
C3	멀티플렉서(MUX)를 많이 사용하는 경우
C4	조합회로를 많이 사용하는 경우
C5	데이터 버스 폭이 넓은 경우
C6	가산기, 감산기, 곱셈기 등 연산 회로를 많이 사용하는 경우
C7	조합회로 출력 값의 변화가 많은 경우
C8	여러 가지 동작이 동시에 수행되는 경우(병렬처리)
C9	큰 규모의 Lookup 테이블을 가지는 경우

표 2를 보면, 전력 소비 요인을 크게 두 가지로 다시 분류할 수 있는데, 레지스터 혹은 메모리를 많이 사용(C1)하거나 멀티플렉서를 많이 사용하는 경우(C3), 조합회로를 많이 사용하는 경우(C4), 데이터 버스 폭이 넓은 경우(C5), 가산기, 감산기, 곱셈기 등 연산회로를 많이 사용하는 경우(C6), 큰 규모의 Lookup 테이블을 가지는 경우(C9)는 하드웨어의 복잡도와 소비전력과의 밀접한 상관관계가 있음을 보여준다. 그리고 두 번째로는 레지스터의 값의 변화가 많은 경우(C2)와 조합회로의 출력값 변화가 많은 경우(C7), 병렬로 여러 가지 동작이 동시에 수행되는 경우(C8)는 스위칭 동작이 소비 전력과 밀접한 상관관계가 있음을 알 수 있다.

3.2.1 Luffa

표 1을 보면 네 가지 해쉬함수 중에서 Luffa의 소비 전력 값이 가장 작다는 것을 알 수 있다. 이는 Luffa가 Sponge 구조를 가지며 Message Injection 블록(그림5의 MI 블록)과 Permutation 블록(그림5의 P 블록) 등 비교적 단순한 조합회로로 이뤄진다는 사실 때문이다. 또한 사용하는 레지스터의 개수가 많지 않고 알고리즘에서 사용하는 S-Box의 크기가 16 X 4bits로 비교적 작다. 더욱이 해당 S-Box는 조합 회로로도 쉽게 구현할 수 있기 때문에 소비되는 전력이 낮다. 하지만, Luffa 알고리즘은 Permutation 블록 연산시 Step 블록이 병렬로 계산된다. 이는 성능 향상 측면에서는 장점으로 작용하지만 전력 소비 차원에서는 단점으로 작용한다. 즉, Luffa 해쉬함수는 전력 소비를 많이 하는 요소 중에서 C8 한 경우만을 가진다.

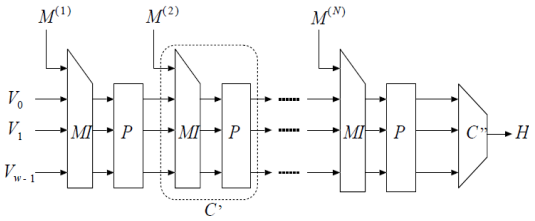

그림 5. Luffa 해쉬함수 기본 구조
Fig. 5. Basic architecture for the Luffa hash function

3.2.2 Keccak

Keccak 해쉬함수도 소비 전력 차원에서 매우 잘 만들어진 함수다. Keccak 함수는 그림 6에서 보는 것처럼 크게 MUX와 레지스터, 그리고 permutation 으로 구성된다. 본 해쉬함수는 표 2에서 정의된 전력 소비 주요 원인 중에서 C1, C3, C8 요소를 가진다. 즉, 1600 비트 정도의 큰 크기의 내부 상태 레지스터를 가지며(C1), 네 개의 1600 비트 값을 선택하는 큰 크기의 MUX를 갖는다(C3). 또한, θ와 ρ, π, χ, ι로 정의되는 Keccak permutation function $Keccak-f$는 복잡한 조합 회로를 유발할 가능성이 있다(C4). 하지만, $Keccak-f$ 함수를 분석해 본 결과, 과도한 조합회로를 유발하지 않았다. 이러한 Keccak 해쉬함수의 특성은 표 1에서처럼 Keccak 해쉬함수가 Luffa보다는 소비 전력이 많지만, 다른 해쉬함수보다는 현저히 적은 소비 전력을 가지도록 한다.

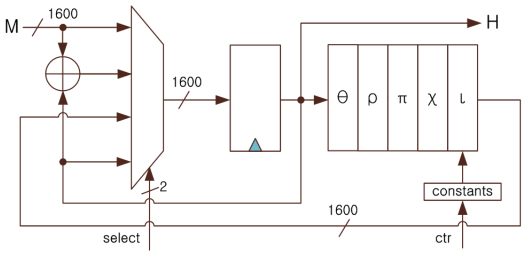

그림 6. Keccak 해쉬함수 기본 구조
Fig. 6. Basic architecture for the Keccak hash function

3.2.3 Fugue

Fugue 함수도 전형적인 Sponge 구조를 가진다. 하지만, 이 해쉬함수의 가장 큰 특징은 AES S-Box를 사용한다는 점이다. Luffa도 S-Box를 가지지만, 4비트 입력에 4비트 출력을 갖는 경량의 S-Box이지만, AES S-Box는 이보다는 복잡도가 높은 256 바이트 크기를 갖는다. 즉 이러한 Fugue의 특성은 표 2의 C9 전력 소비 요소에 해당 된다. 이 때문에 Fugue 해쉬함수는 Luffa보다 더 많은 전력을 소비하며, 또한 S-Box를 사용하지 않는 Keccak보다 더 많은 전력을 소비한다. 그림 7의 ROR(Right Rotation)은 단순한 wiring으로 실현되므로 복잡도는 높지 않다. TIX도 exclusive OR와 truncate, insert 정도의 연산이므로 전력을 많이 소비하는 요소에 해당되지는 않는다. 하지만, 그림7에서 보는 것처럼 데이터패스가 비교적 길며, 데이터패스를 제어하기 위해

MUX를 최소한 4개 사용해야 하기 때문에, 이는 C3 전력 소비 요소에 해당되어 Luffa와 Keccak에 비해 많은 전력을 소비한다.

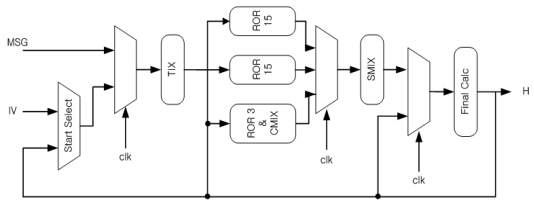

그림 7. Fugue 해쉬함수의 기본 구조
Fig. 7. Basic architecture for the Fugue hash function

3.2.4 Grøstl

grøstl 해쉬함수도 Fugue처럼 AES의 S-Box를 사용하고 있다. 구조를 기반으로 하고 있다. 이 때문에 C9 전력 소비 요소를 가진다.

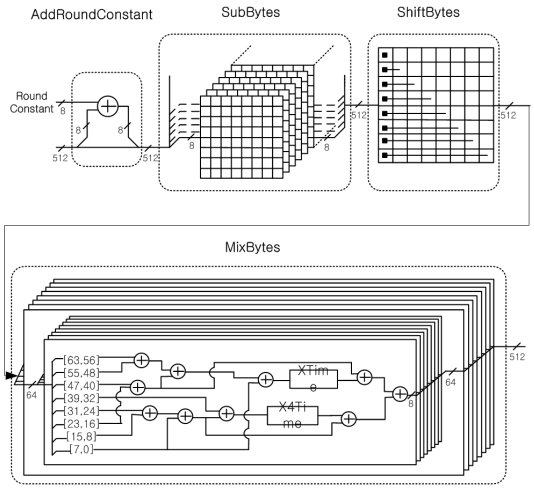

그림 8. Grøstl 해쉬함수의 기본 구조
Fig. 8. Basic architecture for the Grøstl hash function

또한, 그림 8에서 보는 것처럼 SubByte 블록과 ShiftBytes 블록, MixBytes은 전력 소비 요소 C7에 해당한다. Grøstl 해쉬함수는 비록 전력 소비 요소에 해당하는 요소는 두 가지에 불과했지만, 그 영향이 다른 요소보다 크기 때문에 본 논문에서 소비 전력 분석 대상인 네 가지 해쉬함수 중에서 가장 높은 전력을 소비한다는 것을 확인했다.

지금까지 SHA-3 후보 해쉬함수 네 가지에 대해 소비 전력 값을 추정하고 이에 대해 분석했다. 한편, SHA-2 (SHA-256)에 대해서도 하드웨어 설계를 통해 소비 전력 값을 구했다. 표 3에서 볼 수 있는 것처럼, SHA-2가 기존의 SHA-3 후보군보다 더 적은 전력을 소비함을 알 수 있었다. 이는 SHA-3에서는 해쉬함수의 안전도(level of security)를 높이기 위해 더 많은 암호학적인 기법을 사용했다. SHA-3의 소비전력이 높은 것은 이러한 이유 때문인 것으로 판단된다. 수동형 RFID 태그나 센서노드, M2M, 스마트그리드 등, 다양한 유비쿼터스 응용 환경에서 해쉬함수를 사용하기 위해선 기존의 해쉬함수보다 저전력으로 구현할 수 있어야 한다. 이에 본 논문에서는 분석 대상이 된 네 가지 SHA-3 후보 해쉬함수 중에서 가장 적은 전력을 소비하는 Luffa에 대해 더욱 저전력 하드웨어 구조를 제안하고 그 소비 전력을 분석했다. 다음 절에 이에 대해 자세히 살펴본다.

표 3. SHA-256 해쉬함수에 대한 소비전력 추정값
Table 3. Power consumption estimation for SHA-256

SHA-2 해쉬함수	동적 전력(dynamic power) (uW)
SHA-256	0.45

Ⅳ. 저전력 Luffa 해쉬함수 구조

4.1 Luffa 해쉬함수 구조 및 특성

그림 9는 K.Kobayashi[6]가 하드웨어 성능평가를 위해 사용한 Luffa 해쉬 하드웨어를 개략적으로 나타낸 구조도다. 256비트로 구성된 하나의 상태 블록을 처리하기 위한 Step 함수는 SubCrumb, MixWord, AddConstant의 세 단계로 구성되어 있는데, 기존에 구현된 Luffa의 경우 각각의 상태 블록 마다 Step 함수를 가지도록 구현하였다. 또한 초기 값이 다른 각각의 Constant Generator를 각 상태 블록마다 가진다. 이 구조의 경우 높은 throughput은 달성할 수 있지만, 각각의 상태마다 Step 함수를 가지기 때문에, 칩의 면적을 많이 차지하며, 구성되는 조합회로가 크고 복잡해진다. Luffa 해쉬값 출력은 각 상태 블록을 XOR하여 출력하게 된다. 기존 구조에서는 이를 위

해 상태 블록의 출력값을 그대로 XOR을 통하여 출력을 하게 된다. 하지만, 해쉬 연산 중 매 Step 마다 상태 블록이 갱신되기 때문에, 매 Step 이 반복될 때 마다 출력 포트를 통해 중간 과정 값이 불필요하게 출력되고 있다. 이러한 구조는 불필요한 스위칭을 야기하여 전력 소모에 좋지 않은 영향을 미칠 수 있다.

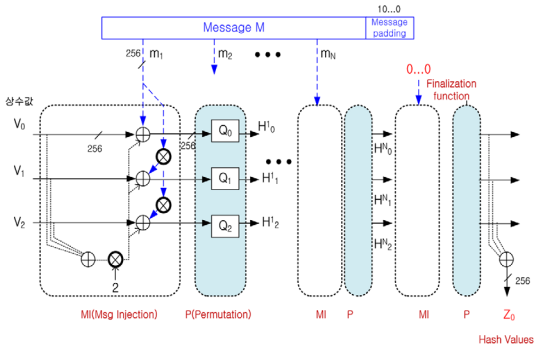

그림 9. Luffa 해쉬함수 구조도
Fig. 9. Architecture for Luffa hash function

4.2 저전력 Luffa 해쉬함수 하드웨어 구조

그림 10은 제안하는 저전력의 Luffa 해쉬 하드웨어 구조도를 보여주고 있다. 세 개의 상태 블록이 Step 함수를 공유하도록 하였으며, 한 클럭에 처리되었던 SubCrumb / MixWord, AddConstant는 두 단계로 나누어 수행될 수 있도록 하였다. 따라서 throughput은 기존의 구조에 비해 떨어지지만, 세 블록이 Step 함수 하나를 공유하기 때문에, 해당 부분의 면적이 줄어들며, 조합회로가 간단해진다는 장점을 가진다. 또한 각 Step 함수 블록은 AddConstant 단계에서 사용되는 상수를 생성해주는 블록을 가지는데, 이 블록의 경우 세 개의 상태에서 같은 구조를 가지지만, 각각 다른 초기 값을 가진다. 따라서 Constant Generator를 구현할 때 블록은 공유하되 초기 값을 변경할 수 있도록 하여 해당 블록을 세 개의 상태에 대하여 공유할 수 있도록 하였다. 또한 해쉬값이 출력되는 부분에 있어서도 출력을 제어하여 해쉬 계산이 완료되었을 때만 결과 값을 출력하도록 하여 불필요한 출력 포트의 스위칭을 제거하고자 하였다.

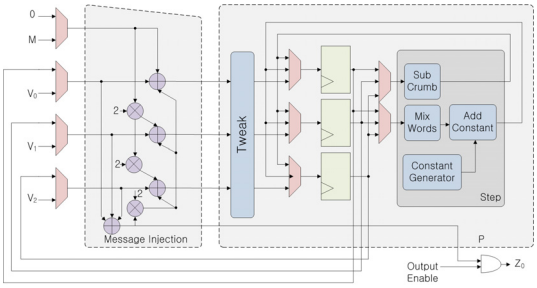

그림 10. 제안된 저전력 Luffa 해쉬 하드웨어 구조
Fig. 10. Low power Luffa hash hardware architecture

4.3 저전력 Luffa 해쉬함수 소비전력 특성 분석

표 3에 기존의 Luffa 하드웨어 구조와 제안하는 구조의 동적 전력 소모량의 추정값을 비교하였다. 제안하는 구조는 기존의 구조에 비해 약 10% 정도의 동적 전력을 덜 소모하는 것으로 나타났다. 기존의 Luffa 하드웨어의 구조와 제안하는 구조의 가장 큰 차이점은 각각의 상태 블록에서 사용된 Step 함수 구조를 세 개의 상태 블록이 공유하여 사용한다는 점이다. 표 2에서 제시된 내용을 참고하여 설명하자면, 제안하는 구조의 경우 기존의 구조에 비해 조합회로의 수가 약 1/3 정도 줄어들었을 것이라 예상되며(C4, C5, C6), MI 단계 이후 Step 함수가 실행될 때에는 하나의 상태 블록이 처리가 끝이 나면 다음 블록에 대한 처리를 하므로, Step 함수를 실행할 때 하나의 상태 블록만 값이 변화하므로 변화하는 레지스터의 수가 기존 구현에 비해 적다고 할 수 있다(C2).

표 4. 저전력 Luffa SHA-3 해쉬함수에 대한 소비전력 추정값
Table 4. Power consumption estimation for low power Luffa SHA-3 hash function

SHA-3 해쉬함수	동적 전력 (dynamic power) (uW)
기존 Luffa 하드웨어 구조[6]	0.567
제안하는 저전력 Luffa 하드웨어 구조	0.506

각 Step 블록에서 사용하고 있는 Constant Generator는 내부에 LFSR 구조를 포함하고 있기 때문에 하드웨어 내부에 레지스터를 가지고 있다. 세 개의 상태 블록이 하나의 Constant Generator를 공유하고 있으므로, 이 부분에서 필요한 레지스터의 양을 1/3로 줄일 수 있었다. 기존 구현에 비해 레지스터를 적게 사용할 수 있었기 때문에, 전력 소비량 감소에 영향을 준 것이라고 볼 수 있다(C1).

또한 해쉬값 출력 포트의 출력값을 Output Enable 신호로 제어하여 해쉬 연산이 완료되었을 때만 각 상태를 XOR하여 출력할 수 있도록 하여 불필요한 스위칭을 제거함으로써 전력 소모량 감소에 영향을 주었다고 볼 수 있다(C7).

V. 결 론

본 논문에서는 무결성 및 인증, 서명 등과 같은 핵심 보안 서비스를 제공하는 SHA-3 해쉬함수의 전력 특성을 분석하고 Luffa 해쉬함수에 대해서는 저전력 구조를 제안했다. 기존의 해쉬함수(SHA-1, SHA-2)가 안전성이 위협받게 되자 새롭게 SHA-3 해쉬함수 선정 작업이 진행되고 있는데, 이 중에서 본 논문에서는 하드웨어 및 소프트웨어로 구현했을 때, 높은 성능을 가진다고 알려진 네 가지 해쉬함수(Luffa, Keccak, Fugue, Grøstl)에 대한 소비 전력을 분석했다. 본 논문에서 사용한 전력 추정 방법은 하드웨어 설계를 통한 RTL 레벨 소비 전력 추정 방법이다.

본 방법은 SPICE를 사용한 트랜지스터 수준 시뮬레이션과 비교해 볼 때, 약 80% 정도의 정확도를 가진다고 알려져 있기 때문에[7] 본 논문에 제시된 소비 전력 값은 해쉬함수의 개략적인 소비 전력 특성을 충분히 파악할 수 있는 수준이다. 또한, 본 논문에서는 하드웨어에서의 전력을 많이 소비하는 요인을 체계적으로 정의했으며, 정의된 전력 소비 요인 관점에서 SHA-3 함수 소비 전력 분석 결과를 보였다.

분석과 실험을 통해, Luffa 해쉬함수가 전력 소비 차원에서 가장 효율적이라는 결과를 얻었다. 이에, Luffa 해쉬함수는 안전도 및 하드웨어/소프트웨어 성능, 소비 전력 관점에서 우수한 해쉬함수라고 할 수 있다. SHA-3 해쉬함수는 SHA-1이나 SHA-2보다 소비 전력이 많은 것으로 알려져 있다. 이는 SHA-3의 안전도를 높이기 위해 복잡한 암호학적인 기법을 사용하기 때문이다. 이 때문에 본 논문에서는 전력 소비가 적은 SHA-3를 얻기 위해, Luffa 해쉬함수에 대한 저전력 하드웨어 구조를 제안했다. 그 결과 비록 SHA-2보다는 소비 전력이 많지만 기존의 결과[6]보다는 약 10% 정도 소비 전력을 줄일 수 있었다.

참고문헌

[1] Cryptographic Hash Project, available at http://csrc.nist.gov/groups/ST/hash/index.html

[2] Kris Gaj, Ekawat Homsirikamol, Marcin Rogawski: Fair and Comprehensive Methodology for Comparing Hardware Performance of Fourteen Round Two SHA-3 Candidates Using FPGAs. CHES 2010: 264-278

[3] D. Soundris, C.Piquet, C Goutis, Designing CMOS Circuits for Low Power, Springer Verlag, 2010.

[4] Power estimation tutorial with Synopsys tools, http://www.tkt.cs.tut.fi/tools/public/tutorials/synopsys/pwr_est/gspe.html#rtl_pwr_est_flow

[5] Synopsys, "Power Compiler User Guide", https://solvnet.synopsys.com/dow_retrieve/E-2010.09/ni/power.html#Power%20Compiler

[6] K.Kobayashi, J.Ikegami, S. Matsuo, Evaluation of Hardware Performance for the SHA-3 Candidates Using SASEBO-GII, http://www.iacr.org

[7] http://www.synopsys.com, Power Compiler Reference Manual

저자소개

김성호(Sungho Kim)

인하대학교 전자공학과 석사
한양대학교 공과대학 융합전자공학부 박사과정
※관심분야 : 정보보호, 유비쿼터스 센서네트워크

조성호(Sungho Kim)

1989-1992 한국전자통신연구원 (ETRI) 선임연구원
1992-1997 한양대학교 전자공학과 조교수
1997-2002 한양대학교 정보통신대학 부교수
2002-현재 한양대학교 공과대학 융합전자공학부 교수
※관심분야 : 정보보호, 임베디드 시스템, 유비쿼터스 센서네트워크