情 報 保 護 學 會 誌
第19卷 第4號, 2009. 8

해쉬 함수 SHA-3 개발 동향

이 유 섭*, 이 제 상*, 강 진 건*, 홍 석 희*, 성 재 철**

요 약

2005년 중국의 Wang 교수 연구팀에 의해 SHA-1에 대한 충돌쌍 공격이 발표됨에 따라, SHA-1대신 SHA-2를 사용하도록 하였다. 아직까지 SHA-2에는 SHA-1과 같은 문제점이 발생하지 않고 있지만, SHA-1과 설계 논리가 유사한 SHA-2에 문제점이 생겼을 경우 대체 알고리즘이 부재한 현 상황에 따라 SHA-3 알고리즘 개발의 필요성이 제기되었다. 이에 미국 국립기술 표준원 (NIST, National Institute of Standards and Technologies)는 신규 표준 해쉬 알고리즘을 개발을 위하여 2007년부터 2012년까지 6년간의 "SHA-3 프로젝트"를 시작하였다. 2008년 11월 1일 64개의 알고리즘이 제출되었으며, 12월 11일 51개의 알고리즘이 1 후보 알고리즘으로 선정되었다. 2009년 7월 현재, 10개의 알고리즘이 제안자에 의해 철회되어 41개의 알고리즘이 1 라운드에서 심사되고 있다. 본 논문에서는 SHA-3 개발의 요구 사항과 현재까지 SHA-3 개발 동향을 서술한다.

Ⅰ. 서 론

미국 국립기술표준원 (NIST, National Institute of Standards and Technologies)은 신규 표준 해쉬 알고리즘 SHA-3 개발을 목표로 현재 진행 중에 있다[6]. SHA-3는 NIST에 의해 공식 문서로 발간될 네 번째 알고리즘이며, 이미 앞서 SHA-0, SHA-1, SHA-2 알고리즘이 각각 1993년, 1995년, 2002년에 FIPS PUB 180[9], FIPS PUB 180-1[10], FIPS PUB 180-2[11]에 발표되었다.

이미 SHA-0,1,2가 있는데도, NIST에서 SHA-3를 새롭게 개발하고자 하는 직접적인 이유는 1998년 SHA-0에 대한 충돌쌍 공격[3]이 Chabaud와 Joux에 의해 소개된 이후 전 세계 암호학자들이 SHA-1를 분석하려고 시도하였지만 그 어떤 취약성도 발견되지 않아 SHA-1은 안전한 해쉬 알고리즘이라고 믿어져 왔다. 그러나 2005년 중국의 Wang 교수 연구팀에 의해 SHA-1에 대한 충돌쌍 공격[17],[18]이 발표됨에 따라, NIST는 SHA-1에 기반한 전자서명 알고리즘 사용을 중단하는 한편, SHA-1 대신 SHA-2를 사용하도록 조치를 취하였다[7]. 아직까지 SHA-2는 SHA-1과 같은 문제점이 발생하지 않고 있지만, SHA-1과 설계 논리가 유사한 SHA-2에 문제점이 생겼을 경우, 대체 알고리즘이 부재한 현 상황에 따라 SHA-3 알고리즘 개발의 필요성이 제기되었다.

SHA-0,1,2 와는 달리 SHA-3 개발은 전 세계 연구기관들로부터 제안된 후보 알고리즘들 중에서 선정한다는 원칙하에 수행된다. 기존 SHA-0,1,2는 NIST 내부에서 자체적으로 개발되고 공표된 해쉬 알고리즘들이기 때문에 충분한 안전성 검증이 없는 상태에서 표준화가 되었다. 해쉬 알고리즘은 키 생성 알고리즘[14], 전자서명[12], 난수생성기[15], 메시지인증코드[1],[13] 등 여러 응용환경에서 사용되기 때문에, 해쉬 알고리즘의 취약성이 곧바로 응용환경에서의 취약성으로 연결될 수 있다는 점에 있어서 안전하면서도 효율적인 해쉬 알고리즘의 개발은 매우 중요하다고 할 수 있다. 또한 2005년 SHA-1의 취약성이 발견됨에 따라, SHA-3 개발은 소수의 암호학자들에 의해 수행되기보다는 전 세계 암호 연구기관들로부터 후보 알고리즘을 공개 모집하여 선정하는 방법으로 수행 된다.

SHA-3 선정 과정에 참여하는 인력은 크게 두 가지로 나누어진다. 첫째로, NIST 소속 직원 및 초청 연구원들

이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임 (No. 2009-0060420).
* 고려대학교 정보경영공학 전문 대학원 ({yusubi, jslee, jkkang, hsh}@cist.korea.ac.kr.hong@cist.korean.ac.kr)
** 서울시립대학교 수학과 (jcsung@uos.ac.kr)

이다. "SHA-3 개발 프로젝트"에 참여하고 있는 인력들은 SHA-3 개발과 관련된 전문가들로 이루어져 있다. SHA-3 선정 과정의 공정성과 투명성을 높이기 위하여, NIST의 참여 인력들은 SHA-3 후보 알고리즘과는 어떠한 관련도 없는 중립적인 인물들로 선출 되었다. NIST 내부 인력은 고정된 것은 아니며, 필요시 외부 인력을 스카웃할 수 있도록 하고 있다. 둘째로, SHA-3 개발이 공개경쟁을 통해 진행되는 것이기 때문에 전 세계 암호학자들이 모두 SHA-3 개발에 참여 할 수 있다. 암호관련 학술대회 또는 홈페이지를 통해 SHA-3 후보 알고리즘에 대한 분석 결과를 발표할 수 있으며, 이는 SHA-3 선정 과정에 직접적인 영향을 주게 된다.

SHA-3 선정은 지금까지 제안된 그리고 앞으로 제시될 분석 및 안전성 증명 기술에 근거하여 객관적인 검증 과정을 통해 이루어지게 된다. 반면, 현재 시점에서 해쉬 알고리즘에 대한 설계 논리, 안전성 분석 기술 및 이론에 대한 이해가 아직은 깊지 않는 상황이다. 2005년 Wang의 분석 기술이 소개되기 이전까지는, 일반적인 해쉬 알고리즘 분석 기법이 제안되지 않고 특정 알고리즘 각각에 대한 분석 결과만이 소개된 정도였다. 블록암호의 경우를 보더라도 1990년대 초반에 차분 공격 (Differential Attack)[2]과 선형 공격 (Linear Attack)[5]이 소개된 이후, 지금까지도 새로운 분석 기법들이 제안되고 있는 현실을 감안할 때, 해쉬 알고리즘의 경우에 대해서도 현재 시점에서 분석 기술 및 이론에 대한 연구가 더욱 절실히 요구되고 있다. 무엇보다 SHA-3 개발 및 선정은 분석 기술 및 설계 이론과 맞물려서 이루어질 수밖에 없기 때문에, 충분한 시간을 갖고 신중하게 SHA-3를 최종 선정해야 한다.

본 논문은 다음과 같이 구성된다. 제 2 장에서는 1 라운드 후보 알고리즘 선정, 2 라운드 후보 알고리즘 선정, SHA-3최종 선정 과정과 관련된 요구 조건 및 평가 기준에 대해 서술한다. 제 3장에서는 1 라운드 후보로 선정된 41개 알고리즘에 대한 특징과 현재까지 이루어진 분석 결과를 제시한다. 마지막으로 제 4 장에서 결론을 맺는다.

Ⅱ. SHA-3 개발 과정

SHA-3 개발 프로젝트는 2007년부터 2012년까지 총 6년 동안 진행된다. 최종적으로 선정된 한 개의 알고리즘을 SHA-3로 선정하며, 만약 선정된 SHA-3에 심각한 취약성이 발견될 경우, 다른 4 개의 최종 후보 알고리즘들 중에서 SHA-3를 선택한다. 구체적으로, SHA-3를 선별해내는 과정은 다음과 같이 다섯 단계로 나뉜다.

- SHA-3 개발 가이드라인 수립 : 2007년 1월 23일 SHA-3 후보 알고리즘을 위한 기본 요구 사항 및 평가 기준 자료 배포. 이후 전 세계 여러 연구 기관으로부터 제공된 코멘트를 토대로 2007년 11월 2일에 최종 SHA-3 개발 가이드라인을 발표함.

- 해쉬 알고리즘 공모 및 1 라운드 후보 알고리즘 선정 : 2008년 10월 31일에 공모를 완료하였으며 총 64개의 알고리즘이 접수되었고, 2008년 12월 11일에 1 라운드 후보 알고리즘으로 선정된 51개 후보 알고리즘이 발표되었으며 2009년 7월 현재, 10개의 알고리즘이 제안자들에 의해 철회되어 41개의 후보 알고리즘이 심사 중.

- 2 라운드 후보 알고리즘 선정 : 41개의 1 라운드 후보 알고리즘 중에서, 약 15개의 2 라운드 후보 알고리즘들이 2009년 8월에 개최되는 국제 암호학회 Crypto 2009이전까지 선정될 예정임. 2 라운드 시작 이전까지, 선정된 2 라운드 후보 알고리즘에 대해 알고리즘 및 구현 결과에 대한 변경 또는 수정 기회를 제공함.[1)]

- 최종 후보 알고리즘 선정 : 2010년 하반기에 대략 5 개 최종 후보 알고리즘을 선정할 예정. 최종 라운드 시작 이전까지, 선정된 최종 후보 알고리즘에 대해 알고리즘 및 구현 결과에 대한 변경 또는 수정 기회를 제공함.[2)]

1) 2 라운드 후보 알고리즘 선정 이후, 변경 또는 수정 사항을 제출할 수 있는 구체적인 기간에 대한 문서 기록은 없으나 몇 개월 밖에 제공되지 않을 것으로 예상됨. 단, 변경 또는 수정 사항이 크면 안 됨. 또한, 제안자들에 의해 제출된 변경 또는 수정 사항을 받아들일지에 대한 여부는 NIST에 의해 최종 결정될 것임.

2) 최종 후보 알고리즘 선정 이후, 변경 또는 수정 사항을 제출할 수 있는 구체적인 기간에 대한 문서 기록은 없으나 몇 개월 밖에 제공되지 않을 것으로 예상됨. 단, 변경 또는 수정 사항이 크면 안 됨. 또한, 제안자들에 의해 제출된 변경 또는 수정 사항을 받아들일지에 대한 여부는 NIST에 의해 최종 결정될 것임.

○ SHA-3 선정 : 2012년 4월에서 6월 사이에 발표할 예정임.

위의 각 단계별 선정 과정을 돕기 위해 다음과 같이 세 차례 컨퍼런스를 개최한다.

○ 1 차 컨퍼런스 : 2008년 2월 25일부터 28일까지 벨기에의 K.U. Leuven의 University Hall에서 개최됨. 51개의 1 라운드 후보 알고리즘에 대해 각 제안자들에 의한 발표 및 질의응답 시간을 가짐. NIST 관계자들이 향후 일정 및 2 라운드 후보 알고리즘 선정과 관련된 고려 사항들에 대한 내용을 발표함[8].

○ 2 차 컨퍼런스 : 2010 년 4 월에서 6 월 사이에 개최할 예정. 이는 최종 후보 알고리즘을 선정하기 이전에 최종적으로 의견을 조율하기 위한 자리임.

○ 3 차 컨퍼런스 : 2012 년 1 월에서 3 월 사이에 개최할 예정. 최종 후보 알고리즘 제안자들이 자신들이 제출한 알고리즘들에 대한 최종적인 입장 및 서로의 의견에 대한 토의. 본 컨퍼런스는 SHA-3 선정 바로 직전에 이루어짐.

본 장에서는 SHA-3 선정 과정의 기준을 서술한다. 먼저 1 절에서는 1 라운드 후보 알고리즘 선정 기준을 살펴보고, 2 절에서는 2 라운드 후보 알고리즘 선정 기준, 이어서 3 절에서는 최종 후보 알고리즘 선정 기준, 끝으로 4 절에서는 SHA-3 알고리즘 선정 기준을 기술한다.

2.1 라운드 후보 알고리즘 선정 기준

1 라운드 후보 알고리즘 선정을 위한 최종 가이드라인은 NIST 홈페이지를 통하여 2007년 11월 2일에 발표되었으며, 기본 요구 조건을 충족하기만 하면 제안된 알고리즘은 1 라운드 후보 알고리즘으로 선정된다. 기본 요구 조건은 다음과 같이 Complete와 Proper 로 구성된다[16].

○ Complete : 주어진 API를 사용하여 레퍼런스 C 코드 및 최적화된 C 코드 구현 결과 제시, NIST 에서 요구한 알고리즘 테스트 결과를 제시하고 상세 기술 문서를 제출한다.

○ Proper : 로열티 없이 전 세계적으로 누구나 사용 가능해야 하고, 다양한 하드웨어 및 소프트웨어 구현 환경에서 구현 가능한다. 또한, 224, 256, 384, 512-비트의 출력 크기를 제공해야 하며 처리 가능한 메시지의 비트 길이의 최대 크기가 최소한 $2^{64}-1$ 이상이어야 한다.

2008년 10월 31일까지 64개의 알고리즘이 등록 되었다. 최종 등록된 64 개의 알고리즘 중 Complete와 Proper를 만족시키는 51 개의 알고리즘이 1 라운드 후보 알고리즘을 선정 되어 2008년 12월 11일에 발표되었다.

"Complete"를 만족시키기 위해서는 다음의 다섯 가지 자료를 2008 년 10 월 31 일까지 NIST에 도착하도록 항공 우편으로 제출해야 한다. 제출할 자료는 알고리즘 제안자 및 개발 참여자 소개서 (Cover Sheet), 알고리즘 명세서 및 보충 문서 (Algorithm Specifications and Supporting Documentation), 구현 및 관련 모든 자료를 CD 또는 DVD 한 장에 저장 (Optical Media), 서명된 지적 재산권 관련 동의 자료 (Intellectual Property Statements/Agreements/Disclosures), 일반적인 제출 요구 사항 (General Submission Requirements) 으로 이루어진다. "Proper"의 경우는 간단하므로 자세한 설명을 생략한다.

2.2 라운드 후보 알고리즘 선정 기준

2 라운드 후보 알고리즘들은 2009년 8월에 열리는 Crypto 2009 이전까지 15개 정도로 압축되어 발표될 예정이다. 이는 2007년에 발표된 SHA-3 개발 가이드라인 문서에서 명시한 일정과는 많은 차이가 있다. 2007 년 문서에서는 2라운드 후보 알고리즘을 2010년 중순에 열릴 예정인 2차 컨퍼런스 이후에 5 개 정도로 압축하여 발표할 예정이었으나, 예상보다 많은 수의 알고리즘이 1 라운드 후보 알고리즘으로 선정되어, 한번에 5 개의 알고리즘으로 압축하기 보다는 먼저 15 개의 알고리즘을 분류한 후, 그 이후에 5개의 알고리즘을 선

별하는 것으로 방향을 선회한 것이다. 구체적으로, 2009년 Crypto 2009 이전까지 15 개로 줄인 후 2010년에 원래 예정대로 5개로 압축하는 것으로 계획이 수정되었다. 본 절에서는 Crypto 2009 이전까지 15개로 압축될 2 라운드 알고리즘 선정과 관련된 평가 기준에 대해 설명하고자 한다. 2 라운드 후보 알고리즘 선정 과정은 세 가지 평가 분야로 나누어 진행이 된다. 첫째, 안전성 (Security), 둘째, 비용 (Cost), 셋째, 알고리즘 및 구현 특징 (Algorithm and Implementation Characteristics)으로 나누어지며, 각각의 평가 기준에 대한 구체적인 내용은 다음과 같다.

2.2.1 안전성

알고리즘의 안전성 평가는 가장 중요한 평가 기준이며, 다음의 다섯 가지 요소에 대해 수행된다.

① 해쉬 알고리즘 응용 분야 : 전자서명 (FIPS 186-2), 키 생성 알고리즘 (NIST Special Publication 800-56A), 해쉬 기반 메시지 인증 코드 (FIPS 198), 결정적인 랜덤 비트 생성기 (Deterministic random bit generators, SP 800-90), NIST 또는 외부 전문가에 의해 제기될 부가적인 응용분야들에 대한 안전성을 검토함.

② HMAC, 의사난수 함수 (PRFs, Pseudo Random Functions), 랜덤마이징 해싱 (Randomizing Hashing) 구성 시 특정 요구 사항 : 후보 알고리즘은 의사난수 함수 구성 방법으로 HMAC을 지원해야 하며, 부가적으로 HMAC 기반 아닌 의사난수 함수 구성 방법과 랜덤마이징 해싱 기법을 제안할 수 있음. 단, 다음의 안전성 기준을 충족시켜야 함 (n 비트 출력 크기인 경우).

- 구별 공격 에 대한 안전성 : 2n/2 개 보다 작은 수의 질문을 사용하여, 역상 탐색 공격 보다 작은 수행 시간을 갖는 공격 방법이 존재하지 않아야 함.
- 부가적인 의사난수 구성 방법의 안전성 : 제안 알고리즘 명세서에 기록된 안전도를 충족시켜야 함.
- 랜덤마이징 해싱 기법에 대한 안전성 : 공격에 이용되는 메시지 (target message) 길이가 최대 2k 비트인 경우에 대해, n-k 비트 안전성 제공해야 함.

③ 해쉬 알고리즘 안전성 요구 사항 : 역상 저항성 등과 같이 해쉬 함수 자체에 대한 안전도를 명시하고 있으며, 이에 대한 요구 사항은 다음과 같음. (n 비트 출력 크기인 경우).

- n/2 비트 안전성 : 충돌저항성.
- n 비트 안전성 : 역상 저항성, 메시지 길이 확장 공격에 대한 안전성 (length-extension security).
- n-k 비트 안전성 : 공격에 이용되는 메시지 (target message) 길이가 최대 2k 비트인 경우에 대해, 제 2 역상 저항성.
- m 비트 절단 (truncation) : 위의 세 가지 안전성 요구 조건에서 n 대신 m으로 대체하였을 때의 안전성 제공.
- 트랩도어 (trap-door) 유무 : 트랩도어 존재 가능성이 없음을 논리적으로 설명할 수 있어야 함.
- 안전성/효율성 간의 trade-off 제시 : 제 2 역상 탐색 공격 또는 다중 충돌쌍 탐색 공격에 대한 안전성을 높이는 방안을 제시한 후보 알고리즘에 해당.

④ 여러 공격의 안전성과 관련된 평가 사항 : 랜덤하지 않은 특성이 없어야 함, 통계 테스트에 통과해야 함, 기존의 분석 기법뿐만 아니라 향후 개발될 분석 기법에 대해서도 평가가 이루어질 것임.

⑤ 그 밖의 고려 사항 : 후보 알고리즘 제안자에 의해 수행된 안전성 증명 및 분석 수준, 알고리즘 기술의 명확성, 알고리즘의 단순성, 후보 알고리즘의 안전성에 대한 NIST와 암호 커뮤니티의 신뢰도.

2.2.2 비용

1 라운드 후보 알고리즘의 효율성 및 메모리 사용량을 분석하기 위하여, 제안자들이 제출한 최적화된 구현 결과를 이용한다.

① 구현 효율성 : 다양한 구현 환경과 여러 메시지 입력 길이에 대하여, 후보 알고리즘의 구현 속도 분석. 1 라운드 후보 알고리즘의 경우, 주로 소프트

웨어 구현 결과에 초점을 맞춤. 하드웨어 구현 결과는 2 라운드에서 집중적으로 검토할 예정.

② 메모리 요구 사항 : 소프트웨어 구현 시, 알고리즘 코드 크기와 램 (RAM) 사용량을 분석함.

2.2.3 알고리즘 및 구현 특징

다음과 같이, 탄력성 (Flexibility)과 단순성 (Simplicity)으로 나누어 평가가 진행된다.

① 탄력성 : 좀 더 큰 탄력성을 가진 알고리즘은 그렇지 않은 알고리즘에 비해 많은 사용자들의 호응을 얻게 될 것임. 다음은 탄력성에 관련된 요소들임.
- 조정 가능한 패러미터 : 안전성과 구현 속도 간의 trade-off 결과 값 제시.
- 다양한 구현 환경에 적용 가능 : 스마트카드와 같은 구현 환경을 포함하여, 안전하면서도 효율적으로 구현될 수 있음.
- 병렬 처리 가능 : 구현 속도 향상.

② 단순성 : 상대적으로 단순한 설계 논리에 바탕을 둔 알고리즘을 우선시 함.

2.3 최종 후보 알고리즘 선정 기준

5 개의 최종 후보 알고리즘 선정은 2010 년 중순에 개최될 2 차 컨퍼런스 이후에 이루어질 예정이다. 최종 후보 알고리즘 선정 기준은 15 개의 2 라운드 후보 알고리즘 선정 기준 외에 하드웨어 구현관련 평가 기준이

〔표 1〕 안전성 분석 결과

알고리즘	분석 결과	알고리즘	분석 결과
ARIRANG		Keccak	
AURORA	제 2 역상(C), 충돌쌍(B)	LANE	
BLAKE		Lesamnta	
Blender	역상(C), 충돌쌍(C)	Luffa	
Blue Midnight Wish		LUX	제 2 역상(C), 충돌쌍(C)
Cheetah	메시지 길이 확장 공격	MCSSHA-3	제 2 역상(B), 충돌쌍(C)
CHI		MD6	
CRUNCH	메시지 길이 확장 공격	NaSHA	충돌쌍(B,C)
CubeHash	역상 공격(A)	SANDstorm	
Dynamic SHA	충돌쌍(C), 메시지 길이 확장 공격	Sarmal	역상 공격(B)
Dynamic SHA2	메시지 길이 확장 공격	Sgàil	충돌쌍(D)
ECHO		Shabal	
ECOH	제 2 역상 공격(C)	SHAvite-3	
Edon-R	역상 공격(B)	SIMD	
EnRUPT	역상(B), 충돌쌍(D)	Skein	
ESSENCE		Spectral Hash	충돌쌍(D)
FSB		SWIFFTX	
Fugue		TIB3	충돌쌍(B)
Grøstl		Twister	역상(C), 제 2 역상(C), 충돌쌍(A)
Hamsi		Vortex	역상(B), 충돌쌍 공격(A)
JH	역상 공격(A)		

(A) : 공격에 필요한 압축함수 연산 횟수 < 일반적인 공격에 필요한 압축함수 연산 횟수
(B) : 공격에 필요한 압축함수 연산 횟수 < (일반적인 공격에 필요한 압축함수 연산 횟수)×1/n
(C) : 시간복잡도 × 메모리 사용량 < (일반적인 공격에 필요한 압축함수 연산 횟수)×1/n
(D) : 공격이 되었다는 실제적인 예를 든 경우

추가된다. 안전성과 알고리즘 및 구현 특징에 대한 선정 기준은 1 라운드 선정 기준과 동일하다. 본 절에서는 하드웨어 구현 관련 평가 기준에 대해 자세히 살펴보고자 한다.

2.3.1 비용

2 라운드 후보 알고리즘 선정 시, 주로 소프트웨어 구현 결과 분석에 초점을 맞춘 반면, 5 개의 최종 후보 알고리즘 선정을 위하여, 소프트웨어 구현과 더불어 하드웨어 구현 분석에 초점을 맞춘다. 이를 위해 15 개의 2 라운드 후보 알고리즘 제안자들은 2 라운드를 진행하기 이전에, 최적화된 구현 결과를 제출할 수 있다.

① 구현 효율성 : 2 라운드를 시작하기 전, 제안자들에 의해 새롭게 제출된 구현 결과물을 기초로, 소프트웨어 뿐 아니라 하드웨어 구현 효율성을 분석한다. 제안자들은 비독점 (nonproprietary) 하드웨어 언어 (Hardware Description Language, HDL)로 작성된 구현 결과물을 제출할 수 있다.

② 메모리 요구 사항 : 소프트웨어 구현 시, 알고리즘 코드 크기와 램 (RAM) 사용량을 분석하고, 하드웨어 구현 시, FPGA 또는 ASIC의 경우에 대한 게이트 사용량을 분석한다.

2.4 SHA-3 선정

최종 SHA-3 알고리즘 선정은 2012년 상반기에 이루어질 예정이며, SHA-3 최종 선정을 위하여 2012년 1월에서 4월 사이에 개최될 최종 3 차 컨퍼런스에서 5개의 최종 후보 알고리즘 제안자들이 각 알고리즘에 대해 최종 의견을 발표한 후, SHA-3를 선정할 것이다. 이어서 2012년 하반기에 SHA-3 표준 문서 초안을 발표하고, 이와 관련된 여러 의견을 종합하여 최종적으로 2012년 말 경에 SHA-3 최종 문서를 발간할 예정이다.

Ⅲ. 1 라운드 후보 알고리즘 분류

총 64 개의 후보 알고리즘이 제안되었으며, 이중 "Complete"와 "Proper" 기준을 충족시키는 51개의 알고리즘이 1 라운드 후보 알고리즘으로 선정되었다. 2009년 7월 현재, 10개의 알고리즘이 제안자에 의해 철회되어 41개의 알고리즘이 1 라운드에서 심사되고 있다. 본 장에서는 41 개의 1 라운드 후보 알고리즘을 간략히 소개한다. 1 절에서는 현재까지의 안전성 분석 결과를 제시하고, 2 절에서는 1 라운드 후보 알고리즘을 설계 논리 관점에서 분류한 뒤 각 경우에 대한 특징을 설명한다. 3 절에서는 구현 결과 관점에서 각 알고리즘을 분류하고, 각 경우에 대한 특징을 설명한다.

3.1 절 안전성 분석 결과

SHA-3 후보 알고리즘들의 안전성 분석 관련 정보는 ECRYPT II 프로젝트 (2008년 8월 1일 시작, 4년 동안 진행)에서 관리하는 SHA-3 ZOO라는 홈페이지에서 관리되고 있다. SHA-3 ZOO 홈페이지를 보면, 2009 년 7월까지 41 개의 1 라운드 후보 알고리즘 중, 안전성에 문제가 발견된 알고리즘은 총 21 개로 집계되어 있다. 여기서 안전성에 문제가 있다는 뜻은, NIST에서 요구하고 있는 안전성 개념의 요구 조건을 조금이라도 충족시키지 못한다는 것을 의미한다. 구체적으로 21개의 알고리즘들은 역상 저항성, 제 2 역상 저항성, 충돌저항성, 메시지 길이 확장 공격 중 최소 한 가지 이상에서 취약점을 갖고 있다. 본 절에서는 SHA-3 ZOO에서 언급된 안전성 분석 결과를 설명 한다. [표 1]은 각 후보 알고리즘이 갖고 있는 취약성을 나타낸다.

3.2 구현 효율성에 따른 분류

본 절에서는 각 알고리즘의 32 비트, 64 비트 소프트웨어 구현 효율성을 SHA-2의 구현 효율성을 기준으로 비교 한다. Fleischmann, Forler, Gorski의 결과를 바탕으로 작성되었다. [표 2]는 각 알고리즘의 구현 효율성을 나타내며, 이때의 구현 속도 값은 제안자들에 의해 제안된 구현 결과를 이용하고, SHA-2의 경우는 NIST 표준 문서에 나온 결과를 이용한다. [표 3]는 알고리즘의 구현 속도를 SHA-2의 속도와 비교하여 AA, A, B, C, D, E로 분류하는 기준이다.

Ⅴ. 결 론

SHA-3 최종 알고리즘을 선정하는 데에 있어서 가장 중요한 요소는 안전성의 검증에 있다. 이와 동시에 효율

〔표 2〕 알고리즘 구현 속도 비교

알고리즘	32 비트 구현	64 비트 구현	알고리즘	32 비트 구현	64 비트 구현
	cpb	cpb		cpb	cpb
SHA-256 SHA-512	29.3(C) 55.2(C)	20.1(C) 13.1(C)	JH-256 JH-512	21.3(B) 21.3(AA)	16.8(B) 16.8(D)
ARIRANG-256 ARIRANG-512	20(A) 14.9(AA)	55.3(E) 11.2(B)	Keccak-256 Keccak-512	35.4(C) 68.9(C)	10.1(A) 20.3(D)
AURORA-256 AURORA-512	24.3(B) 46.9(B)	15.4(B) 27.4(E)	LANE-256 LANE-512	40.4(D) 152.2(E)	25.6(D) 145.3(E)
BLAKE-32 BLAKE-64	28.3(B) 61.7(C)	16.7(B) 12.3(B)	Lesamnta-256 Lesamnta-512	59.2(E) 54.5(B)	52.7(E) 51.2(E)
Blender Blender	105.8(E) 122.4(E)	105.8(E) 164.2(E)	Luffa-256 Luffa-512	13.9(AA) 25.5(AA)	13.4(A) 23.2(D)
BMW-256 BMW-512	8.6(AA) 13.37(AA)	7.85(AA) 4.06(AA)	LUX-256 LUX-512	16.7(A) 14.9(AA)	28.2(D) 12.5(B)
Cheetah-256 Cheetah-512	15.3(A) 83.8(D)	10.5(A) 15.6(C)	MCSSHA-3	-	-
CHI-256 CHI-512	49(C) 78(D)	26(D) 16(C)	MD6-256 MD6-512	68(E) 106(D)	28(D) 44(E)
CRUNCH-256 CRUNCH-512	29.9(C) 86.4(D)	16.9(B) 46.9(E)	NaSHA-256 NaSHA-512	39(D) 38.9(A)	28.4(D) 29.3(E)
CubeHash8/1	200(E)	148(E)	SANDstorm-256 SANDstorm-512	62.5(E) 296.8(E)	36.5(D) 95.3(E)
Dynamic SHA-256 Dynamic SHA-512	27.9(B) 47.2(B)	27.9(D) 47.2(E)	Sarmal-256 Sarmal-512	19.2(A) 23.3(AA)	10(A) 12.6(B)
Dynamic SHA2-256 Dynamic SHA2-512	21.9(B) 67.3(C)	21.9(C) 67.1(E)	Sgàil	-	61(E)
ECHO-256 ECHO-512	38(D) 83(D)	32(D) 66(E)	Shabal-256 Shabal-512	18.4(A) 18.4(AA)	13.5(A) 13.5(C)
ECOH	- -	- -	SHAvite-3-256 SHAvite-3-512	35.3(C) 55(B)	26.7(C) 38.2(E)
Edon-R-256 Edon-R-512	9.1(AA) 13.7(AA)	5.9(AA) 2.9(AA)	SIMD-256 SIMD-512	12(AA) 118(E)	11(A) 85(E)
EnRUPT-256 EnRUPT-512	8.3(AA) 5.1(AA)	8.3(AA) 5.1(AA)	Skein-256 Skein-512	21.6(A) 20.1(AA)	7.6(AA) 6.1(AA)
ESSENCE-256 ESSENCE-512	149.8(E) 176.5(E)	19.5(B) 23.5(D)	Spectral Hash	454.6(E)	454.6(E)
FSB-256 FSB-512	324(E) 507(E)	- -	SWIFFTX-256 SWIFFTX-512	57(D) 57(C)	
Fugue-256 Fugue-512	36.2(C) 74.6(D)	61(E) 132.7(E)	TIB3-256 TIB3-512	12.9(AA) 17.5(AA)	7.6(A) 6.3(AA)
Grøstl-256 Grøstl-512	22.9(B) 37.5(A)	22.4(D) 30.1(E)	Twister-256 Twister-512	35.8(C) 39.6(A)	15.8(B) 17.5(D)
Hamsi	-	-	Vortex-256 Vortex-512	46.2(D) 56(C)	69.4(E) 90(E)

〈표 3〉 알고리즘 구현 속도 분류 기준

속도	분류
$x < \frac{1}{2}$ SHA-2	AA
$\frac{1}{2}$ SHA-2 $\leq x <$ $\frac{3}{4}$ SHA-2	A
$\frac{3}{4}$ SHA-2 $\leq x <$ SHA-2	B
SHA-2 $\leq x <$ $\frac{5}{4}$ SHA-2	C
$\frac{5}{4}$ SHA-2 $\leq x \leq$ 2 SHA-2	D
$x >$ 2 SHA-2	E

적으로 구현될 수 있어야 한다. 안전성과 구현 효율성을 동시에 추구하는 것은 쉽지 않지만, NIST가 SHA-3 프로젝트를 진행하면서 알고리즘 제안자로 하여금 알고리즘을 수정할 수 있는 기회를 제공하고, 효율적인 구현 결과를 새롭게 제시할 수 있도록 한다는 점에 있어서 SHA-3으로 선정될 최종 알고리즘은 안전성과 효율성 관점에서 동시에 극대화할 수 있으리라고 본다.

NIST가 6년이라는 긴 기간을 통해 SHA-3를 선정하고 SHA-3 선정을 지원하기 위한 현재까지의 전 세계 여러 기관에서의 왕성한 연구 활동을 볼 때에, NIST가 공정하면서도 객관적으로 SHA-3를 선정할 것이라 생각된다.

참고문헌

[1] M. Bellare, R. Canetti and H. Krawczyk, "Keying Hash Functions for Message Authentication", Crypto 1996, LNCS 1109, pp. 1-15, 1996.

[2] Eli Biham, Adi Shamir, "Differential cryptanalysis of DES-like cryptosystems", Journal of Cryptology, 1991.

[3] Florent Chabaud, Antoine Joux, "Differential Collisions in SHA-0", Crypto 1998, LNCS 1462, pp. 56-71, 1998.

[4] Ewan Fleischmann, Christian Forler, Michael Gorski, "Classification of the SHA-3 Candidates", Cryptology ePrint Archive: Report 2008/511.

[5] Mitsuru Matsui, "Linear cryptanalysis method for DES cipher", Eurocrypt 1993, LNCS 765, pp. 386-397, 1994.

[6] NIST Homepage for Hash Project : http://csrc.nist.gov/groups/ST/hash/sha-3/.

[7] NIST Hash Policy : http://csrc.nist.gov/groups/ST/hash/policy.html.

[8] NIST 1 Round Hash Workshop : http://csrc.nist.gov/groups/ST/hash/sha-3/Round1/Feb2009/program.html.

[9] NIST, "FIPS 180" (superseded by FIPS 180-1 and FIPS 180-2). See also NIST's Secure Hashing site.

[10] NIST, "FIPS 180-1" (superseded by FIPS 180-2). See also NIST's Secure Hashing site.

[11] NIST, "FIPS 180-2: Secure Hash Standard (SHS)", August 2002 (change notice: February 2004). See also NIST's Secure Hashing site.

[12] NIST, "FIPS PUB 186-2: DIGITAL SIGNATURE STANDARD (DSS)", 27 January 2000.

[13] NIST, "FIPS PUB 198: The Keyed-Hash Message Authentication Code (HMAC)", 6 March 2002.

[14] NIST SP 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, http://csrc.nist.gov/publications/nistpubs/800-56A/SP800-56A_ Revision1_ Mar08-2007.pdf.

[15] NIST SP 800-90, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, http://csrc.nist.gov/publications/nistpubs/800-90/SP800-90revised_ March2007.pdf.

[16] NIST Hash Project, Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA-3) Family, http://csrc.nist.gov/groups/ST/hash/documents/SHA-3_FR_Notice_Nov02_ 2007%20-%20more%20readable%20version.pdf.

[17] X. Wang, A. C. Yao and F. Yao, "Cryptanalysis on SHA-1", CRYPTOGRAPHIC HASH WORKSHOP, October 31-November 1, 2005.

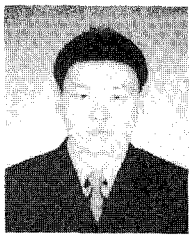
[18] X. Wang, Y. L. Yin and H. Yu, "Finding Collisions in the Full SHA-1", Crypto 2005, LNCS 3621, pp. 17-36, 2005.

〈著者紹介〉

이 유 섭 (Yuseop Lee)

학생회원

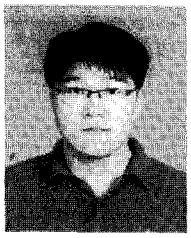
2007년 2월: 서울시립대학교 수학과 학사
2007년 3월~현재: 고려대학교 정보경영공학전문대학원 석박사 통합과정
<관심분야> 대칭키 암호 분석·설계



강 진 건 (Jinkeon Kang)

학생회원

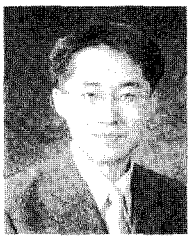
2007년 8월: 고려대학교 산업시스템정보공학과 학사
2007년 9월~현재: 고려대학교 정보경영공학전문대학원 석박사통합과정
<관심분야> 해쉬 함수 분석·설계



이 제 상 (Jesang Lee)

학생회원

2003년 2월: 고려대학교 수학과 학사
2006년 8월: 고려대학교 정보보호대학원 석사
2006년 9월~현재: 고려대학교 정보경영공학전문대학원 박사과정
<관심분야> 대칭키 암호 분석·설계



성 재 철 (Jaechul Sung)

종신회원

1997년 8월: 고려대학교 수학과 학사
1999년 8월: 고려대학교 수학과 석사
2002년 8월: 고려대학교 수학과 박사
2002년 8월~2004년 1월: 한국정보보호진흥원 선임연구원
2004년 2월~현재: 서울시립대학교 수학과 조교수
<관심분야> 암호 알고리즘 설계 및 분석

홍 석 희 (Seokhie Hong)

종신회원

1995년 2월: 고려대학교 수학과 학사
1997년 2월: 고려대학교 수학과 석사
2001년 2월: 고려대학교 수학과 박사
1999년 8월~2004년 2월: (주) 시큐리티 테크놀로지스 선임연구원
2003년 2월~2004년 2월: 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월: K.U.Leuven, ESAT/SCD-COSIC 박사후연구원
2005년 3월~2008년 8월: 고려대학교 정보보호대학원 조교수
2008년 9월~현재: 고려대학교 정보경영공학전문대학원 부교수
<관심분야> 대칭키 암호의 분석 및 설계, 컴퓨터 포렌식