정 보 보 호 학 회 지
제 32 권 제1호, 2022. 2

NIST 경량암호 공모 최종 후보 10종에 대한 경량 AEAD 최신 동향

이 용 성*, 홍 석 희**

요 약

전자기기의 통신에 있어서 암호 시스템은 안전한 통신을 가능하게 해주는 주요 수단이다. 사물 인터넷과 같은 소형화된 전자기기가 등장함에 따라 기존에 사용하던 AES와 같은 암호 시스템은 소형 디바이스가 작동하는 저전력, 저면적 환경에서 동작하기에 큰 부담을 주게되었다. 이에 따라 다양한 경량 암호들이 제안되어 왔다. 2018년 NIST에서는 이러한 경량 암호의 표준화 작업을 위하여 공모사업을 시작하였고, 2021년 3월에 최종 후보 10종이 발표되었다. 최종 후보로 선택된 10종의 경량 암호의 구조는 향후 새로운 환경에서 사용 가능한 암호 알고리즘을 설계하거나 암호 시스템이 특정 환경에서 소비하는 자원을 가늠하는데 중요한 척도가 될 수 있다. 본 논문에서는 최종후보 10종에 대한 특징을 확인하고자 한다.

Ⅰ. 서 론

사물 인터넷(IOT, Internet of Things)의 발달로 소형화된 전자기기의 안전한 정보통신이 중요한 문제로 대두되고 있다. 이러한 저전력, 저면적을 요구하는 제한된 환경에서 안전한 통신을 위해서는 효과적으로 동작할 수 있는 암호시스템이 필요하다. 현재 널리 사용하고 있는 AES는 안전성은 충분히 검증되었으나, AES를 구현하는데 필요한 하드웨어적인 면적이 크고 암호화를 진행하는 데 소비되는 전력이 많아 제한된 환경에 적합하지 않다. 따라서 저전력, 저면적에서 효율적으로 동작할 수 있는 경량암호의 필요성이 제기되었고, 많은 경량암호가 개발되었다.

2006년 CHES에 발표된 64-비트 블록암호 HIGHT[1]의 제안 논문에는 경량암호의 필요성을 제기하였다. HIGHT는 8-비트 프로세서의 효율적인 구현을 목표로 8-비트 단위의 덧셈(addition), 비트회전(rotation), XOR 연산을 주로 사용하는 ARX 구조로 설계되었다. ARX 구조는 S-box를 사용하지 않고 덧셈 연산으로 비선형성(non-linearity)을 제공하기 때문에, 적은 비용으로 효율적인 암호알고리즘을 설계할 수 있다. ARX 구조를 사용하는 다른 암호로 블록암호 LEA[2], SPECK[3] 등이 있다. SPECK과 함께 발표된 블록암호 SIMON[3]은 ARX와 유사한 구조를 사용하고 있지만, 덧셈 연산 대신 AND 비트연산(&)을 사용하는 특징이 있다.

ARX 구조와 다른 방향으로 일반적인 암호 구조에 사용되는 8-비트 S-box 대신 4-비트 S-box를 사용하는 방향으로 경량암호를 설계하기도 한다. 블록암호 PRESENT[4], GIFT[5], SKINNY[6]가 대표적인 4-비트 S-box를 사용하여 경량암호를 설계한 예이다. 4-비트 S-box는 S-box의 크기가 작아 경량암호에 적합하기도 하지만 S-box를 사용하지 않아도 ANF(Algebraic Normal Form) 식으로 구성하더라도 비교적 간단하게 구성할 수 있다는 장점이 있다. 또한, 4-비트 S-box는 전수조사 할 수 있으므로[7], 설계자가 원하는 성질을 우선하여 S-box를 탐색할 수 있다.

다양한 경량암호가 제안된 와중에 미국 국립표준기술연구소(NIST, The National Institute of Standard and Technology)에서 제한된 환경에서 사용 가능한 표준 경량암호 공모를 진행하였다. 57종의 후보가 접수되었고 최종 후보로 10종의 암호가 선정되었다. 본 논문에서는 최종 후보로 선택된 10종의 암호의 특징을 분석한다.

* 고려대학교 정보보호대학원 (대학원생, yslee0804@korea.ac.kr)
** 고려대학교 정보보호대학원 (교수, shhong@korea.ac.kr)

Ⅱ. NIST 경량암호 표준화 공모

2018년 8월, NIST에서 경량암호 표준화 공모 제출물들에 대한 요구사항과 평가 기준을 공표하였다[8]. 기존에 블록 암호로 제안된 알고리즘과는 달리 AEAD(Authenticated Encryption with Associated Data) 형태로 제출할 것을 요구하였다. AEAD는 암복호화 기능뿐 아니라 무결성을 인증할 수 있는 태그(tag)를 같이 제공하며, 평문 메시지 이외로 넌스(nonce)와 연관데이터(associated data)를 입력으로 받는 암호화 방식이다. 2019년 2월까지 57종의 암호가 제출되었고, 2019년 5월에 56종의 암호가 1라운드 후보로 선정되었다.

2019년 8월에 1라운드 후보 알고리즘 가운데 32종의 알고리즘이 2라운드 후보로 발표되었다. 그와 함께 9월에는 1라운드 후보에 대한 현황 보고서가 게시되었다[9].

2021년 3월, 10종의 최종 후보 알고리즘이 선정되었다. 선정된 알고리즘은 [표 5]에서 확인할 수 있다. 또한, 같은 해 7월에 2라운드 후보에 대한 현황 보고서가 게시되었다[10]. 2라운드 후보에 대한 보고서에는 모든 32종 알고리즘에 대한 평가내용과 함께 다양한 환경에서 성능을 측정한 내용을 담고 있다.

Ⅲ. 최종 진출 후보 특징 분석

NIST 경량 암호 표준화 공모전에 최종 후보로 선정된 10종의 후보 알고리즘은 내부 함수의 기반에 따라 크게 3 분류로 나눌 수 있다. 첫 번째로 경량 블록암호나 경량 트위커블(tweakable) 블록암호를 기반으로 AEAD가 가능한 운영모드를 결합한 분류가 있다. 기존의 개발되어 안전성과 효율성이 검증된 블록암호를 사용 가능하다는 장점과 내부에 사용된 블록암호를 변경하더라도 안전성이 증명된 운영모드는 그대로 사용 가능한 장점이 있다. 두 번째로 순열(permutation) 함수를 기반으로 AEAD를 설계한 분류가 있다. 대부분의 순열 기반 암호들은 duplex 구조나 스폰지(sponge) 구조로 불리는 운영모드를 사용하고 있으며, 일반적으로 별도의 키 스케줄을 요구하지 않는 장점이 있다. 하지만, 순열 함수를 돌리기 위한 내부 상태값의 크기가 큰 경우가 있고, 여러 메시지 블록을 병렬적으로 처리하기 어려울 수 있다. 세 번째로 스트림 암호를 기반으로 AEAD를 설계한 분류가 있다. 스트림 암호의 경우 블록 기반 암호보다 하드웨어 자원을 고려한 경우가 많아 제한된 환경에 더욱 적합할 수 있으나, 설계된 구조의 안전성 증명이 어려운 단점이 존재한다.

〔표 1〕 NIST 경량암호 표준화 공모 최종 후보 알고리즘

	해시함수 미지원	해시함수 지원
블록 암호 또는 트위커블 블록 암호 기반	GIFT-COFB TinyJAMBU	Romulus
순열 기반	Elephant ISAP	ASCON PHOTON -Beetle SPARKLE Xoodyak
스트림 암호 기반	Grain -128AEAD	-

본 장에서는 최종 진출한 10종의 후보 알고리즘을 내부 기반 함수를 기준으로 구별하여 각각의 특징을 확인한다.

3.1. 블록 암호/트위커블 블록 암호 기반

GIFT나 SKINNY를 포함한 경량 블록 암호들을 활용하며 AEAD를 설계하는 방법 중 하나는 MAC(Message Authentication Code)을 포함한 운영체제와 결합하여 AEAD를 설계하는 것이다. 최종 후보로 올라온 GIFT-COFB[11]의 경우 경량블록암호 GIFT-128[5]과 운영모드 COFB(COmbined FeedBack)[11]를 결합한 AEAD이며, 키 길이, 넌스 길이, 태그 길이는 모두 128-비트 한 종류만 지원한다.

GIFT-COFB의 내부 함수로 사용된 GIFT-128은 4-비트 S-box를 사용하고 BOGI(Bad Output must go to Good Input) 설계기법을 적용한 비트 순열을 사용하는 경량암호이다. BOGI 설계기법은 차분 경로를 구성하거나 선형근사식을 구성할 때, 한 라운드에서 높은 확률을 갖는 차분이나 큰 편향성(bias)을 갖는 선형근사식이 다음 라운드에서는 낮은 확률이나 편향성이 작아지도록 비트 순열을 구성하는 설계기법이다. 실제로 GIFT-128의 차분분석은 40라운드 중 26라운드까지밖에 분석이 되지 않았다[13]. 4-비트 S-box와 BOGI 설계기법을 적용한 GIFT-128을 사용함으로써 경량성과 안전성이 검

증된 기반 블록 암호를 확보하였다.

이와 함께 사용된 운영모드 COFB는 출력값이 입력값으로 피드백되는 OFB(Output FeedBack) 모드의 형태를 기본적인 틀로 잡고 있지만, 출력값에서 입력값으로 피드백되기 전에 간단한 피드백 함수를 거치고 연관데이터나 메시지 블록이 입력값에 XOR되는 형식으로 태그값까지 생성할 수 있게 구성되었다. GIFT-128을 사용한 다른 후보 SUNDAE-GIFT[14]의 경우 안전성에 대한 지적사항이 존재하였지만[10]. 피드백 함수의 존재와 태그값 생성방식의 차이점 때문에 GIFT-COFB의 안전성에는 영향을 끼치지 못하였다.

다음으로 Romulus[15]는 경량 트위커블 블록암호 SKINNY-128[6]을 기반으로 설계된 AEAD이다. 4종류의 운영모드가 존재하며, 넌스가 올바르게 사용된 경우(nonce respecting)를 가정한 Romulus-N, 넌스를 중복하여 사용하는 등 잘못 사용한 경우(nonce missuse)를 가정한 Romulus-M, 부채널 정보가 노출되었을 경우 저항성(leakage resilient)이 있는 Romulus-T, 그리고 해시함수로 사용되는 Romulus-H가 존재한다. 초기 제안에는 Romulus-T와 Romulus-H가 존재하지 않았으나 부채널 정보가 노출되었을 경우의 취약점을 보완하기 위하여 Romulus-T가 추가되었고, 해시함수인 Romulus-H를 최종 진출본에 추가하였다.

SKINNY-128은 AES와 유사한 SPN 구조로 내부 구성 요소를 경량화하여 설계하였다. 8-비트 S-box는 8개의 NOR 게이트와 8개의 XOR 게이트로 구현이 가능하며, MixColumn 연산은 바이너리(binary) 행렬로 구성되었다. 또한, 키스케줄을 넌스와 유사하게 사용되는 트윅(tweak)과 키를 함께 사용하는 tweakey framework[16]으로 구성하였다. 평문과 키를 제외한 추가적인 트윅을 입력받을 수 있는 점을 활용하여 Romulus의 운영모드는 연관 데이터나 넌스값이 트윅으로 사용되는 특징이 있다.

TinyJAMBU[17]는 다른 블록 암호 기반 암호들과 달리 기존에 개발된 경량 블록 암호를 사용하지 않았다. 또한, TinyJAMBU는 AEAD 공모전인 CAESAR 경진대회 3라운드에 진출한 JAMBU[31]를 변형한 알고리즘으로, 128-비트의 키 순열(keyed permutation)을 내부 함수로 사용하며 운영모드로 duplex 구조를 사용한다. 128-비트의 키 순열은 NFSR(Nonlinear Feedback Shift Register)로 동작하며, 넌스와 연관데이터를 처리할 때는 640번 NFSR을 동작시키고, 키 초기화, 암복호화 과정에는 NFSR을 1024번 동작한다. Saha 등에 의해 TinyJAMBU의 이전 버전이 차분 분석과 선형 분석이 진행되었다[18]. 이 분석은 이전 버전의 TinyJAMBU의 안전성 마진이 약 12%임을 분석하였고, 넌스와 연관데이터 처리할 때 동작하는 NFSR의 횟수를 384번에서 640번으로 수정하였다.

3.2. 순열 함수 기반

순열(permutation) 기반 AEAD는 duplex 구조나 스폰지(sponge) 구조를 사용하거나 약간의 변형을 거쳐서 사용하는 경우가 대부분이다. 또한 SHA-3[19]와 같은 스폰지 구조로 해시함수의 설계가 가능하므로 해시함수를 지원하는 경우가 많다. 순열 함수 기반의 암호들은 내부 상태(state)의 정보가 노출될 경우 비밀 정보가 노출될 위험이 있어 내부 상태값의 일부값만 활용하여 암호화나 태그값, 해시값을 생성한다. 따라서 순열 함수의 상태값의 크기가 큰 경우가 많다.

ASCON[20]은 CAESAR 경진대회의 경량 AEAD 부분 우승 알고리즘 중 하나이다. 주요 구성요소인 순열의 내부 상태값은 320-비트이다. 라운드 함수는 8-비트 상수 XOR, 5-비트 S-box, 64-비트 선형연산으로 구성되어 있다. ASCON의 가장 큰 특징은 라운드 수가 서로 다른 두 종류의 순열을 쓴다는 점이다. 파라미터에 따라 p^a와 p^b 두 종류의 순열 함수를 사용하는데 AEAD에서는 연관 데이터가 입력되기 전 초기화 과정과 태그값 생성 전에 p^a를 사용하고 연관 데이터가 흡수되거나 암복호화 과정에는 p^b를 사용한다. p^a는 12라운드 순열을 사용하고, p^b는 입력받는 메시지 길이에 따라 6 또는 8라운드 순열을 사용한다. 또한, 초기화 과정과 태그값 생성 전에 추가적으로 키 정보가 내부 상태값에 반영되어 ASCON의 중간 내부 상태값의 정보를 알더라도 비밀키를 찾거나 태그값을 위조하기 어렵게 설계되어있는 특징이 있다.

다음으로 Elephant는 최종 후보로 올라온 순열 기반 AEAD 중 duplex 구조를 사용하지 않은 유일한 암호이다. 주어진 파라미터에 따라 (160, 176, 200)-비트의 내부 상태값 크기를 갖는다. 160-비트나 176-비트 크기의 상태값을 가질 경우, Spongent 순열을 사용하는데, 상수 XOR, 4-비트 S-box, 비트 위치 변환 함수로 한 라운

드가 구성되어 있고 각각 80라운드 또는 90라운드 동작한다. 200-비트의 내부 상태값 크기를 가질 경우엔 200-비트 내부 상태값 크기를 갖는 Keccak 함수를 18라운드 사용한다.

Elephant의 암호화 과정은 암호문 생성과 태그 생성이 분리되어 있고, 암호문 생성 과정은 병렬적으로 진행할 수 있다. 내부에 순열이 사용될 때는, 순열을 기준으로 입출력값에 비밀 정보값이 XOR 되는 키순열을 사용한다. 암호문 생성시에는 넌스를 입력으로 나온 출력값을 메시지와 XOR하여 암호문을 생성한다. 태그값 생성시에는 연관데이터와 암호문을 입력으로 생성된 출력값들을 모두 XOR 하여 합한 뒤 키 순열 함수를 한 번 더 거친 뒤 태그값을 생성한다.

ISAP[21]의 특징은 암호화 과정과 태그 생성과정이 따로 분리된 스폰지(sponge) 형태라는 점이다. 내부 순열은 따로 설계하지 않고 ASCON 순열이나 Keccak 순열을 사용한다. 암호화 과정에서는 넌스로부터 초기화 과정을 거친 뒤 순열 함수를 거치며 키 스트림을 압착(squeezing)하여 암호문을 생성한다. 태그 생성시에는 넌스와 초기값으로 초기화를 진행하고 연관데이터와 암호문을 흡수(absorbing)하여 태그값을 생성한다. 이러한 구조는 ISAP이 부채널 정보누출에 저항성을 가질 수 있게 해준다.

PHOTON-Beetle[22]은 PHOTON 해시 함수[23]에 사용되는 PHOTON256 순열을 기저함수로 사용하며 운영모드로 스폰지 구조 기반인 Beetle[24]을 사용한다. PHOTON256은 AES와 유사한 구조로 설계되어 있는데, 내부는 8×8 형태로 64개의 4-비트 셀(Cell)로 구성되어 있으며 상수 XOR, 4-비트 S-box, ShiftRows, MixColumns로 구성되어 12라운드를 반복한다. 운영모드 Beetle은 일반적인 스폰지구조의 AEAD와 유사하지만 암호문 생성 시 추출한 상태값을 셔플하여 평문과 XOR하여 생성한다.

SPARKLE[25]은 경량 블록 암호 SPARX[26]을 변형한 내부 순열을 사용한다. 내부 상태값의 크기는 256, 384, 512 비트가 있으며, 파라미터에 따라 다른 내부 상태값을 사용한다. SPARX은 64-비트 ARX-box인 Alzette를 사용하며 내부 상태값에 따라 각각 4-브랜치, 6-브랜치, 8-브랜치를 갖는 파이스텔 구조와 유사하게 라운드가 구성되어있다. 각각의 브랜치는 Alzette를 통과한 후 절반의 브랜치가 선형함수를 통과하여 남은 절반의 브랜치에 XOR된 후 스왑(swap)되며 라운드 함수가 동작한다. 운영모드는 SCHWAEMM이라 불리는 스폰지 기반 구조를 사용한다. 이 구조는 ASCON과 유사하게 라운드 수가 다른 두 종류의 순열을 사용하며 연관데이터 입력 전, 암호화 과정 전, 태그 생성 전에 좀 더 많은 라운드 갖는 순열이 존재한다.

Xoodyak[27]은 SHA3 알고리즘으로 선정된 Keccak[19]과 유사한 Xoodoo[23]라는 순열을 사용한다. Xoodoo의 내부 상태값의 크기는 384-비트이고, $4\times32\times3$의 3차원 공간으로 표현된다. 라운드 함수는 2차 비선형 함수인 χ, 선형함수인 θ, ρwest, ρeast, 그리고 상수 덧셈 ι로 한 라운드가 구성된다. 운영모드는 duplex 구조로 Cyclist라 불리는 초기화 함수를 사용하여 Cyclist 모드로 명명하고 있다. Keccak과 유사한 구조 때문에 Keccak 순열에 적용가능한 공격이 Xoodoo에 유사하게 적용 될 수 있다.

3.3. 스트림 암호 기반

Grain-128AEAD[29]는 스트림암호 Grain[30]을 변형하여 AEAD로 개량하였다. 128-비트의 NFSR과 128-비트의 LFSR이 결합되어 있는 형태이다. 128-비트의 키와 96-비트의 넌스로 초기 상태값을 NFSR과 LFSR의 초기 상태값을 설정한 이후 384 클록(clock)동안 공회전을 한다. 이후 출력되는 64-비트 키 스트림은 태그값 생성을 위하여 Accumulator에 저장하고 다음 64-비트는 64-비트 레지스터에 저장을 한다. 이후 키 스트림을 생성하는데 짝수 번째 클록에 생성되는 키 비트는 메시지와 XOR하여 암호화를 진행하고, 홀수 번째 클록에 생성되는 키 비트는 레지스터에 저장하며 메시지 비트가 1일 때 레지스터의 값으로 Accumulator를 업데이트한다. Grain-128AEAD를 제외한 다른 스트림 암호 기반 AEAD는 공모전의 2라운드에 진출하지 못하였다.

Ⅳ. 결 론

2018년부터 시작된 NIST 경량 암호 표준화 공모가 막바지에 접어들었다. 최종 후보가 된 10종의 AEAD 모두 안전성과 효율성이 뛰어난 암호들이며 저마다의 설계전략이 존재한다. 경량 암호의 특성상 환경에 따라

퍼포먼스의 차이가 존재할 수 있으므로 알고리즘 사이의 우열을 가리는 것은 매우 어려운 문제이다. 향후 기술이 발달하여 더욱 다양한 제한된 환경이 추가되어 새로운 암호시스템이 필요한 경우, 최종 진출 10종의 암호가 암호시스템이 차지하는 면적이나 속도를 가늠하는 척도가 되어줄 것으로 기대한다.

참 고 문 헌

[1] Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B.-S, Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., Chee, S. "HIGHT: A New Block Cipher Suitable for Low-Resource Device", CHES 2006. LNCS, vol. 4249, pp. 46-59, 2006.

[2] Hong, D., Lee, J. K., Kim, D. C., Kwon, D., Ryu, K. H., & Lee, D. G., "LEA: A 128-bit block cipher for fast encryption on common processors", International Workshop on Information Security Applications, pp. 3-27, 2013.

[3] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L., "The SIMON and SPECK lightweight block ciphers", Proceedings of the 52nd Annual Design Automation Conference, pp. 1-6, 2015.

[4] Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J., Bobshaw, M. J., Seurin. Y., Vikkelsoe, C., "PRESENT: An Ultra-Lightweight Block Cipher", CHES 2007, pp. 450-466, 2007.

[5] Banik, S., Pandey, S. K., Peyrin, T., Sasaki, Y., Sim, S. M., Todo, Y., " GIFT: a small present", International Conference on Cryptographic Hardware and Embedded Systems, pp. 321-345, 2017.

[6] Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S. M., "The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS", CRYPTO 2016, pp. 123-153, 2016.

[7] Leander, G., Poschmann, A. "On the classification of 4 bit s-boxes", In International Workshop on the Arithmetic of Finite Fields, pp. 159-176, 2007.

[8] NIST, "Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process", "https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf", 2018.

[9] NIST, "Status Report on the First Round of the NIST Lightweight Cryptography Standardization Process", NISTIR 8268, 2019.

[10] NIST, "Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process", NISTIR 8369, 2021.

[11] Banik, S., Chakraborti, A., Iwata, T., Minematsu, K., Nandi, M., Peyrin, T., Sasaki, Y., Sim, S., M., Todo, Y., "GIFT-COFB", "https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/gift-cofb-spec-final.pdf", 2021.

[12] Wen, F., Liu, J., Shan, W. "The COFB Mode of Operation and Its Security Analysis", 2006 International Conference on Computational Intelligence and Security, Vol. 2, pp. 1335-1338, IEEE, 2006.

[13] Li, L., Wu, W., Zheng, Y., Zhang, L. "The Relationship between the Construction and Solution of the MILP Models and Applications" IACR Cryptol. ePrint Arch., 2019/49, 2019.

[14] Banik, S., Bogdanov, A., Peyrin, T., Sasaki, Y., Sim, S., M., Tischhauser, E., Todo, Y., "SUNDAE-GIFT", "https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/SUNDAE-GIFT-spec.pdf", 2021.

[15] Iwata, T., Khairallah, M., Minematsum, K., Peyrin, T., Guo, C., "Romulus", "https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/romulus-spec-final.pdf", 2021.

[16] Jean, J., Nikolić, I., Peyrin, T. "Tweaks and keys for block ciphers: The TWEAKEY framework". In International Conference on the Theory and Application of Cryptology and Information

Security, pp. 274-288, Springer, 2014.

[17] Hongjun Wu, Tao Huang, "TinyJAMBU", "https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/tinyjambu-spec-final.pdf", 2021.

[18] Saha, D., Sasaki, Y., Shi, D., Sibleyras, F., Sun, S., Zhang, Y. "On the security margin of TinyJAMBU with refined differential and linear cryptanalysis". IACR Transactions on Symmetric Cryptology, pp. 152-174, 2020.

[19] NIST, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", FIPS 202, 2015.

[20] Dobraunigm, C., Eichlseder, M., Mendel, F., Schläffer M., "ASCON", "https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/ascon-spec-final.pdf", 2021.

[21] Dobraunig, C., Eichlseder, M., Mangard, S., Mendel, F., Mennink, B., Primas, R., Unterluggauer, T., "ISAP", "https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/isap-spec-final.pdf", 2021.

[22] Bao, Z., Chakraborti, A., Datta, N., Guo, J., Nandi, M., Peyrin, T., Yasuda, K., "PHOTON-Beetle", "https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/photon-beetle-spec-final.pdf", 2021.

[23] Guo, J., Peyrin, T., Poschmann, A. "The PHOTON family of lightweight hash functions" In Annual Cryptology Conference, pp. 222-239. Springer, 2011.

[24] Chakraborti, A., Datta, N., Nandi, M., Yasuda, K. "Beetle family of lightweight and secure authenticated encryption ciphers". IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 218-241, 2018.

[25] Beierle, C., Biryukov, A., Santos, L., C., Großschädl, J., Perrin, L., Udovenko, A., Velichkov, V., Wang, Q., Moradi, A., Shahmirzadi, A., R., "SPARKLE", "https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/sparkle-spec-final.pdf", 2021.

[26] Dinu, D., Perrin, L., Udovenko, A., Velichkov, V., Großschädl, J., Biryukov, A. "Sparx: a family of ARX-based lightweight block ciphers provably secure against linear and differential attacks", In NIST Lightweight Cryptography Workshop 2016. 2016.

[27] Daemen, J., Hoffert, S., Peeters, M., Assche, G., V., Keer, R., V., Mella, S., "Xoodyak", "https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/xoodyak-spec-final.pdf", 2021.

[28] Daemen, J., Hoffert, S., Van Assche, G., Van Keer, R., "The design of Xoodoo and Xoofff", IACR Trans. Symmetric Cryptol. no. 4, pp. 1–38, 2018.

[29] Hell, M., Johansson, T., Meier, W., Sönnerup, J., Yoshida, H., Maximov, A., "Grain-128AEAD", https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/grain-128aead-spec-final.pdf", 2021.

[30] Hell, M., Johansson, T.,Meier, W. "Grain: a stream cipher for constrained environments", International journal of wireless and mobile computing, 2(1), pp. 86-93. 2007.

[31] Wu, H., Huang, T., “JAMBU lightweight authenticated encryption mode and AES-JAMBU”. CAESAR competition proposal. 2014.

〈저 자 소 개〉

이 용 성 (Yongseong Lee)

학생회원

2015년 2월 : 고려대학교 수학과 졸업
2018년 2월 : 고려대학교 정보보호대학원 석사
2018년 3월~현재 : 고려대학교 정보보호대학원 박사과정
<관심분야> 대칭키 암호알고리즘 설계 및 분석

홍 석 희 (Seokhie Hong)

정회원

1995년 : 고려대학교 수학과 학사
1997년 : 고려대학교 수학과 석사
2001년 : 고려대학교 수학과 박사
1999년 8월~2004년 2월 : ㈜시큐리티 테크놀로지 선임연구원
2003년 3월~2004년 2월 : 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월 : K.U. Leuven ESAT/SCD-COSIC 박사후 연구원
2005년 3월~2013년 8월 : 고려대학교 정보보호대학원 부교수
2013년 9월~현재 : 고려대학교 정보보호대학원 정교수
<관심분야> 대칭키 및 공개키 암호 알고리즘, 부채널 공격 및 대응기법, 디지털 포렌식