Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.4, Aug. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.4.617

양자 컴퓨팅 환경에서의 Ascon-Hash에 대한 Free-Start 충돌 공격*

조 세 희,[1†] 백 승 준,[1] 김 종 성[2‡]
[1,2]국민대학교 (대학원생, 교수)

A Quantum Free-Start Collision Attack on the Ascon-Hash*

Sehee Cho,[1†] Seungjun Baek,[1] Jongsung Kim[2‡]
[1,2]Kookmin University (Graduate student, Professor)

요 약

Ascon은 2015년부터 진행되고 있는 NIST 경량암호 공모사업의 최종 라운드 후보 중 하나이며, 해시 모드 Ascon-Hash와 Ascon-Xof를 지원한다. 본 논문에서는 Ascon-Hash의 충돌 공격을 위한 MILP 모델을 개발하고, 해당 모델을 통해 양자 컴퓨팅 환경에서 활용 가능한 차분 경로를 탐색한다. 또한, 탐색한 차분 경로를 이용하여 양자 컴퓨터를 사용할 수 있는 공격자가 3-라운드 Ascon-Hash의 양자 free-start 충돌쌍을 찾을 수 있는 알고리즘을 제시한다. 본 공격은 Ascon-Hash에 대한 충돌 공격을 양자 컴퓨팅 환경에서 최초로 분석했다는 점에서 유의미하다.

ABSTRACT

Ascon is one of the final round candidates of the NIST lightweight cryptography contest, which has been underway since 2015, and supports hash modes Ascon-Hash and Ascon-Xof. In this paper, we develop a MILP model for collision attack on the Ascon-Hash and search for a differential trail that can be used in a quantum setting through the model. In addition, we present an algorithm that allows an attacker who can use a quantum computer to find a quantum free-start collision attack of 3-round Ascon-Hash using the discovered differential trail. This attack is meaningful in that it is the first to analyze a collision attack on Ascon-Hash in a quantum setting.

Keywords: Ascon, Ascon-Hash, Quantum collision, Free-start collision, MILP

I. 서 론

현재 양자 컴퓨팅 환경이 빠른 속도로 성장함과 더불어, 양자 알고리즘의 활용이 현실로 다가오고 있다. 양자 알고리즘의 활용은 암호학계에도 지대한 영향을 미치고 있으며, 공개키 암호의 경우 쇼어 알고리즘, 대칭키 암호와 해시함수의 경우에는 그로버 알고리즘이 암호 알고리즘의 안전성을 위협하고 있다.

1993년 Eli Biham, Adi Shamir에 의해 제안된 차분 분석은 현재까지도 블록 암호, 해시함수 분석에 사용되고 있다. 일반적으로 해시함수 충돌 공격에 차분 분석을 적용할 때에는 생일 공격 경계를 기반으로 차분 경로의 확률을 결정한다. 그러나 양자 컴퓨팅 환경에서는 양자 알고리즘을 이용해 기존 컴퓨팅 환경보다 더 낮은 확률을 가지는 차분 경로를 해시함수 충돌 공격에 사용할 수 있다. 그로버 알고리즘을 이용한 차분 분석기반 해시함수 공격은 2020

Received(05. 20. 2022), Accepted(06. 14. 2022)
* 본 연구는 고려대 암호기술 특화연구센터(UD210027XD)를 통한 방위사업청과 국방과학연구소의 연구비 지원으로 수행되었습니다.
† 주저자, ghfaos7708@kookmin.ac.kr
‡ 교신저자, jskim@kookmin.ac.kr(Corresponding author)

년 후반기부터 꾸준히 수행되고 있다[1,2,3,4,5,6].

NIST는 2015년부터 현재까지 경량암호 공모사업을 진행 중이며, 10개의 최종 후보 알고리즘이 발표된 상태이다. 최종 후보 중 5종이 해시 모드를 지원하고, Ascon[7]은 이에 포함된다. CAESAR 경진대회의 후보 중 하나이기도 한 Ascon의 특징으로는 스펀지 구조를 가지며 라운드 수가 서로 다른 순열 (P_a, P_b)을 사용한다는 점이다. 스펀지 구조의 대표적인 특징은 메시지 Absorbing 과정과 출력값 Squeezing 과정을 수행한다는 점이다. NIST 최종 후보인 ISAP[8], PHOTON-Beetle[9], SPARKLE[10] 알고리즘들도 스펀지 구조를 사용하고 있다. 특히, ISAP의 경우 별도의 순열을 사용하지 않고 Ascon이나 Keccak[11]의 순열을 사용한다. 따라서, Ascon의 해시 모드 Ascon-Hash에 대한 안전성 분석은 Ascon 자체 알고리즘뿐만 아니라 다른 NIST 후보, 스펀지 구조 기반 알고리즘의 안전성에도 영향을 줄 수 있다는 점에서 매우 중요하다.

본 논문의 구성은 다음과 같다. 2장에서는 논문 전개에 필요한 사전 지식을 서술한다. 3장에서는 [12]에서 제안된 2-라운드 Ascon-Hash 충돌 공격과 본 논문에서 제안하는 3-라운드 Ascon-Hash 양자 충돌 공격을 설명한다. 4장에서는 결론 및 향후 연구를 제시한다.

II. 사전 지식

본 장에서는 Ascon-Hash를 간략히 정리하고 양자 컴퓨팅 환경에서 사용되는 기호, 그로버 양자 알고리즘, 여러 가지 양자 컴퓨팅 환경에서의 충돌 공격을 소개한다.

2.1 Ascon-Hash

Ascon-Hash는 256-비트 해시값을 제공하는 해시함수이다. Ascon의 구조와 같이 Initialization, Absorbing, Squeezing 과정을 진행한다. Initialization 과정에서는 $IV\|0^c$값이 입력으로 들어오며, 순열이 적용된다. Ascon-Hash는 P_a, P_b로 나뉘어 서로 다른 라운드를 가지는 Ascon의 순열과 달리 12-라운드로 고정된 순열 $(P_a = P_b)$이 적용된다. Absorbing 과정에서는 64-비트 메시지가 각 블록마다 XOR되며, Squeezing 과정에서는 여러

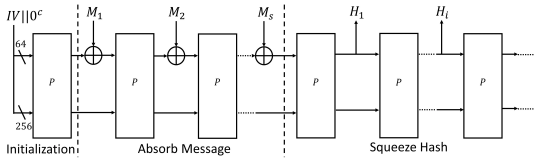

Fig. 1. Hashing mode of Ascon-Hash

319	···	256	x_0
255	···	192	x_1
191	···	128	x_2
127	···	64	x_3
63	···	0	x_4

Fig. 2. State of Ascon

번의 64-비트 값이 출력된 후 그것들을 연접한 256-비트 해시값이 생성된다. 다음 Fig. 1은 Ascon-Hash의 해싱 과정 도식이다.

순열은 상수 덧셈, 비선형 계층, 선형 계층 세 단계로 진행된다. Ascon의 320-비트 state는 메시지가 XOR되는 rate 부분 64-비트와 capacity 부분 256-비트로 나뉜다. 다음 Fig. 2는 5×64 행렬로 표현된 Ascon의 state이다. 각 행을 $x_i (i=0,1,2,3,4)$로 정의한다.

2.1.1 상수 덧셈

상수 덧셈(Addition of Constant)은 각 라운드에 정해진 8-비트 상수를 Ascon state x_2에 XOR하는 과정이다. 각 라운드에 정해진 상수는 다음 Table 1.과 같다.

Table 1. Constant for each round

0	$0x00000000000000f0$	6	$0x0000000000000096$
1	$0x00000000000000e1$	7	$0x0000000000000087$
2	$0x00000000000000d2$	8	$0x0000000000000078$
3	$0x00000000000000c3$	9	$0x0000000000000069$
4	$0x00000000000000b4$	10	$0x000000000000005a$
5	$0x00000000000000a5$	11	$0x000000000000004b$

2.1.2 비선형 계층

비선형 계층(Substitution Layer)인 S는 Sbox를 이용해 각 비트에 비선형 연산을 적용한다. Ascon의 Sbox는 5-비트 값을 입력받아 5-비트 값에 매핑한다. 해당 Sbox는 Ascon state를 기준으로 각 열에 적용된다. 즉, 전체 Ascon state에 64

번의 Sbox 연산 과정이 진행된다. 다음 Table 2.는 Ascon의 5-비트 Sbox이고, Table 3.은 Ascon Sbox의 대수적 정규 형식(Algebraic Normal Form, ANF)이다.

Table 2. Sbox of Ascon ($y = Sbox(x)$)

x	0	1	2	3	4	5	6	7
y	4	11	31	20	26	21	9	2
x	8	9	10	11	12	13	14	15
y	27	5	8	18	29	3	6	28
x	16	17	18	19	20	21	22	23
y	30	19	7	14	0	13	17	24
x	24	25	26	27	28	29	30	31
y	16	12	1	25	22	10	15	23

Tablep 3. The ANF of Ascon Sbox

y_0	$x_4x_1+x_3+x_2x_1+x_2+x_1x_0+x_1$
y_1	$x_4+x_3x_2+x_3x_1+x_3+x_2x_1+x_2+x_1+x_0$
y_2	$x_4x_3+x_4+x_2+x_1+1$
y_3	$x_4x_0+x_4+x_3x_0+x_3+x_2+x_1+x_0$
y_4	$x_4x_1+x_4+x_3+x_1x_0+x_1$

2.1.3 선형 계층

선형 계층(Linear Diffusion Layer)인 L은 Ascon state를 기준으로 각 행에 독립적으로 순환 연산과 XOR 연산을 적용한다. 각 행에 적용되는 연산은 Table 4.의 규칙을 따른다.

Table 4. Linear diffusion layer

x_0	$x_0 \oplus (x_0 \gg 19) \oplus (x_0 \gg 28)$
x_1	$x_1 \oplus (x_1 \gg 61) \oplus (x_1 \gg 39)$
x_2	$x_2 \oplus (x_2 \gg 1) \oplus (x_2 \gg 6)$
x_3	$x_3 \oplus (x_3 \gg 10) \oplus (x_3 \gg 17)$
x_4	$x_4 \oplus (x_4 \gg 7) \oplus (x_4 \gg 41)$

2.2 양자 컴퓨팅

본 절에서는 양자 알고리즘 구성을 위한 양자 기호, 양자 오라클, 그로버 알고리즘, 여러 가지 양자 컴퓨팅 환경에서의 충돌 공격을 소개한다.

양자 컴퓨팅 환경에서는 기존 컴퓨팅 환경의 단위인 비트와는 다르게 큐비트를 사용한다. 큐비트는 양자 게이트를 통해 0과 1의 상태를 동시에 지닐 수 있으며, 이를 중첩상태라 한다. 중첩상태는 관측을 통해 정확한 값이 결정된다.

2.2.1 양자 기호

본 논문에서는 표준 양자 회로 모델을 양자 컴퓨팅 모델로 산정하고 $\{H, CNOT, T\}$의 양자 게이트를 사용한다. 각 양자 게이트는 다음 식 (1), (2), (3)과 같다.

$$H: |b\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b|1\rangle) \quad (1)$$

$$CNOT: |a\rangle|b\rangle \mapsto |a\rangle|b \oplus a\rangle \quad (2)$$

$$T: |0\rangle \mapsto |0\rangle, |1\rangle \mapsto e^{\frac{i\pi}{4}}|1\rangle \quad (3)$$

2.2.2 양자 오라클

부울 함수 $f: \{0,1\}^n \rightarrow \{0,1\}$를 통한 양자 오라클 U_f는 식 (4)와 같이 정의된다. 이때, $x \in \{0,1\}^n$, $y \in \{0,1\}$을 만족한다.

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle \quad (4)$$

목표 요소 x'에 따른 부울 함수 f는 다음과 같다.

$$f(x) = \begin{cases} 1\,(\text{if } x = x') \\ 0\,(otherwise) \end{cases}$$

U_f 입출력은 $(n+1)$개의 큐비트로 이루어져 있으며, n개의 큐비트로 구성된 입력에서 x'에 해당하는 값과 연결된 $|y\rangle$는 1의 값을 가지게 된다. U_f는 역연산할 수 있어야 하며, 부울 함수 f가 역연산이 불가능하더라도 U_f가 역연산이 가능하도록 구현할 수 있다.

2.2.3 그로버 알고리즘

그로버 알고리즘은 1996년 구조화되지 않은 데이터베이스에서 목표 요소를 찾기 위해 제안되었다[13]. 즉, $f(x):\{0,1\}^n \rightarrow \{0,1\}$로 정의된 부울 함수 f에 대해 $f(x')=1$인 입력 x'를 찾는 문제와 같다. 이는 기존 컴퓨팅 환경에서는 2^n의 black-box 오라클의 접근이 필요하지만, 양자 컴퓨팅 환경에서는 그로버 알고리즘을 이용해 $2^{n/2}$번의 오라클 U_f 질의만으로 x'를 찾을 수 있다. 다음은 그로버 알고리즘의 동작 과정이다.

1) n개의 큐비트를 $|0\rangle$상태로 초기화하여 레지스터에 저장한다.

$$|\Psi\rangle = |0^n\rangle$$

2) 레지스터의 각 큐비트에 H를 이용하여 균일한 중첩상태로 변환한다.

$$|\Psi\rangle = H^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle$$

3) 아래 과정을 $R \approx \frac{\pi}{4}\sqrt{\frac{N}{M}} - \frac{1}{2}$번의 반복을 통해 목표 요소의 위상을 증폭한다.($N(=2^n)$은 검색 공간이며, M은 목표 요소 공간이다.)

3.a. 부울 함수 f를 통해 목표 요소의 위상을 반전시킨다.

$$U_{x'}|x\rangle = (-1)^{f(x)}|x\rangle$$

3.b. 반전된 위상들의 진폭을 증폭한다.

$$U_s = 2|\Psi\rangle\langle\Psi| - I$$

4) 관측을 통해 유효한 요소인지 확인한다.

반복 횟수 R은 $M=1$인 경우 약 $\sqrt{N}$번 반복하게 된다. 공격자가 U_f를 구성했을 때, 그로버 알고리즘을 이용한 공격의 복잡도는 약 $\frac{\pi}{4}\times\sqrt{N}\times T_{U_f}$이다. 이때, T_{U_f}는 U_f의 복잡도이다.

2.2.4 양자 컴퓨팅 환경

기존 컴퓨팅 환경에서 충돌 공격의 일반적인 공격은 생일 공격으로, n-비트 해시함수의 공격에 필요한 복잡도는 $2^{n/2}$이다. 하지만, 양자 컴퓨팅 환경에서 필요한 복잡도는 현재 가정되고 있는 양자 컴퓨팅 환경의 양자 알고리즘에 의해 정해진다. 본 논문에서는 충돌 공격에 사용되는 양자 알고리즘과 그 환경을 기준으로 양자 컴퓨팅 환경을 정의한다.

BHT 환경은 작은 규모의 양자 컴퓨터와 큰 규모의 qRAM(양자 컴퓨팅 환경에서의 RAM)을 사용할 수 있는 양자 컴퓨팅 환경에서 BHT 알고리즘[14]을 일반적인 충돌 공격 알고리즘으로 사용하는 환경이다. 이 환경에서는 n-비트 해시함수에 대해 $2^{n/3}$의 qRAM이 사용 가능할 때, $2^{n/3}$의 계산 복잡도가 필요하다. 차분 경로의 확률을 p라 할 때, 그로버 알고리즘이 적용된 식 (5)의 계산을 통해 $2^{-2n/3}$보다 높은 확률을 가지는 차분 경로를 분석에 이용할 수 있다. 256-비트 해시함수 Ascon-Hash의 경우 $2^{-170.7}$보다 높은 확률을 가지는 차분 경로를 분석에 이용할 수 있다.

$$\sqrt{p^{-1}} < 2^{n/3} \Rightarrow p^{1/2} > 2^{-n/3} \Rightarrow p > 2^{-2n/3} \quad (5)$$

CNS 환경은 큰 규모의 양자 컴퓨터와 큰 규모의 cRAM(기존 컴퓨팅 환경의 RAM)을 가지는 환경이고, 일반적인 충돌 공격 알고리즘으로 CNS 알고리즘[15]을 사용하는 환경이다. 이 환경에서는 n-비트 해시함수에 대해 $2^{n/5}$의 cRAM이 사용 가능할 때, $2^{2n/5}$의 복잡도가 필요하다. 식(6) 과정을 통해 $2^{-4n/5}$보다 높은 확률을 가지는 차분 경로를 분석에 이용할 수 있다. 따라서, Ascon-Hash의 경우 이 환경에서 $2^{-204.8}$보다 높은 확률을 가지는 차분 경로를 분석에 이용할 수 있다.

$$\sqrt{p^{-1}} < 2^{2n/5} \Rightarrow p^{1/2} > 2^{-2n/5} \Rightarrow p > 2^{-4n/5} \quad (6)$$

기존 컴퓨팅 환경과 각 양자 컴퓨팅 환경에서 Ascon-Hash에 활용 가능한 차분 경로 확률 p의

경계를 정리하면 다음과 같다.

- 기존 환경 : $p > 2^{-128}$ (Birthday attack)
- BHT 환경 : $p > 2^{-170.7}$
- CNS 환경 : $p > 2^{-204.8}$

III. 3-라운드 Ascon-Hash 양자 충돌 공격

본 장에서는 [12]에서 제안된 2-라운드 Ascon-Hash 공격을 간략하게 소개하고, 이를 확장하여 3-라운드 Ascon-Hash 양자 free-start 충돌 공격을 제안한다.

3.1 2-라운드 Ascon-Hash 충돌 공격

Zong 등은 2-라운드 Ascon-Hash에 대한 충돌 공격[12]을 제안했다. Ascon-Hash는 Absorbing 과정에서 메시지가 rate 부분에만 XOR되는 특징이 있다. 따라서, 차분 경로의 입력 차분이 capacity 부분에 존재한다면, 입력 차분을 만족하는 메시지 쌍을 구성하지 못한다. 즉, 차분 경로의 입력 차분은 rate 부분 x_0를 제외한 capacity 부분 $x_i (i=1,2,3,4)$에는 차분 경로의 차분을 구성할 수 없다.

[12]에서는 차분 경로의 출력 차분을 rate 부분에만 존재하도록 구성하여 다음 블록에서 XOR되는 메시지로 해당 차분을 상쇄시키는 전략을 사용했다. 결과적으로, 해시값을 생성하는 Squeezing 과정에서 충돌이 발생한다. 다음 Fig. 3은 해당 공격 전략을 도식화한 것이다.

2-라운드 차분 경로는 입력 차분이 rate 부분에만 집중되는 특징으로 인해 발생하는 Sbox 성질을 이용하여 첫 라운드의 S에서 확률이 발생하지 않도록 구성됐다. 다음 Table 5.은 $\Delta x_i = 0 (i=1,2,3,4)$을 만족하는 입력이 Sbox에 적용될 때 발생하는 성질들을 나타낸다. 출력 차분 $\Delta y_i (i=0,1,2,3,4)$가 결정

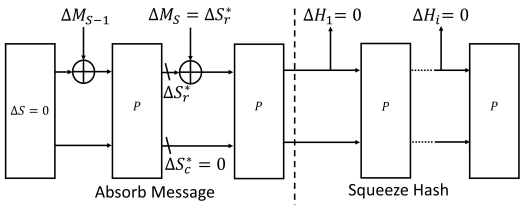

Fig. 3. Collision attack strategy on Ascon-Hash

된 경우 Table 5.의 성질을 만족하도록 Sbox 입력 x_1, x_3, x_4의 값을 출력 Δy_0와 Δy_3의 값과 연결해 추가적인 확률 발생 없이 원하는 입력 차분으로 연결되도록 한다. 해당 성질은 active Sbox에 적용되며, 각 active Sbox마다 $x_1, x_3 \oplus x_4$에 대한 2-비트 조건을 만족하는 값을 찾기 위해 2^2의 자유도가 필요하다. [12]에서 구성한 2-라운드 차분 경로의 확률은 2^{-103}이며, 입력 차분의 active Sbox의 개수는 43개이다. 따라서, 2-라운드 차분 경로의 입력 차분을 만족하는 메시지 쌍을 구성하기 위해서는 총 2^{86}의 자유도가 필요하다. 공격은 4-블록으로 구성됐으며, 첫 두 블록은 2^{86}의 자유도를 충족하는 데 필요하다. 결과적으로, 2^{125}의 시간 복잡도로 충돌 공격을 수행했다.

Table 5. Properties of Ascon Sbox

▪ $\Delta y_0 \oplus \Delta y_4 = 1$
▪ $\Delta y_1 = \Delta x_0$
▪ $\Delta y_2 = 0$
▪ $x_1 = \Delta y_0 \oplus 1$
▪ $x_3 \oplus x_4 = \Delta y_3 \oplus 1$

3.2 확장된 3-라운드 Ascon-Hash 양자 충돌 공격

본 절에서는 [12]에서 제안한 2-라운드 Ascon-Hash 충돌 공격을 확장하여 3-라운드 Ascon-Hash 양자 충돌 공격을 제안한다.

3.2.1 MILP 모델 개발

MILP (Mixed Integer Linear Programming)는 [16]에서 블록 암호 분석 도구로 사용되었고, 현재까지 암호 분석 도구로 활발하게 쓰이고 있다. 이에 본 절에서는 3-라운드 Ascon-Hash 양자 충돌 공격에 활용되는 차분 경로를 탐색하는데 필요한 MILP 모델을 개발한다.

Ascon-Hash의 순열에서 차분 경로에 영향을 주는 단계는 S와 L이므로, 해당 단계들을 MILP로 모델링한다. 선형 제약식의 개수는 MILP 복잡도에 직접적인 영향을 주기 때문에, 복잡도를 낮추기 위해 선형 제약식의 개수를 최소화해야 한다. 우선 S는

Sbox의 DDT를 계산하고, 이를 Sage 프로그램을 통해 약 37,000개의 선형 제약식으로 이루어진 Convex Hull 형태로 나타냈다. 이후, 해당 Convex Hull을 그리디 알고리즘을 이용하여 40개의 선형 제약식으로 최적화했다. Ascon-Hash의 L은 3개의 비트가 XOR되어 1개의 비트로 매핑되며, 이 과정을 MILP에 적용하기 위해 8개의 선형 제약식으로 모델링 했다. S와 L에 대한 선형 제약식들은 부록 A에 제시되어 있다.

효과적인 차분 경로 탐색을 위해 [12]에서 구성한 MILP 모델에 사용된 성질 이외에 추가적인 Sbox 성질을 구성할 수 있다. 해당 성질은 Inverse Sbox의 ANF를 통해 계산 가능하며, 다음 Table 6.은 Inverse Sbox의 ANF이다.

Fig. 3의 공격 전략과 같이, 차분 경로의 출력 차분을 $\Delta x_i = 0 (i = 1,2,3,4)$를 만족하도록 구성한다면, 다음과 같은 계산 과정을 통해 추가적인 성질을 유도할 수 있다.

$$\Delta y_1 = \Delta x_0 (1 \oplus x_2 \oplus x_2 x_4),\ \Delta y_3 = \Delta x_0 (x_2 \oplus x_2 x_4)$$
$$\Rightarrow \Delta y_1 \oplus \Delta y_3 = \Delta x_0$$

위 계산 과정을 통해 active Sbox에 적용되는 Sbox 성질은 $\Delta y_1 \oplus \Delta y_3 = 1$와 같으며, 입력 차분과의 연관을 위해 $\Delta y_1 \oplus \Delta y_3 \oplus \Delta x_0 = 0$으로 수정하여 모델에 적용했다. 이 식을 적용하여 식 개수, 검색 공간을 효율적으로 줄일 수 있다.

Table 6. The ANF of Ascon inverse Sbox

y_0	$x_1 + x_2 + x_3 + x_0x_1 + x_2x_3 + x_0x_2x_3 + x_0x_3x_4 + x_1x_3x_4 + x_2x_3x_4 + 1$
y_1	$x_0 + x_1 + x_4 + x_0x_2 + x_2x_3 + x_0x_2x_4$
y_2	$x_0 + x_1 + x_2 + x_4 + x_0x_2 + x_1x_2 + x_1x_3 + x_2x_3 + x_2x_4 + x_3x_4 + x_0x_1x_2 + x_0x_1x_3 + x_0x_2x_4 + x_1x_2x_4 + x_1x_3x_4 + 1$
y_3	$x_1 + x_3 + x_4 + x_0x_2 + x_1x_2 + x_1x_4 + x_2x_4 + x_0x_2x_4 + x_1x_2x_4$
y_4	$x_3 + x_0x_3 + x_0x_2 + x_1x_2 + x_2x_3 + x_2x_4 + x_0x_2x_3 + x_0x_2x_4 + x_1x_2x_4 + x_2x_3x_4$

3.2.2 3-라운드 차분 경로 구성

[12]에서 제안한 2-라운드 차분 경로는 기존 컴퓨팅 환경에 맞춰 2^{-128}보다 높은 확률로 구성했다. 하지만, 양자 컴퓨팅 환경에서는 더 낮은 차분 경로의 확률을 이용할 수 있다. 즉, 해당 차분 경로의 앞에 BHT 환경에서는 $2^{-67.7}$보다 높은 확률을 갖는 경로를, CNS 환경에서는 $2^{-101.8}$보다 높은 확률을 갖는 경로를 기존 2^{-103}의 경로에 연결한다면 3-라운드로 확장된 공격을 수행할 수 있다.

기존의 공격 전략은 x_0에만 차분이 존재하도록 차분 경로가 구성됐다. 그러나 free-start 공격가정에서는 해시함수의 IV에 차분이 존재할 수 있다. 이 점을 활용해 기존 2-라운드 차분 경로의 앞에 추가되는 라운드는 x_0에만 차분이 존재해야 하는 제약을 완화했다.

다음 Fig. 4는 3-라운드 Ascon-Hash 양자 충돌 공격에 사용되는 3-라운드 차분 경로이다. Sbox의 입력 state를 $X_i (i = 1,2,3)$, 출력 state를 $Y_i (i = 1,2,3)$, 그리고 차분 경로의 마지막 state를 $O (= P(Y_3))$로 정의한다. 차분 경로들의 각 라운드 확률과 차분값은 Table 7.와 같다. 해당 차분 경로는 3.2.1에서 구성한 MILP 모델을 통해 얻었으며, 확률이 $2^{-67.7}$보다 높고 최적의 확률을 가지도록 탐색 됐다. 결과적으로, 두 라운드 앞에 연결될 수 있는 2^{-62}의 확률을 가지는 한 라운드 차분 경로를 기존 경로에 연결할 수 있게 된다. 3-라운드 차분 경

Table 7. The free-start 3-round differential trail

Round (r)	Bias (2^{-a})	Input difference ($\triangle X_r$)	Output difference ($\triangle Y_r$)
1	62	0000000000000000 4b0f0adeaad807c9 4b0f0adeaad807c9 0000000000000000 0000000000000000	4b0f0adeaad807c9 0000000000000000 0000000000000000 0000000000000000 0000000000000000
2	0	e6765f2bfb737f78 0000000000000000 0000000000000000 0000000000000000 0000000000000000	00144000c0404000 e6765f2bfb737f78 0000000000000000 0400000008101000 e6621f2b3b333f78
3	103	0c10400249045804 8232408ad1246801 0000000000000000 0c0102000812100c 8233428ad1366809	8e33428ad936780d 0000000000000000 0000000000000000 0000000000000000 0000000000000000

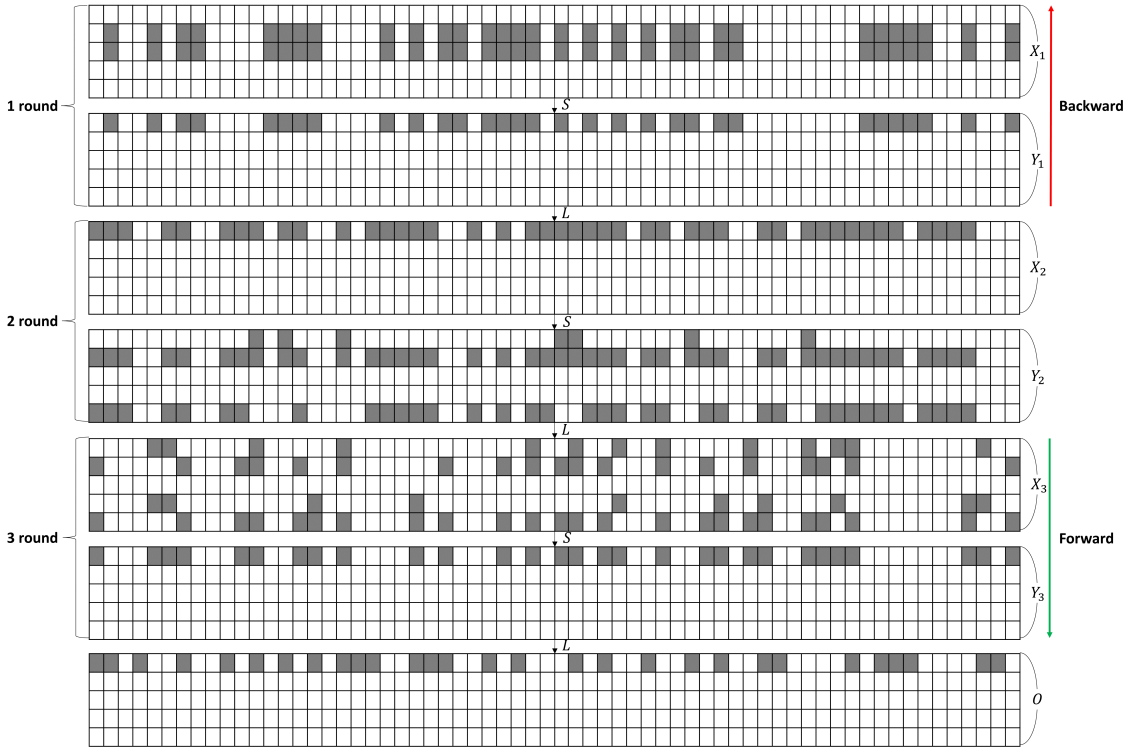

Fig. 4. 3-round differential trail on Ascon-Hash

로의 확률은 2^{-165}이며, 이는 BHT와 CNS 환경 모두에서 충돌 공격에 적용할 수 있다. 차분 경로의 2라운드에서 발생하는 확률은 Table 5. Sbox 성질을 이용해 발생하지 않도록 했다.

3.2.3 공격 과정

공격 과정의 설명을 위해 2.1절에서 정의된 Sbox의 입출력 비트 $x_i, y_i (i=0,1,2,3,4)$를 $x_i^r, y_i^r (r$:라운드, i:비트 위치)로 정의한다. 새로운 공격 전략에서는 Fig. 5와 같이 서로 다른 메시지 쌍 (M_1, M_1^*)에 대해 서로 다른 (IV, IV^*)가 사용된다. 먼저, Initialization 단계의 P에 차분 경로를 적용하고 차분 경로의 출력 차분(ΔO)과 같은 차분을 가지는 메시지 쌍($M_1 \oplus M_1^* = \Delta O$)을 Absorbing 과정에서 XOR한다. 이 과정을 통해 Squeezing 과정에서는 충돌($H_i \oplus H_i^* = 0 (i=1,2,3,4)$)이 발생한다. 자세한 공격 설명을 위해, 부울 함수 f를 다음과 같이 정의한다. F_2^{31}은 중첩상태의 Y_1 active 비트를 뜻하고,

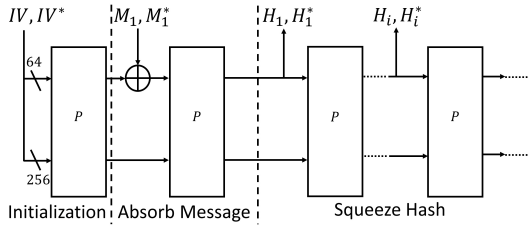

Fig. 5. The free-start collision attack strategy on Ascon-Hash

F_2^{66}은 중첩상태의 X_3 active 비트를 뜻한다. F_2^{68}은 계산 과정에서 필요한 2^{165}의 자유도를 만족시키기 위한 추가적인 비트이다.

$$f: F_2^{31} \times F_2^{66} \times F_2^{68} \rightarrow F_2$$

해당 부울 함수는 2라운드의 입력쌍 $(X_2, X_2^* (X_2 \oplus X_2^* = \Delta X_2))$이 다음과 같은 조건들을 만족하면 $f(X_2, X_2^*) = 1$이다.

1) $x_1^2 = \triangle y_0^2 \oplus 1$, $x_3^2 \oplus x_4^2 = \triangle y_3^2 \oplus 1$을 만족하는 Sbox 입력쌍 (X_2, X_2^*)을 통해 (Y_1, Y_1^*)을 계산한다. 계산된 (Y_1, Y_1^*)에 대해 $\Delta X_1 = S^{-1}(Y_1) \oplus S^{-1}(Y_1^*)$를 만족한다. 즉, Fig. 4의 Backward 방향을 만족한다.
2) 1)을 만족한 (X_2, X_2^*)을 통해 (X_3, X_3^*)을 계산한다. 계산된 (X_3, X_3^*)에 대해 $\Delta Y_3 = S(X_3) \oplus S(X_3^*)$를 만족한다. 즉, Fig. 4의 Forward 방향을 만족한다.

만약 $f(X_2, X_2^*) = 1$을 만족하는 (X_2, X_2^*)를 찾는다면, 충돌쌍 (M, M^*)를 찾을 수 있다. 1)의 과정에서 $x_1^2 = \triangle y_0^2 \oplus 1$, $x_3^2 \oplus x_4^2 = \triangle y_3^2 \oplus 1$의 조건을 만족시키는 쌍을 찾기 위해 2^{86}의 자유도가 필요하다. 각 블록에서 메시지 Absorbing을 통해 2^{64}의 자유도를 얻을 수 있는 기존과는 달리, free-start 가정에서는 rate 부분은 물론, capacity 부분에서도 실제값을 조절할 수 있다. 즉, 메시지의 Absorbing과 상관없이 2^{320}의 자유도를 확보할 수 있다. (X_2, X_2^*)에 따른 부울 함수 $f(X_2, X_2^*)$는 기존 컴퓨팅 환경에서 다음과 같이 계산된다.

1) $x_1^2 = \triangle y_0^2 \oplus 1$, $x_3^2 \oplus x_4^2 = \triangle y_3^2 \oplus 1$을 만족하는 (X_2, X_2^*)에 대해 $L^{-1}(X_2), L^{-1}(X_2^*)$의 역연산을 통해 Y_1, Y_1^*를 계산한다.
2) $S^{-1}(Y_1), S^{-1}(Y_1^*)$의 역연산을 통해 X_1, X_1^*을 계산하고, $X_1^* = X_1 \oplus \Delta X_1$을 만족하는지 확인한다.
3) 2)의 과정을 만족한 (X_2, X_2^*)에 대해 $L(S(X_2))$, $L(S(X_2^*))$의 연산을 통해 (X_3, X_3^*)를 계산한다.
4) (X_3, X_3^*)에 대해 $S(X_3), S(X_3^*)$의 연산을 통해 Y_3, Y_3^*를 계산하고, $Y_3^* = Y_3 \oplus \Delta Y_3$를 만족하는지 확인한다.
5) 4)의 과정까지 모두 만족하는 (X_2, X_2^*)에 대해, $f(X_2, X_2^*)$는 1을 반환하고 만족하는 (X_2, X_2^*)가 없으면 0을 반환한다.

$f(X_2, X_2^*) = 1$을 만족하는 (X_2, X_2^*)에 대해 X_1, X_1^*을 충돌쌍 $M, M^* (M^* = M \oplus \Delta O)$의 IV, IV^*로 설정한다.

3.2.4 양자 오라클 U_f 구현

양자 오라클 U_f는 3.2.3에서 언급된 부울 함수 f의 공격 과정을 양자 컴퓨팅 환경에서 구현한 것이다. U_f는 $U_f : |X_2, X_2^*\rangle |q\rangle \mapsto |X_2, X_2^*\rangle |q \oplus f(X_2, X_2^*)\rangle$로 정의되며, 그 과정은 Fig. 6과 같다.

Input: $|X_2, X_2^*\rangle |q\rangle$
Output: $|X_2, X_2^*\rangle |q \oplus f(X_2, X_2^*)\rangle$
1. /* Backward */
2. Compute the $Y_1 = L^{-1}(X_2)$, $Y_1^* = L^{-1}(X_2^*)$ from X_2, X_2^*, where X_2, X_2^* satisfies the conditions of $x_1^2 = \triangle y_0^2 \oplus 1$ and $x_3^2 \oplus x_4^2 = \triangle y_3^2 \oplus 1$.
3. Compute the $X_1 = S^{-1}(Y_1)$, $X_1^* = S^{-1}(Y_1^*)$.
4. **if** (X_1, X_1^*) fulfills the $X_1 \oplus X_1^* = \Delta X_1$ **then** set 1-bit flag $flag_1 = 1$; otherwise set $flag_1 = 0$.
5. /* Forward */
6. Compute the $X_3 = L(S(X_2))$, $X_3^* = L(S(X_2^*))$ from X_2, X_2^*, where X_2, X_2^* same as 2.
7. Compute the $Y_3 = S(X_3)$, $Y_3^* = S(X_3^*)$.
8. **if** (Y_3, Y_3^*) fulfills the $Y_3 \oplus Y_3^* = \Delta Y_3$ **then** set 1-bit flag $flag_2 = 1$; otherwise set $flag_2 = 0$.
9. **if** $flag_1 = 1$ and $flag_2 = 1$ **then**
   return $|X_2, X_2^*\rangle |q \oplus 1\rangle$
   else
   return $|X_2, X_2^*\rangle |q\rangle$

Fig. 6. Implementation of U_f

3.2.5 복잡도 분석

최종적으로 충돌을 찾는데 필요한 복잡도 분석은 다음과 같은 조건들을 고려했다.

- 3-라운드 Ascon-Hash의 연산은 총 $64 \times 3 = 192$번의 Sbox 연산이다.
- 한 번의 S^{-1} 연산은 두 번의 S의 연산이다.[17]

- 계산 과정의 역연산은 U_f의 작동 이후 양자 회로를 푸는 것으로 가능하다.

[17]에서는 양자 환경에서의 AES Sbox에 대한 분석을 제시한다. Ascon Sbox의 경우 AES Sbox보다 적은 게이트로 이루어지며, 양자 회로에서도 적은 양자 게이트로 구성될 것으로 예상한다. 따라서, [17]의 결과를 참고하여 이후의 과정에서도 한 번의 S^{-1} 연산을 두 번의 S로 설정하며, 이 설정이 전체 공격 복잡도에 주는 효과는 미미하다.

양자 오라클 U_f를 작동하기 위한 Sbox 연산은 Algorithm 1.의 3번 과정에서 Y_1에 대해 Sbox 역연산이 적용되므로, 총 $2\times2\times64=256$번의 Sbox 연산이 필요하다. 6, 7번 과정에서는 각각 64번의 Sbox 연산이 발생해 총 $2\times(64+64)=256$번의 Sbox 연산이 필요하다. 즉, T_{U_f}는 $512/192\approx2^{1.4}$번의 3-라운드 Ascon-Hash 연산으로 고려한다.

$f(X_2,X_2^*)=1$을 만족하는 (X_2,X_2^*)를 찾기 위해 그로버 알고리즘을 이용하여 총 $\frac{\pi}{4}\times\sqrt{2^{165}}$번의 U_f 쿼리가 필요하다. 따라서, 충돌을 찾기 위한 총 복잡도는 $\frac{\pi}{4}\times\sqrt{2^{165}}\times2^{1.4}\approx2^{83.55}$이다. Table 8.은 Ascon 해시 모드에 대한 충돌 공격을 정리한 것이다.

본 논문에서 제시한 차분 경로를 이용해 free-start 가정이 아닌 일반 충돌 공격에 적용할 경우, Absorbing 과정에서 하나의 순열을 추가하여 $\triangle X_1$을 만족하는 메시지 쌍을 찾아야 하며, 추가적인 복잡도가 소요된다. 해당 복잡도는 기존 컴퓨팅 환경의 충돌 공격 복잡도 경계인 2^{128}은 물론, 각 양자 컴퓨팅 환경의 충돌 공격 복잡도 경계 $2^{170.7}$, $2^{204.8}$ 또한 상회한다.

Table 8. Summary of collision attacks on Ascon-Hash and Ascon-Xof

Target	Setting	Size	Rounds	Time	Method	Ref.
Ascon-Xof	Classic	64	2/12	2^{15}	Differential	[12]
Ascon-Hash	Classic	256	2/12	2^{125}	Differential	[12]
Ascon-Hash	Quantum	256	3/12	$2^{83.55}$	Differential	This paper

IV. 결론 및 향후 연구

본 논문에서는 NIST 경량암호 공모사업 최종 후보 Ascon-Hash를 양자 컴퓨팅 환경에서 분석했다. 분석을 위해 Ascon-Hash의 순열에 대한 MILP 모델을 개발했으며, Ascon Sbox 연산에 필요한 선형 제약식은 40개, L 연산에서 필요한 제약식은 8개로 구성했다. 총 3-라운드를 공격했으며, MILP 모델 구성을 통해 2^{-165}의 확률을 가지는 차분 경로를 구성했다. 해당 차분 경로는 기존 컴퓨팅 환경에서는 생일 공격의 복잡도를 넘어 충돌 공격에 활용할 수 없지만, 양자 컴퓨팅 환경에서는 그로버 알고리즘을 이용해 충돌 공격에 활용할 수 있다. 결과적으로, BHT와 CNS 양자 컴퓨팅 환경에서 적용 가능한 3-라운드 Ascon-Hash 양자 free-start 충돌 공격을 구성했다. 본 논문에서 제안하는 Ascon-Hash에 대한 공격은 최초의 양자 충돌 공격이라는데 의미가 있다.

본 논문의 결과는 Ascon의 설계자들이 주장하는 안전성 수준에 영향을 주지는 않으며, 차분 분석기반 충돌 공격과 관련하여 Ascon-Hash를 더 잘 이해할 수 있도록 도움을 준다. 스펀지 구조는 NIST 경량암호 공모사업의 최종 후보 ISAP, PHOTON-Beetle, SPARKLE에도 사용되므로, 제안하는 공격 전략이 이들 분석에 영향을 줄 수 있다. 특히, ISAP의 경우 Ascon의 순열을 그대로 활용하므로, 본 논문에서 제안한 MILP 모델은 ISAP의 안전성에 직접적인 영향을 줄 수 있다.

References

[1] A. Kumar Chauhan, A. Kumar and S. Kumar Sanadhya, "Quantum Free-Start Collision Attacks on Double Block Length Hashing with Round-Reduced AES-256", IACR Trans. Symmetric Cryptol, vol. 2021(1), pp. 316-336, 2021.

[2] X. Dong, S. Sun, D. Shi, F. Gao, X. Wang and L. Hu, "Quantum Collision Attacks on AES-like Hashing with Low Quantum Random Access Memories", ASIACRYPT'20, LNCS 12492, pp. 727-757, 2020.

[3] A. Hosoyamada and Y. Sasaki,

"Quantum Collision Attacks on Reduced SHA-256 and SHA-512", CRYPTO'21, LNCS 12825, pp. 616-646, 2021.

[4] S. Baek, S. Cho and J. Kim. "Quantum cryptanalysis of the full AES-256-based Davies-Meyer, Hirose and MJH hash functions" Quantum Information Processing, vol. 21(5), pp. 1-32, 2022.

[5] B. Ni, X. Dong, K. Jia and Q. You, "(Quantum) collision attacks on reduced simpira v2" IACR Transactions on Symmetric Cryptology, vol. 2021(2), pp. 222-248, 2021.

[6] A. Flórez-Gutiérrez, G. Leurent, M. Naya-Plasencia, L. Perrin, A. Schrottenloher and F. Sibleyras, "New Results on Gimli: Full-Permutation Distinguishers and Improved Collisions", ASIACRYPT'20, LNSC 12491, pp. 33-63, 2020.

[7] C. Dobraunig, M. Eichlseder, F. Mendel and M. Schläffer, "Ascon v1.2", "https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/ascon-spec-final.pdf", 2021.

[8] C. Dobraunig, M. Eichlseder, S. Mangard, F. Mendel, B. Mennink, R. Primas and T. Unterluggauer, "Isap v2.0", "https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/isap-spec-final.pdf", 2021.

[9] Z. Bao, A. Chakraborti, N. Datta, J. Guo, M. Nandi, T. Peyrin and K. Yasuda, " PHOTON-beetle authenticated encryption and hash family", "https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/photon-beetle-spec-final.pdf", 2019.

[10] C. Beierle, A. Biryukov, L.C. dos Santos, J. Großschädl, L. Perrin, A. Udovenko, V. Velichkov, and Q. Wang, "Schwaemm and Esch: lightweight authenticated encryption and hashing using the Sparkle permutation family", "https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/sparkle-spec-final.pdf", 2019.

[11] G. Bertoni, J. Daemen, M. Peeters and G. V. Assche, "Keccak", EUROCRYPT'13, LNCS 7881, pp. 313-314, 2013.

[12] R. Zong, X. Dong and X. Wang, "Collision attacks on round-reduced Gimli-Hash/Ascon-Xof/Ascon-Hash", Cryptology ePrint Archive, 2019.

[13] L. K. Grover, "A fast quantum mechanical algorithm for database search", Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pp. 212-219, 1996.

[14] G. Brassard, P. Høyer and A. Tapp, "Quantum Cryptanalysis of Hash and Claw-Free Functions", LATIN'98, LNCS 1380, pp. 163-169, 1998.

[15] A. Chailloux, M. Naya-Plasencia and A. Schrottenloher, "An Efficient Quantum Collision Search Algorithm and Implications on Symmetric Cryptography", ASIACRYPT'17, LNCS 10625, pp.211-240, 2017.

[16] S. Sun, L. Hu, L. Song, Y. Xie and P. Wang, "Automatic security evaluation of block ciphers with S-bP structures against related-key differential attacks", INSCRYPT'13, LNCS 8567, pp. 39-51, 2013.

[17] S. Jaques, M. Naehrig, M. Roetteler and F. Virdia, "Implementing Grover oracles for quantum key search on AES and LowMC", EUROCRYPT'20, LNCS 12106, pp. 280-310, 2020.

부 록

A. MILP 모델의 선형 제약식

다음 Table 9.의 선형 제약식은 Sbox 입력 비트 5개 $\alpha[0],\alpha[1],\alpha[2],\alpha[3],\alpha[4]$, 출력 비트 5개 $\beta[0],\beta[1],\beta[2],\beta[3],\beta[4]$, DDT의 확률 정보를 담은 $\gamma[0],\gamma[1],\gamma[2]$ 그리고 더미 비트 $\delta[0]$으로 이루어진 식 $\alpha[0]+\alpha[1]+\alpha[2]+\alpha[3]+\alpha[4]+\beta[0]+\beta[1]+\beta[2]+\beta[3]+\beta[4]+\gamma[0]+\gamma[1]+\gamma[2]+\delta[0] \geq 0$을 만족한다.

Table 9. Linear inequality of Ascon Sbox

(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, -1, -1, 1, 0),
(-8, 3, -29, -6, -17, 3, -3, -2, -2, -10, 57, 69, 36, 0, 0),
(7, -9, 16, -2, -1, 0, 0, 3, 3, 1, 10, 3, -7, 0, 0),
(2, 5, 2, 10, 8, -1, 1, 5, 0, -1, -10, -8, -2, 0, 0),
(-21, 0, 9, -9, 8, -3, 2, -4, -4, -2, 41, 22, 26, 0, 0),
(0, 5, 0, 2, -2, 1, 0, 0, 0, 1, -3, 1, 0, 0, 0),
(3, -2, -4, 3, 6, 0, 0, -1, -1, 1, 2, 4, 6, 0, 0),
(2, 6, 2, -4, 5, 4, 0, 0, 1, 4, -2, -4, -3, 0, 0),
(19, -13, 7, 5, -21, -1, 7, -1, -2, -1, 39, 18, 7, 0, 0),
(-6, -2, -5, 2, -2, 1, 3, 0, 0, 1, 8, 14, 6, 0, 0),
(5, 1, -1, -2, -3, 0, 0, 0, 1, 0, 3, 5, 0, 0, 0),
(-1, 3, 3, 1, -1, -2, 0, 0, 0, -2, 2, 3, 6, 0, 0),
(-4, -4, -3, -10, 8, -2, -8, -1, -1, 2, 33, 20, 15, 0, 0),
(-4, 2, -1, 5, -3, -2, -2, 2, 0, 2, 4, 8, 8, 0, 0),
(-6, -2, 2, -5, -2, 0, -2, 4, 4, -3, 16, 12, 11, 0, 0),
(8, -3, 2, -3, -4, -2, 0, 1, -5, -1, 18, 11, 4, 0, 0),
(-3, 2, -3, 3, 4, -9, 5, -5, 1, -2, 19, 13, 16, 0, 0),
(1, 1, 0, 6, -1, 3, 3, 4, -1, 4, -2, -4, -3, 0, 0),
(2, -2, -4, 4, 1, 8, -8, 1, 0, -8, 16, 17, 14, 0, 0),
(-2, 0, 1, -4, -2, -1, 0, -3, -3, 0, 13, 11, 11, 0, 0),
(-3, -2, -2, 5, -2, -2, -1, -5, 5, 6, 13, 11, 6, 0, 0),
(4, -2, -5, -10, -4, 0, 7, 3, 5, 4, 11, 14, 7, 0, 0),
(-1, -6, -6, 6, -1, -6, 6, -2, -2, -8, 25, 24, 20, 0, 0),
(1, -4, 1, -1, 3, 0, -1, -8, 3, -2, 14, 12, 6, 0, 0),
(3, 1, -1, 0, 2, 0, 0, -1, -1, 3, 0, 1, 3, 0, 0),
(-5, 6, -3, 5, -3, 4, 1, 0, 0, 1, -1, 10, 3, 0, 0),
(1, -2, 1, 0, 3, 0, 2, 3, -4, -3, 6, 7, 4, 0, 0),
(-4, -3, 7, -8, 1, 3, -3, -1, -2, 2, 19, 9, 16, 0, 0),
(0, 2, -3, -1, 1, -2, -1, 1, 0, -2, 9, 6, 6, 0, 0),
(1, 2, 3, 4, -2, 9, 4, -3, 0, -5, 5, 6, 0, 0, 0),
(-1, -1, 2, -2, 1, 0, 1, 0, 1, 1, 3, 1, 2, 0, 0),
(1, 1, 3, 1, -2, -3, -7, -2, 0, -3, 15, 14, 9, 0, 0),
(-2, 6, 1, 5, -3, 1, -3, -1, 0, 1, 0, 6, 4, 0, 0),
(6, -2, -8, 3, 5, -1, -5, -1, 0, 1, 11, 9, 15, 0, 0),
(-4, -2, 2, -2, 3, 0, 1, -2, -1, -1, 11, 6, 5, 0, 0),
(0, -2, -1, 2, 1, -3, 3, 3, -1, -3, 6, 7, 6, 0, 0),
(-1, -1, 0, 0, -3, 3, 1, 2, 2, -1, 5, 4, 2, 0, 0),
(-1, -5, 1, -3, 1, 0, -2, 0, -1, 1, 11, 8, 6, 0, 0),
(1, -3, 7, 2, -4, -9, 3, -4, 1, 11, 17, 15, 5, 0, 0),
(-1, -2, 5, -2, 1, 1, -1, 4, -4, 3, 9, 3, 7, 0, 0)

다음 Table 10.의 선형 제약식은 3개의 입력 비트 $\alpha[0],\alpha[1],\alpha[2]$, 출력 비트 $\beta[0]$, 더미 비트 $\gamma[0]$으로 이루어진 식 $\alpha[0]+\alpha[1]+\alpha[2]+\beta[0]+\gamma[0] \geq 0$을 만족한다.

Table 10. Linear inequality of 3-bit XOR

(-1, -1, 1, -1, 2), (1, -1, -1, -1, 2),
(-1, 1, -1, -1, 2), (1, ,1 ,1 -1, 0),
(-1, -1, -1, 1, 2), (-1, 1, 1, 1, 0),
(1, -1, 1, 1, 0), (1, 1, -1, 1, 0)

〈저자소개〉

조 세 희 (Sehee Cho) 학생회원
2021년 2월: 국민대학교 정보보안암호수학과 졸업
2021년 3월~현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 정보보호, 암호 알고리즘

백 승 준 (Seungjun Baek) 학생회원
2019년 2월: 국민대학교 수학과 졸업
2022년 2월: 국민대학교 금융정보보안학과 석사
2022년 3월~현재: 국민대학교 금융정보보안학과 박사과정
〈관심분야〉 정보보호, 암호 알고리즘

김 종 성 (Jongsung Kim) 종신회원
2006년 11월: K.U.Leuven, ESAT/COSIC 정보보호 전공 공학박사
2007년 2월: 고려대학교 정보보호대학원 공학박사
2009년 9월~2013년 2월: 경남대학교 e-비즈니스학과 교수
2013년 3월~2017년 2월: 국민대학교 수학과 교수
2017년 3월~현재: 국민대학교 정보보안암호수학과/금융정보보안학과 교수
〈관심분야〉 정보보호, 암호 알고리즘, 디지털 포렌식