JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 21, No. 5, pp. 997-1008, May 2020

Check for updates

Optimal Shuffle을 적용한 HIGHT의 연관키 렉탱글 공격에 대한 안전성 분석

백 승 준[1] · 김 한 기[2] · 김 종 성[3*]
[1]국민대학교 금융정보보안학과 석사과정
[2]국민대학교 금융정보보안학과 박사과정
[3]국민대학교 정보보안암호수학과 & 금융정보보안학과 부교수

Security Analysis of Related-Key Rectangle Attack on the Block Cipher HIGHT with Optimal Shuffle

Seungjun Baek[1] · Hangi Kim[2] · Jongsung Kim[3*]

[1]Master's Course, Department of Financial Information, Kookmin University, Seoul 02707, Korea

[2]Ph.D. Course, Department of Financial Information, Kookmin University, Seoul 02707, Korea

[3]Associate Professor, Department of Information Security, Cryptology, and Mathematics & Financial Information, Kookmin University, Seoul 02707, Korea

[요 약]

사물인터넷 기술의 발달로 인해 소형 컴퓨팅 장치가 보편화되고 있으며, 경량 환경에서의 정보보호를 위한 경량 암호 알고리즘의 필요성이 대두되고 있다. HIGHT는 이러한 경량 환경에 적합하도록 2006년 설계된 국산 경량 암호 알고리즘이다. 그러나 해당 알고리즘은 연관키 렉탱글 공격 기법을 사용해 전체 라운드 키 복구가 키 전수조사 시간 복잡도보다 빠르게 가능함이 이론적으로 밝혀진 바 있다. 본 논문에서는 FSE 2010에서 제안된 generalized feistel network의 optimal shuffle을 HIGHT의 구조에 적용한 HIGHT-variant의 안전성에 대해 논하며, 결과적으로 local collision 기법을 이용한 연관키 렉탱글 공격은 HIGHT-variant 분석에 효과적이지 않음을 보인다.

[Abstract]

Due to the development of Internet of Things technology, small computing devices are becoming common, and the necessity of a lightweight cryptographic algorithm for information protection in a lightweight environment is emerging. HIGHT is a lightweight cryptographic algorithm designed for lightweight environments in Korea in 2006. However, for HIGHT, it has been theoretically found that full-round key recovery attack using the related-key rectangle attack technique is faster than the time complexity of brute forcing. In this paper, we discuss the security of the HIGHT-variant which is the application of the generalized feistel network proposed in FSE 2010 to the structure of HIGHT. As a result, we show that the related-key rectangle attack using local collision method is not effective for HIGHT-variant analysis.

색인어 : HIGHT, 블록 암호, GFN, 연관키 렉탱글 공격, Optimal shuffle
Key word : HIGHT, Block cipher, GFN, Related-key rectangle attack, Optimal shuffle

http://dx.doi.org/10.9728/dcs.2020.21.5.997

Received 31 March 2020; **Revised** 15 May 2020
Accepted 25 May 2020

***Corresponding Author; Jongsung Kim**

Tel: +82-2-910-5750
E-mail: jskim@kookmin.ac.kr

Ⅰ. 서 론

사물인터넷 (Internet of Things) 기술의 발달로 모바일, 센서 네트워크, RFID 등 다양한 분야가 함께 부상하고 있으며, 이를 위한 소형 컴퓨팅 장치가 보편화되고 있다. 해당 분야들은 사용자들에게 편리성과 유용성을 제공하기 위해 사용자들의 민감 정보뿐 아니라 각종 생활 정보에 대한 빅데이터를 이용하므로 다양한 형태의 해킹 및 크래킹에 노출될 수 있다. 하지만 대부분의 소형화된 기기는 내부 자원이 제한적이므로 AES[1]와 같은 기존 블록 암호 알고리즘을 사용하기는 어려움이 있다. 따라서 이런 경량 환경에서 보안요소와 성능을 충족시키기 위한 다양한 경량 블록 암호 알고리즘이 제안되고 있다.

HIGHT (HIGh security and light weigHT)는 이러한 저전력, 경량화가 필요한 컴퓨팅 환경에서 기밀성을 제공하기 위해 국내에서 개발한 64-비트 블록 암호이며[2], 2006년 정보통신단체(TTA)의 표준으로 제정되었고[3], 2010년에 국제표준화기구(ISO/IEC) 18033-3의 표준으로 제정되었다[4]. HIGHT는 Generalized Feistel Network (GFN) 구조로 이루어져 있고, ARX (Addition-Rotation-XOR) 기반 8-비트 단위 연산을 진행한다.

현재까지 알려진 HIGHT에 대한 공격으로는 포화 공격, 불능 차분 공격, 연관키 불능 차분 공격, 바이클릭 공격(Biclique Attack), 연관키 렉탱글 공격이 있다. 구체적으로는 포화 공격을 이용한 22-라운드 공격[5], 불능 차분 공격을 이용한 26-라운드 공격[6], 연관키 불능 차분 공격을 이용한 31-라운드 공격[6], 바이클릭 공격을 이용한 전체 32-라운드 공격[7], 연관키 렉탱글 공격을 이용한 전체 32-라운드 공격이 제안되었다[8]. 바이클릭, 연관키 렉탱글 공격만이 HIGHT에 대한 전체 32-라운드 공격에 성공했고, 본 논문에서는 연관키 렉탱글 공격을 다룬다.

HIGHT에 대한 연관키 렉탱글 공격에서는 키 스케줄이 갖는 선형적 성질을 이용해 local collision을 발생시켜 distinguisher를 구성한다. 블록 암호에서 local collision은 어떤 서브키 차분에 의해 생긴 차분이 몇 라운드 후, 다른 서브키 차분에 의해 일정 확률로 상쇄되는 현상을 일컫는다. 연관키 차분을 이용하여 차분 경로를 구성하면 차분을 상쇄할 수 있으며, local collision 기법을 사용했을 때 이를 극대화할 수 있으므로 높은 확률의 긴 차분 경로를 효율적으로 구성할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기 제안된 HIGHT에 대한 전체 32-라운드 연관키 렉탱글 공격을 살펴본다. 3장에서는 HIGHT에 적용된 GFN 구조를 설명하고 8-branch GFN 구조가 가질 수 있는 최적 shuffle을 소개한다. 4장에서는 최적 shuffle을 적용한 HIGHT-variant에 대해서는 local collision 기법을 이용한 연관키 렉탱글 공격이 효과적이지 않음을 보인다. 마지막으로 5장에서는 본 논문에 대한 결론을 맺는다.

Ⅱ. 기 제안된 HIGHT에 대한 전체 라운드 연관키 렉탱글 공격

2004년 연관키 공격과 렉탱글 공격을 결합한 연관키 렉탱글 공격이 소개됐다[9]. 연관키 렉탱글 공격에서는 하나의 낮은 확률을 갖는 긴 연관키 차분 경로 대신 두 개의 높은 확률을 갖는 짧은 연관키 차분 경로를 이용한다. $\{0,1\}^n$을 평문 또는 암호문 공간, $\{0,1\}^k$를 키 공간이라 할 때, 블록 암호 $E: \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$를 두 sub-cipher E_0과 E_1의 연결이라 하자($E = E_1 \circ E_0$). 여기서 K는 임의의 키, ΔK_0와 ΔK_1는 연관키 차분, $E_{0,K}(X)$는 X를 키 K를 통해 E_0로 암호화하는 것을 의미한다. 이때, E_0, E_1의 연관키 차분 경로의 확률을 다음과 같이 정의한다.

$$\hat{p} = \sqrt{\sum_{\beta}\left(\Pr_{X,K}[E_{0,K}(X) \oplus E_{0,K\oplus\Delta K_0}(X\oplus\alpha) = \beta]\right)^2} \qquad (1)$$
$$\hat{q} = \sqrt{\sum_{\gamma}\left(\Pr_{X,K}[E_{1,K}(X) \oplus E_{1,K\oplus\Delta K_1}(X\oplus\gamma) = \delta]\right)^2}$$

연관키 렉탱글 공격은 두 sub-cipher의 연관키 차분 경로를 연결하여 distinguisher를 구성하며, 이 distinguisher를 만족하는 올바른 quartet의 기댓값을 고려한다. N개의 quartet이 생성됐을 경우, 올바른 quartet의 기댓값은 $N \cdot 2^{-n} \cdot \hat{p}^2 \cdot \hat{q}^2$이다. 반면, random cipher의 경우 올바른 quartet의 기댓값은 $N \cdot 2^{-2n}$이 된다. 따라서 $\hat{p} \cdot \hat{q} > 2^{-n/2}$일 때 연관키 렉탱글 공격을 적용할 수 있다.

본 논문에서 사용하는 E_0, E_1의 연관키 차분 경로의 확률 $\hat{p}$, $\hat{q}$는 기존의 연관키 렉탱글 공격의 확률과 다소 다르다. 기존 연관키 렉탱글 공격에서는 $\hat{q}$의 확률 값으로 E_1에 대한 차분 특성과 연관키 차분 특성을 모두 고려했다. 하지만 [8]에서 제안한 공격에서는 연관키 차분이 있는 경우로 한정했으므로 본 논문에서도 해당 경우만 고려한다. 또한 기존의 확률에서는 평문, 암호문, 연관키에 대해 모두 XOR-차분인 경우를 고려하지만, 본 논문에서는 연관키에 대해서 법 2^8에 대한 덧셈-차분을 고려한다. 따라서 ΔK_0, ΔK_1 대신 $\Delta^+ K_0$, $\Delta^+ K_1$을 사용하며, E_0, E_1의 연관키 차분 경로 확률 $\hat{p}$, $\hat{q}$는 다음과 같이 변경된다.

$$\hat{p} = \sqrt{\sum_{\beta}\left(\Pr_{X,K}[E_{0,K}(X) \oplus E_{0,K\boxplus\Delta^+ K_0}(X\oplus\alpha) = \beta]\right)^2} \qquad (2)$$
$$\hat{q} = \sqrt{\sum_{\gamma}\left(\Pr_{X,K}[E_{1,K}(X) \oplus E_{1,K\boxplus\Delta^+ K_1}(X\oplus\gamma) = \delta]\right)^2}$$

본 장에서는 HIGHT 알고리즘과 키 스케줄에서 발견되는 선형적 성질을 설명한다. 그리고 ICISC 2010에서 Koo 등에 의해 제안된 HIGHT에 대한 24-라운드 연관키 렉탱글 distinguisher를 설명한다[8].

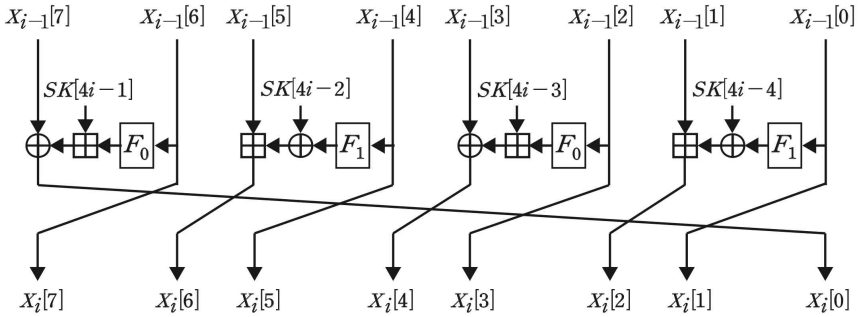

그림 1. HIGHT의 라운드 함수(Type-Ⅱ)
Fig. 1. Round functions of HIGHT(Type-Ⅱ)

2-1 HIGHT Specification

HIGHT[2]는 64-비트 평문, 128-비트 키로부터 64-비트 암호문을 출력한다. 본 논문에서는 다음과 같은 표기법을 사용한다.

- ○ $\oplus$: 배타적 논리합(XOR)
- ○ $\boxplus$: 법 2^8 덧셈(Addition modulo 2^8)
- ○ $\|$: 연접
- ○ P, C : 64-비트 평문, 암호문
- ○ X_i : i 라운드 64-비트 중간값$(0 \le i \le 32)$
- ○ K : 128-비트 키
- ○ Δ : XOR-차분
- ○ Δ^+ : 법 2^8에 대한 덧셈-차분
- ○ $WK[i]$: 8-비트 화이트닝키$(0 \le i \le 7)$
- ○ $SK[i]$: 8-비트 서브키$(0 \le i \le 127)$
- ○ $A^{\lll s}$: A의 s-비트 좌측 순환 이동

평문 P, 암호문 C, 중간값 X_i, 키 K는 다음과 같이 구성된다. 연접되는 각 요소는 모두 1-바이트이다.

$$\begin{aligned} &P = P[7]\|P[6]\|\cdots\|P[0],\\ &C = C[7]\|C[6]\|\cdots\|C[0],\\ &X_i = X_i[7]\|X_i[6]\|\cdots\|X_i[0],\\ &K = K[15]\|K[14]\|\cdots\|K[0]. \end{aligned} \tag{3}$$

1) 키 스케줄

HIGHT의 키 스케줄은 먼저 128-비트 키로부터 초기 변환과 최종 변환에 쓰이는 8개의 8-비트 화이트닝키를 생성한다.

$$\begin{aligned} &WK[i] \leftarrow K[i+12] \quad (0 \le i \le 3)\\ &WK[i] \leftarrow K[i-4] \quad (4 \le i \le 7). \end{aligned} \tag{4}$$

그 후, 키로부터 각 라운드에서 사용하는 128개의 8-비트 서브키를 생성한다.

$$\begin{aligned} &for\ i = 0\ to\ 7\\ &\quad for\ j = 0\ to\ 7\\ &\qquad SK[16i+j] \leftarrow K[j-i \bmod 8] \boxplus \delta[16i+j]\\ &\quad for\ j = 0\ to\ 7\\ &\qquad SK[16i+j+8] \leftarrow K[(j-i \bmod 8)+8] \boxplus \delta[16i+j+8] \end{aligned} \tag{5}$$

여기서 $\delta[i](0 \le i \le 127)$는 LFSR(Linear Feedback Shift Register)을 이용하여 생성하는 8-비트 상수이다.

2) 암호화 과정

HIGHT의 암호화 과정은 다음과 같은 초기 변환 과정으로부터 시작된다.

$$\begin{aligned} &X_0[0] \leftarrow P[0] \boxplus WK[0];\ X_0[2] \leftarrow P[2] \oplus WK[1];\\ &X_0[4] \leftarrow P[4] \boxplus WK[2];\ X_0[6] \leftarrow P[6] \oplus WK[3];\\ &X_0[1] \leftarrow P[1];\ X_0[3] \leftarrow P[3];\\ &X_0[5] \leftarrow P[5];\ X_0[7] \leftarrow P[7]. \end{aligned} \tag{6}$$

HIGHT의 라운드 함수는 아래의 식과 같으며, $i = 1, \cdots, 32$에 대하여 동작한다.(그림 1)

$$\begin{aligned} &X_i[0] \leftarrow X_{i-1}[7] \oplus (F_0(X_{i-1}[6]) \boxplus SK[4i-1]);\\ &X_i[2] \leftarrow X_{i-1}[1] \boxplus (F_1(X_{i-1}[0]) \oplus SK[4i-2]);\\ &X_i[4] \leftarrow X_{i-1}[3] \oplus (F_0(X_{i-1}[2]) \boxplus SK[4i-3]);\\ &X_i[6] \leftarrow X_{i-1}[5] \boxplus (F_1(X_{i-1}[4]) \oplus SK[4i-4]);\\ &X_i[1] \leftarrow X_{i-1}[0];\ X_i[3] \leftarrow X_{i-1}[2];\\ &X_i[5] \leftarrow X_{i-1}[4];\ X_i[7] \leftarrow X_{i-1}[6]. \end{aligned} \tag{7}$$

여기서 F_0과 F_1은 좌측 순환 이동과 XOR을 사용한 함수이다.

$$\begin{cases} F_0(x) = x^{\lll 1} \oplus x^{\lll 2} \oplus x^{\lll 7}\\ F_1(x) = x^{\lll 3} \oplus x^{\lll 4} \oplus x^{\lll 6}. \end{cases} \tag{8}$$

HIGHT 암호화 과정의 마지막인 최종 변환 과정은 다음과 같다.

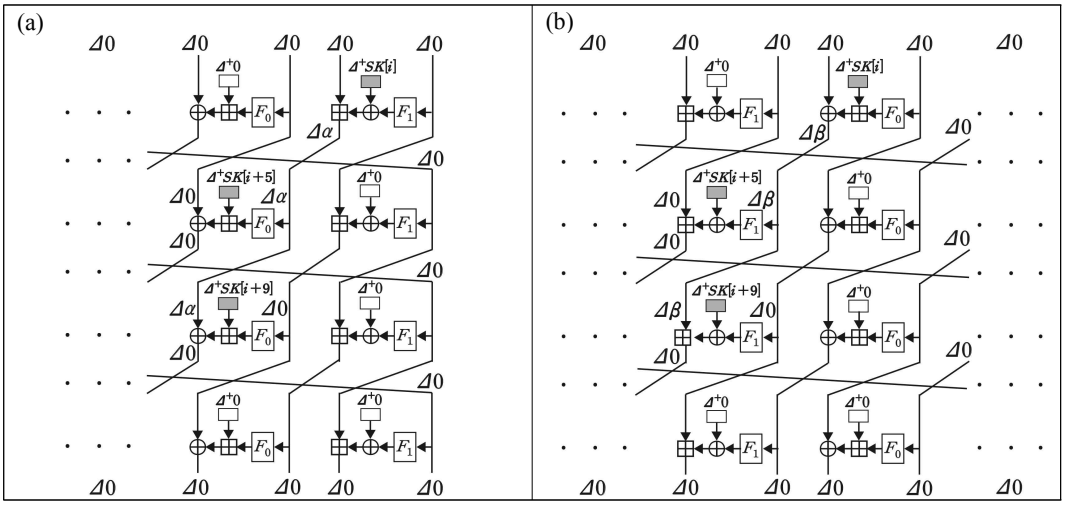

그림 2. HIGHT의 local collisions. (a) Type A local collision (b) Type B local collision
Fig. 2. Local collisions in HIGHT. (a) Type A local collision (b) Type B local collision

$$C[0] \leftarrow X_{32}[1] \boxplus WK[4];\ C[2] \leftarrow X_{32}[3] \oplus WK[5]; \quad (9)$$
$$C[4] \leftarrow X_{32}[5] \boxplus WK[6];\ C[6] \leftarrow X_{32}[7] \oplus WK[7];$$
$$C[1] \leftarrow X_{32}[2];\ C[3] \leftarrow X_{32}[4];$$
$$C[5] \leftarrow X_{32}[6];\ C[7] \leftarrow X_{32}[0].$$

2-2 선형적 성질을 갖는 HIGHT의 키 스케줄

HIGHT의 모든 서브키는 키의 특정 바이트와 LFSR을 통해 미리 정의된 상수 값의 법 2^8 덧셈으로 정의된다. 따라서 서브키의 차분이 키 차분에만 영향을 받는다면, 키 스케줄의 선형적 성질에 의해 특정 키 바이트의 차분이 어떤 서브키에 영향을 주는지 예측할 수 있다. 예를 들어, $i = 4k (k \geq 0$인 정수)이고 특정 키 $K[j]$에 주어진 차분 $\Delta^+K[j]$에 의해 특정 서브키 $SK[i]$도 차분 $\Delta^+SK[i]$를 가지게 됐다고 하자. 그러면 다음과 같은 성질을 만족한다.

$$\Delta^+K[j] = \Delta^+SK[i] = \Delta^+SK[i+17] = \Delta^+SK[i+34] \quad (10)$$
$$= \Delta^+SK[i+51]$$

[8]에서는 본 절에서 서술한 키와 서브키 사이의 성질을 이용하여 local collision을 발생시킨다. 특히 키 스케줄 과정에서 불필요한 확률이 발생하는 상황을 피하고자 연관키에 덧셈-차분을 사용했다. 결과적으로 추가적인 확률 고려 없이 높은 확률의 긴 차분 경로를 구성할 수 있었다.

2-3 HIGHT의 24-라운드 연관키 렉탱글 distinguisher

[8]에서는 확률 $2^{-117.68}$을 갖는 24-라운드 연관키 렉탱글 distinguisher를 구성하고 이를 통해 전체 32-라운드 HIGHT에 대한 공격을 제안한다. distinguisher는 E_0의 8.5-라운드 연관키 부정 차분 경로와 E_1의 15.5-라운드 연관키 부정 차분 경로로 구성되며, 두 경로의 연결 부분에는 ladder switch 기법[10]이 적용되었다.

E_0의 8.5-라운드 연관키 차분 경로는 [11][12]에서 제시한 경로에 기반을 두고 있으며, 확률은 $2^{-12.017}$보다 크다. E_0의 연관키 차분 경로는 (11)과 같다

$$(0x00, 0x00, A, 0x80, 0x00, 0x00, 0x00, 0x00) \quad (11)$$
$$\rightarrow (0x00, 0x00, 0x00, B, 0x80, 0x00, 0x00, 0x00)$$

E_1의 15.5-라운드 연관키 차분 경로는 Type A, Type B 두 가지 4-라운드 local collision을 반복 적용하여 구성한다(그림 2). Type A는 XOR, 법 2^8에 대한 덧셈 순서로 연산하는 branch, Type B는 법 2^8에 대한 덧셈, XOR 순서로 연산하는 branch에 $\Delta^+SK[i]$을 준다. 따라서 $\Delta^+SK[i+5]$, $\Delta^+SK[i+9]$도 Type에 따라 다른 연산 순서를 갖는 branch에 주게 된다. [8]에서는 $K[1]$, $K[5]$, $K[9]$에 각각 $0x10$, $0x68$, $0x10$의 덧셈 차분을 주고 두 local collision의 lower bound를 구한다. 이때 Type A local collision은 $\Delta\alpha$의 값이 $0x10$ 또는 $0x30$일 때만 발생하며, Type B local collision은 $\Delta\beta$의 값이 $0x70$이고 $SK[82] = 0b$???1 1??? 일 때만 발생한다(?는 0 또는 1). 연관키 차분 관계는 식 (12)와 같다.

$$\Delta^+K[1] = \Delta^+SK[69] = \Delta^+SK[86] = \Delta^+SK[103] \quad (12)$$
$$\Delta^+K[5] = \Delta^+SK[65] = \Delta^+SK[82] = \Delta^+SK[99]$$
$$\Delta^+K[9] = \Delta^+SK[60] = \Delta^+SK[77] = \Delta^+SK[94]$$

16~19-라운드에 Type A, 20~23-라운드에 Type B, 24~27-라운드에 Type A local collision을 적용했다. E_1의 연관키 차분 경로는 (13)과 같다.

$$(0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, C) \quad (13)$$
$$\rightarrow (0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00)$$

C 부분에는 총 4개의 값을 고려할 수 있으며, 경로의 확률은 $2^{-41.661}$ 보다 크다.

Ⅲ. GFN 구조

본 장에서는 HIGHT 알고리즘에 쓰인 GFN 구조에 대해 설명한다. 또한 8-branch GFN Type-Ⅱ 구조에 대한 최적의 shuffle을 제시한다.

3-1 GFN 구조와 shuffle

CRYPTO 1989에서 Zheng 등은 GFN 구조를 처음 제안했다[13]. GFN 구조는 기존의 Feistel 구조의 branch 수를 2개에서 k개로 확장한 구조이다. 그들은 논문에서 총 3가지 GFN 구조를 제시했고, HIGHT에서는 그 중 8-branch GFN Type-Ⅱ 구조를 채택했다.

이후 ASIACRYPT 1996에서는 Nyberg가 단순한 회전 시프트가 아닌 다른 shuffle을 갖는 GFN Type-Ⅱ 구조들을 제안했다[14]. TWINE, Piccolo와 같은 블록 암호를 만드는 데 사용되었으나 한 가지 shuffle만을 고려했다는 한계가 존재했다. FSE 2010에서 Suzaki 등은 더 일반적인 shuffle들을 고려했다[15]. 그들은 각각의 모든 암호문 부분 블록이 모든 평문 블록에 의존하게 되는 최소 라운드를 확산 라운드라고 정의하고, 최소 확산 라운드를 갖는 최적의 shuffle을 찾기 위한 연구를 집중적으로 진행했다.

3-2 8-branch에 대한 최적의 shuffle

[15]에서는 전수조사를 통해 8-branch GFN 구조에서 가장 안전성이 높은 No.1 shuffle과 De Bruijn graph를 이용해 찾은 또 다른 No.2 shuffle을 제시한다(표 1).

표 1에서 D는 최소 확산 라운드, IDC는 불능 차분 경로의 최대 라운드, SC는 포화 공격의 최대 라운드, AS_D는 20-라운드 차분 공격(DC)에 대한 최소 active S-box의 개수, AS_L는 20-라운드 선형 공격(LC)에 대한 최소 active S-box의 개수를 의미한다.

표 1. 최적 블록 셔플(8-branch)
Table 1. Optimum Block Shuffles(8-branch)

$k=8$	shuffle	D	IDC	SC	AS_D	AS_L
Type-Ⅱ	{0,7,6,5,4,3,2,1}	8	17	16	27	27
Nyberg	{6,4,7,2,5,0,3,1}	8	14	15	18	18
No.1	{4,7,6,3,0,5,2,1}	6	11	11	30	30
No.2	{2,5,4,3,0,7,6,1}	6	10	11	26	26

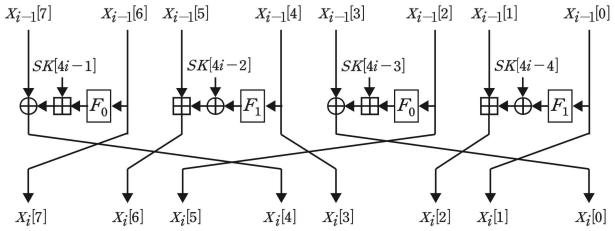

그림 3. HIGHT-variant의 라운드 함수(No.1)
Fig. 3. Round functions of HIGHT-variant(No.1)

D가 작을수록 더 빠르고 효율적인 차분의 확산이 일어나며, IDC와 SC가 작을수록 구성할 수 있는 distinguisher가 짧아지므로 전체 cipher에 대한 공격이 어려워진다. AS_D, AS_L가 커지면 고려해야 하는 확률의 개수가 많아져 DC distinguisher와 LC distinguisher가 짧아지므로 공격이 어려워진다.

Nyberg의 shuffle은 기존의 Type-Ⅱ shuffle보다 IDC, SC 관점에서는 향상됐지만, DC, LC 관점에서는 더 취약해졌다. No.1, No.2 shuffle은 Nyberg의 shuffle보다 여러 측면에서 향상됐다. De Bruijn graph를 통해 제시한 No.2 shuffle은 No.1 shuffle보다 IDC 관점에서는 안전하지만, DC, LC 관점에서는 더 취약하다. 따라서 본 논문에서는 No.1 shuffle을 HIGHT의 최적 shuffle로 정의하며, 최적 shuffle을 적용한 HIGHT의 라운드 함수는 그림 3과 같다.

Ⅳ. 최적 shuffle이 적용된 HIGHT-variant에 대한 기존 연관키 렉탱글 공격의 적용 가능성

본 장에서는 HIGHT의 기존 shuffle을 최적 shuffle로 바꾼 HIGHT-variant를 고려한다. 또한 HIGHT-variant에서 관찰할 수 있는 local collision들을 제시하고, 2.3절과 같은 방식으로 local collision의 반복 교차 적용을 통해 연관키 차분 경로를 구성할 수 있는지 분석한다. 마지막으로 구체적인 distinguisher를 제시하여 HIGHT-variant에서는 연관키 렉탱글 공격을 통한 전체 32-라운드 공격을 방지할 수 있음을 밝힌다.

4-1 HIGHT-variant에서 관찰할 수 있는 4-라운드 local collisions

HIGHT의 shuffle은 규칙적인 단순 회전 시프트이므로 각 라운드 함수의 XOR과 법 2^8에 대한 덧셈 순서만 고려하면 다른 local collision을 구성할 수 있었다. 하지만 HIGHT-variant의 shuffle은 주목할만한 규칙성이 없으므로 한 라운드 함수에 연관키 차분을 줬을 때 기존보다 다양한 양상으로 차분이 퍼지게 된다. 따라서 HIGHT-variant에서는 어떤 branch에 연관키 차분을 주는지에 따라 4가지의 local collision을 관찰할 수 있다.

1) Type A

$X_i[0]$과 $X_i[1]$이 연산 되는 라운드 함수에 연관키 차분을 주는 경우 그림 4와 같은 Type A local collision을 관찰할 수 있다.

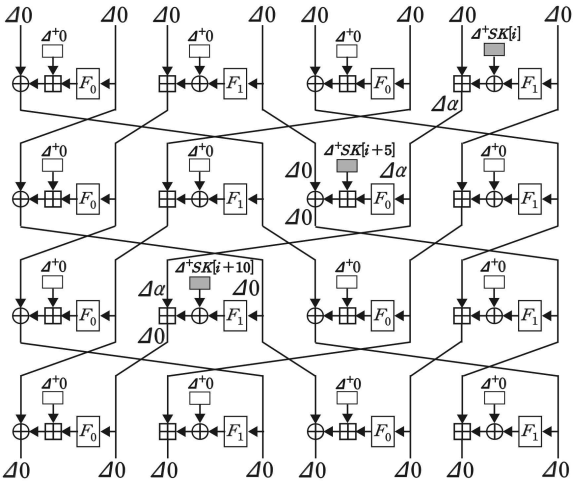

그림 4. HIGHT-variant의 Type A local collision
Fig. 4. Type A local collision in HIGHT-variant

2) Type B

$X_i[2]$와 $X_i[3]$가 연산 되는 라운드 함수에 연관키 차분을 주는 경우 그림 5와 같은 Type B local collision을 관찰할 수 있다.

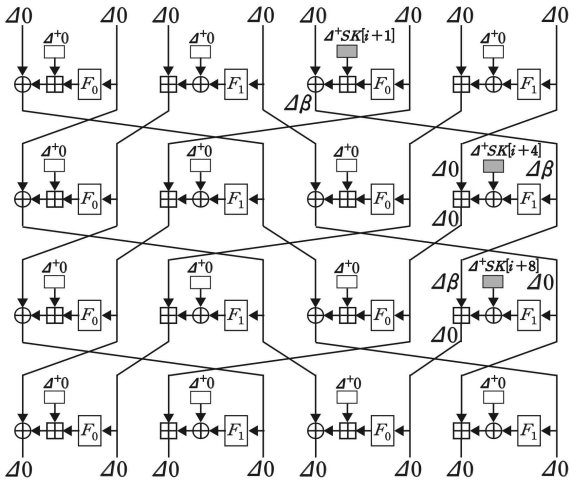

그림 5. HIGHT-variant의 Type B local collision
Fig. 5. Type B local collision in HIGHT-variant

3) Type C

$X_i[4]$와 $X_i[5]$가 연산 되는 라운드 함수에 연관키 차분을 주는 경우 그림 6과 같은 Type C local collision을 관찰할 수 있다.

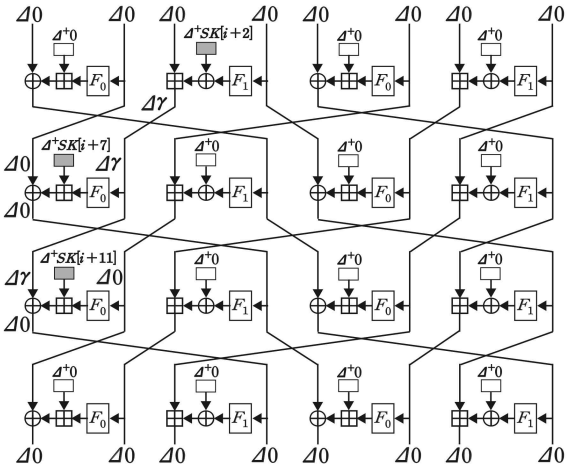

그림 6. HIGHT-variant의 Type C local collision
Fig. 6. Type C local collision in HIGHT-variant

4) Type D

$X_i[6]$와 $X_i[7]$가 연산 되는 라운드 함수에 연관키 차분을 주는 경우 그림 7과 같은 Type D local collision을 관찰할 수 있다.

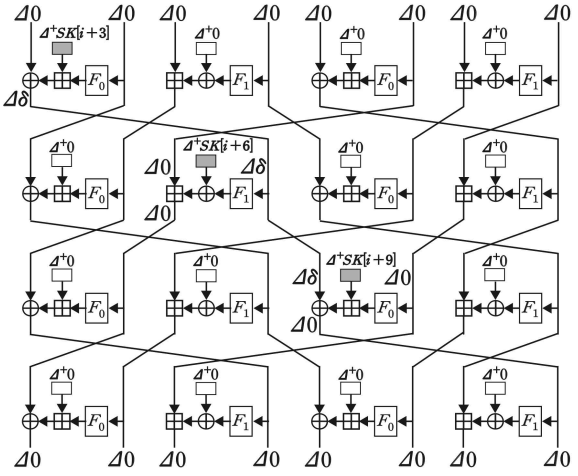

그림 7. HIGHT-variant의 Type D local collision
Fig. 7. Type D local collision in HIGHT-variant

4-2 4-라운드 local collision의 반복 적용을 통한 E_1의 연관키 차분 경로의 구성 가능성

본 절에서는 HIGHT-variant에 대해 2.3절과 같이 local collision을 반복 적용하여 E_1의 연관키 차분 경로를 구성할 수 있는지 분석한다. 특히 반복 적용이 가능한 연관키 차분을 생성하더라도 유의미한 횟수만큼 반복하기 위해서는 그 경로의 차분 확률이 충분히 높아야 한다. 따라서 Type A, B, C, D와 같이 차분이 시작되는 라운드의 연관키 차분이 한 바이트에만 주어지는 반복 차분 경로만을 고려한다.

첫 local collision을 Type A로 선택하고 뒤에 또 다른 local collision을 적용하여 distinguisher를 구성하는 과정을 생각해보자. 이때, 연관키 차분은 첫 local collision 경로를 따라 주어져야 한다. 따라서 식 (14)와 같은 연관키 차분을 생각할 수 있다.

$$\begin{aligned}&\Delta^+K[j_1] = \Delta^+SK[i] = \Delta^+SK[i+17] \\ &= \Delta^+SK[i+34] \\ &\Delta^+K[j_2] = \Delta^+SK[i+5] = \Delta^+SK[i+22] \\ &= \Delta^+SK[i+39] \\ &\Delta^+K[j_3] = \Delta^+SK[i+10] = \Delta^+SK[i+27]\end{aligned} \tag{14}$$

같은 Type의 local collision을 두 번 연속적으로 적용할 수 있는지 확인하기 위해 위 수식과 같이 처음 경로에 Type A local collision이 발생했다고 가정하자. 두 번째 local collision에도 Type A local collision을 발생시키기 위해서는 해당 local collision의 경로가 $\Delta^+SK[i+17]$, $\Delta^+SK[i+22]$, $\Delta^+SK[i+27]$ 연관키 차분이 발생하는 경로와 일치해야 한다. 그러나 바뀐 shuffle을 고려했을 때 해당 상황은 불가능하다. 따라서 HIGHT-variant에서는 Type A local collision를 연속적으로 적용하여 8-라운드 distinguisher를 구성할 수 없다(그림 8).

다른 Type의 local collision을 두 번 연속적으로 적용할 수 있는지 확인하기 위해 이전 방법과 같이 첫 번째 local collision은 Type A, 두 번째는 Type B를 적용한다고 하자. 이때 Type B와 같은 local collision이 발생하기 위해서는 $SK[i+20]$, $SK[i+24]$에 연관키 차분이 추가적으로 주어져야 한다. 이것은 키 스케줄의 의해 $\Delta^+SK[i+7]$, $\Delta^+SK[i+11]$, $\Delta^+SK[i+37]$, $\Delta^+SK[i+41]$ 연관키 차분도 주어진다는 것을 의미한다. 유사한 방식으로 연관키 차분을 반복 적용하기 위해서 총 10-바이트의 연관키 차분이 추가로 주어진다. 이때 13-라운드를 기준으로 최소 Type A가 1번, Type B가 2번, Type C가 1번. Type D가 1번 발생하게 되며, 그 외 상쇄되지 않는 연관키 차분도 존재하게 된다(그림 8).

구체적으로 다른 Type의 local collision이 두 번 연속적으로 적용되는 경우에 대해 E_1의 13-라운드 연관키 차분 경로가 가지는 확률 upper bound를 구해보자. 관련 컴퓨터 계산을 통해

표 2. HIGHT-variant의 4라운드 local collision에 대한 상한 확률
Table 2. Upper bounds of probability of 4-round local collisions in HIGHT-variant

Type	Probability upper bound
Type A	$2^{-5.59061}$
Type B	2^{-6}
Type C	$2^{-4.67807}$
Type D	2^{-6}

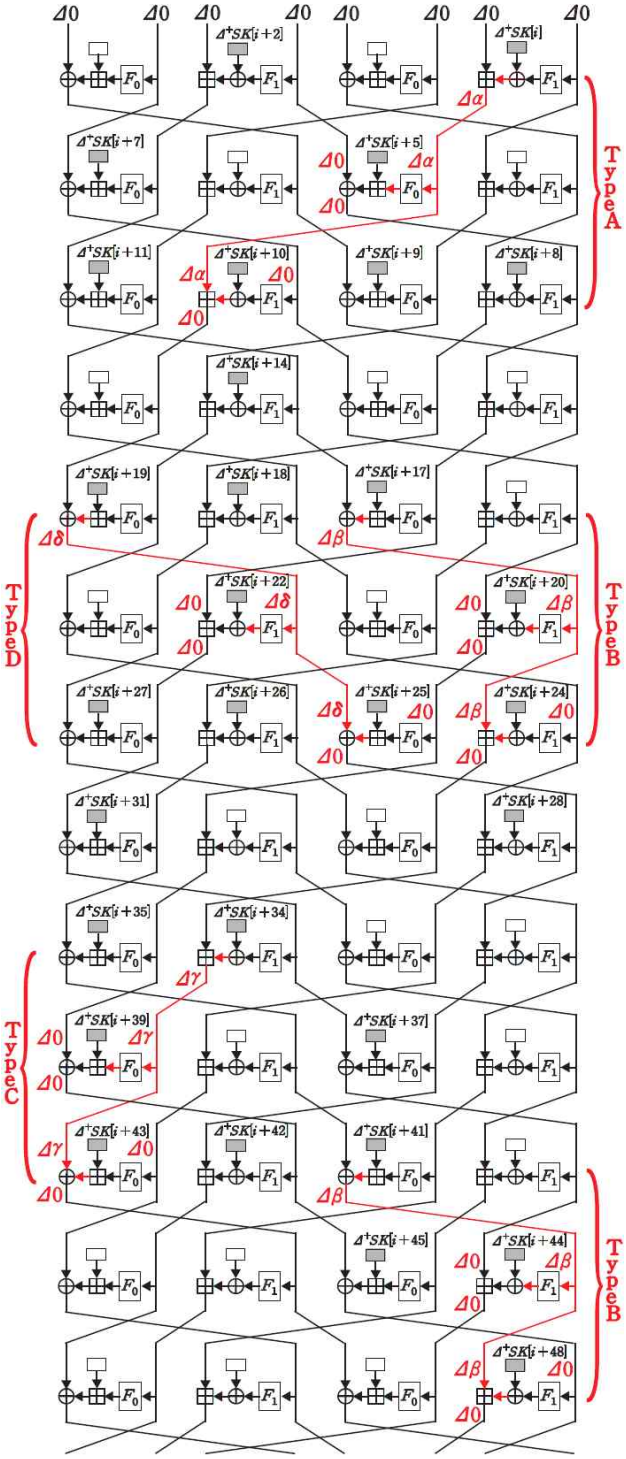

그림 8. HIGHT-variant의 13라운드 local collision 구성의 불가능성
Fig. 8. Infeasibility of local collisions in 13-round HIGHT-variant

표 3. 23라운드 연관키 distinguisher의 상한 확률
Table 3. Upper bounds of probability of 23-round related-key distinguisher

Initial type	Upper bound
Type A	$2^{-70.19736}$
Type B	$2^{-70.19736}$
Type C	$2^{-69.37858}$
Type D	$2^{-67.5535}$

HIGHT-variant에서 관찰할 수 있는 Type A, Type B, Type C, Type D의 확률 upper bound를 계산했으며, 내용은 표 2와 같다.

local collision만을 고려한 E_1의 두 연관키 차분 경로의 확률 $\hat{q}^2$는 다음과 같이 계산된다.

$$\hat{q}^2 < 2^{(-5.59061-12-4.67807-6)\times 2} = 2^{-56.53736} \quad (15)$$

HIGHT-variant의 shuffle을 고려하여 변경된 E_0의 8.5-라운드 연관키 차분 경로와 그림 8과 같은 E_1의 연관키 차분 경로를 고려하자. ladder switch 기법을 고려하면 두 경로를 통해 23-라운드 연관키 렉탱글 distinguisher를 구성할 수 있다. E_0의 두 연관키 차분 경로의 확률 $\hat{p}^2$는 $\hat{p}^2 < 2^{-13.660}$를 만족하고, E_1의 두 연관키 차분 경로의 확률 $\hat{q}^2$의 upper bound를 알고 있으므로 23-라운드 연관키 렉탱글 distinguisher의 확률은 $\hat{p}^2 \cdot \hat{q}^2 < 2^{-70.19735} < 2^{-64}$가 성립함을 알 수 있다. E_1의 연관키 차분 경로의 첫 local collision이 다른 Type인 경우에 대한 $\hat{p}^2 \cdot \hat{q}^2$의 upper bound는 표 3와 같다. 따라서 기존의 4-라운드 local collision을 반복 적용하는 방식으로는 E_1의 효과적인 연관키 차분 경로를 구성할 수 없다는 것을 알 수 있다.

4-3 HIGHT-variant의 연관키 렉탱글 공격 distinguisher

그림 9과 그림 10은 HIGHT-variant에 대해 본 논문에서 제시하는 E_0, E_1의 연관키 차분 경로이며, 이를 통해 15-라운드 연관키 렉탱글 distinguisher를 구성할 수 있다. E_0의 연관키 차분 경로는 [8][11][12]에서 사용한 경로 구성을 본 셔플에 적합하게 응용하여 기존과 동일한 라운드 수를 가지며, 키 복구 공격을 고려하여 4라운드에서 시작한다. 또한 높은 확률의 긴 차분 경로를 구성하기 위해 부정 차분과 ladder switch 기법을 적용했다.

E_1의 높은 확률을 갖는 긴 연관키 차분 경로를 찾기 위해 SAT Solver[1]를 사용했으며, Espresso 알고리즘 사용해서 HIGHT의 법 2^8 덧셈 제약식을 56개로 축소하여 조사의 효율성을 높였다[17]. 만약 local collision이 4번 이상 발생하고 E_1 경로의 입력, 출력 바이트에 연관키에 따른 부정 차분을 고려해야 한다면 표 2의 확률을 고려할 때 연관키 렉탱글 공격을 위한 확률 수치를 만족할 수 없다. 따라서 최대 3번의 local collision을 가지면서 local collision 이외의 연관키 차분은 존재하지 않고 최대 라운드를 갖는 경로를 조사했다. 결과적으로 7라운드 이상 길이를 갖는 E_1의 연관키 차분 경로는 구성할 수 없음을 알 수 있었다. 따라서 본 절에서 제시하는 distinguisher와 같이, HIGHT-variant에 대해 기대할 수 있는 local collision을 이용한 연관키 렉탱글 공격 최대 라운드 수는 15라운드이다.

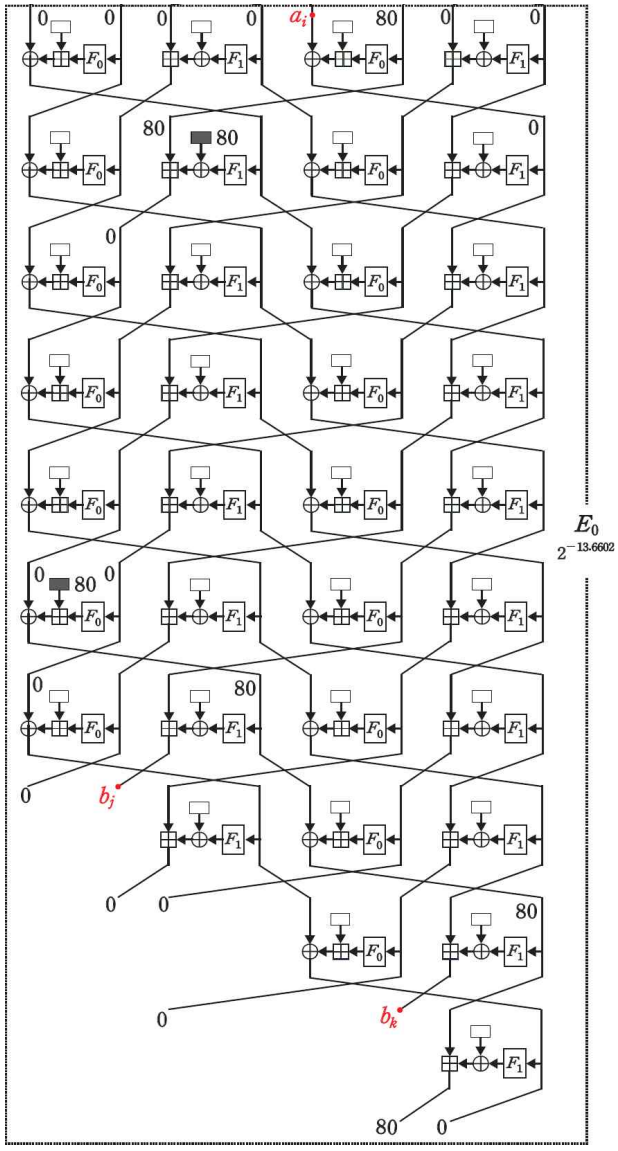

그림 9. HIGHT-variant에서 구성할 수 있는 E_0의 8.5라운드 연관키 차분 경로
Fig. 9. 8.5-round Related-key differential trail for E_0 in HIGHT-variant

1) SAT Solver는 논리곱(Conjunctive Normal Form, CNF)으로 표현된 이진 충족 가능성 문제(Boolean Satisfiability Problem, SAT)에 대한 최적해를 찾는 도구이다[16].

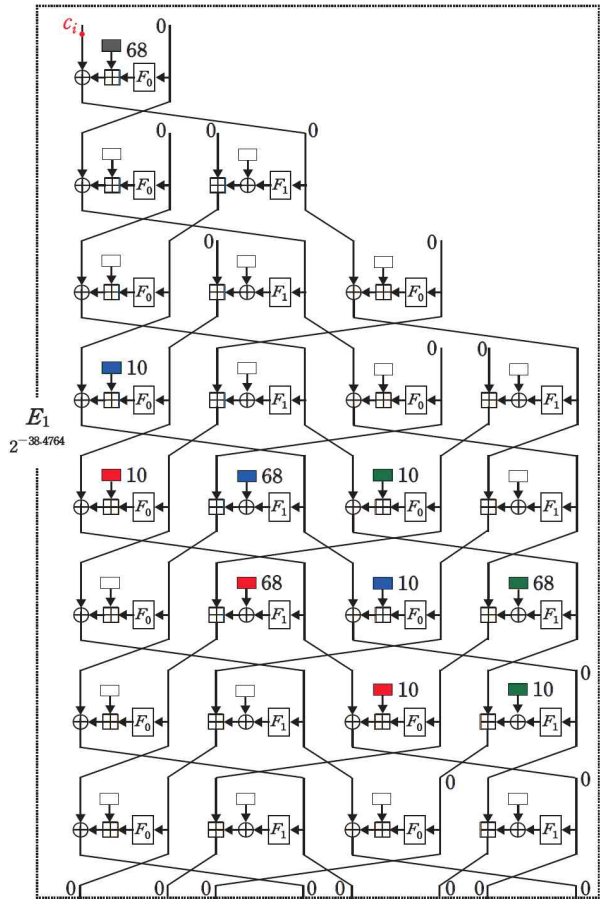

그림 10. HIGHT-variant에서 구성할 수 있는 E_1의 6.5라운드 연관키 차분 경로

Fig. 10. 6.5-round Related-key differential trail for E_1 in HIGHT-variant

E_0, E_1의 연관키 차분 경로에 대한 입력, 출력 바이트와 해당 차분 값은 표 4와 같으며 차분 집합 ΔA, ΔB, ΔC 은 다음과 같이 16진수 값의 집합 표현으로 정의한다.

$$\begin{aligned}\Delta\mathrm{A} &= \{41,43,c1,c3\},\\ \Delta\mathrm{B} &= \{14,1c,24,2c,34,3c,54,5c,64,6c,74,7c,d4,dc,\\ &\quad e4,ec,f4,fc\},\\ \Delta\mathrm{C} &= \{68,78,98,a8,b8,e8,f8\}.\end{aligned} \tag{16}$$

1) E_0의 연관키 차분 경로의 확률

E_0의 두 연관키 차분 경로의 확률을 구하기 위해서는 $\Delta X_4[0]$, $\Delta X_{10}[6]$, $\Delta X_{12}[2]$의 값을 분석할 필요가 있다. 따라서 모든 $a_i \in \Delta\mathrm{A}(0 \le i \le 3)$, $b_j, b_k \in \Delta\mathrm{B}\ (0 \le j, k \le 17)$에 대하여 확률 u_i, v_j, v_k를 다음과 같이 정의한다.

$$\begin{aligned}u_i &:= \mathrm{P}[\Delta X_4[0] = 0 \mid \Delta X_3[3] = a_i] \times \mathrm{P}[\Delta X_3[3] = a_i]\\ v_j &:= \mathrm{P}[\Delta X_{10}[6] = b_j]\\ v_k &:= \mathrm{P}[\Delta X_{12}[2] = b_k]\end{aligned} \tag{17}$$

이때 E_0의 두 연관키 차분 경로의 확률 $\hat{p}^2$는 다음과 같이 계산된다.

$$\begin{aligned}\hat{p}^2 &= \left(\sum_{i=0}^{3} u_i\right)^2 \cdot \sum_{j=0}^{17} v_j^2 \cdot \sum_{k=0}^{17} v_k^2\\ &\ge 2^{-6} \times 2^{-3.83007} \times 2^{-3.83007}\\ &> 2^{-13.6602}\end{aligned} \tag{18}$$

표 4. E_0과 E_1에 관한 distinguisher의 입/출력 바이트 위치와 차분 값

Table 4. Byte positions and differences both inputs and outputs for distinguishers of E_0 and E_1

E_0	Positions	Input	$(X_3[7], X_3[6], X_3[5], X_3[4], X_3[3], X_3[2], X_3[1], X_3[0])$
		Output	$(X_{10}[7], X_{10}[6], X_{11}[6], X_{11}[5], X_{12}[5], X_{12}[2], X_{13}[2], X_{13}[1])$
	Differences	Input	$(0x00, 0x00, 0x00, 0x00, a_i, 0x80, 0x00, 0x00)$
		Output	$(0x00, b_j, 0x00, 0x00, 0x00, b_k, 0x80, 0x00)$
Δ^+K_0 : $(0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x00)$			
E_1	Positions	Input	$(X_{10}[7], X_{10}[6], X_{11}[6], X_{11}[5], X_{12}[5], X_{12}[2], X_{13}[2], X_{13}[1])$
		Output	$(X_{17}[7], X_{17}[6], X_{17}[5], X_{17}[4], X_{17}[3], X_{17}[2], X_{17}[1], X_{17}[0])$
	Differences	Input	$(c_i, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00)$
		Output	$(0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00)$
Δ^+K_1 : $(0x68, 0x10, 0x00, 0x00, 0x68, 0x10, 0x68, 0x10, 0x00, 0x00, 0x10, 0x10, 0x00, 0x00, 0x00, 0x00)$			

2) E_1의 연관키 차분 경로의 확률

E_1의 두 연관키 차분 경로의 확률을 구하기 위해서는 $\Delta X_{11}[4]$ 부분과 local collision 발생 시 고려해야 하는 확률을 종합적으로 분석해야 한다. 본 경로에서는 14~16, 15~17라운드에서 Type D, 15~17라운드에서 Type B local collision을 발생시킨다. 모든 $c_i \in \Delta C\ (0 \le i \le 6)$에 대하여 확률 w_i, q_l를 다음과 같이 정의한다.

$$w_i := \mathrm{P}[\Delta X_{11}[4] = 0 \mid \Delta X_{10}[7] = c_i] \tag{19}$$
$q_l :=$ local collision에 의한 14~17라운드 연관키 차분 경로 확률

이때 E_1의 두 연관키 차분 경로의 확률 $\hat{q}^2$는 다음과 같이 계산된다.

$$\begin{aligned}\hat{q}^2 &= \sum_{i=0}^{6} w_i^2 \cdot q_l^2 \\ &\ge 2^{-2.47643} \times 2^{-36} \\ &> 2^{-38.4764}\end{aligned} \tag{20}$$

3) 전체 distinguisher의 확률

E_0의 8.5-라운드 연관키 차분 경로와 E_1의 6.5-라운드 연관키 차분 경로를 통해 15-라운드 연관키 렉탱글 공격 distinguisher를 구성할 수 있으며 해당 distinguisher의 확률 lower bound는 다음과 같이 계산된다.

$$\begin{aligned}\hat{p}^2 \cdot 2^{-64} \cdot \hat{q}^2 &> 2^{-13.6602-64-38.4764} \\ &= 2^{-116.1366} \\ &> 2^{-116.136}\end{aligned} \tag{21}$$

따라서 $2^{118.136}$개의 quartet을 생성하면 키 복구 공격이 가능하다. 하지만 HIGHT의 전체 라운드 수는 32라운드이므로, 위의 15라운드 distinguisher를 이용한 전체 라운드 키 복구 공격으로의 확장은 불가능하다.

Ⅴ. 결 론

본 논문에서는 HIGHT의 GFN 구조에 최적 shuffle을 적용한 HIGHT-variant를 제안했다. HIGHT-variant에서는 4-라운드 local collision을 반복 적용하는 방식으로 효과적인 연관키 차분 경로를 구성할 수 없었다(표 5).

또한 $2^{-116.136}$의 확률을 갖는 15-라운드 연관키 렉탱글 공격 distinguisher를 구성했다. 하지만 전체 라운드 공격을 위해서는 부족함을 알 수 있었는데, 이는 최적 shuffle로 인하여 반복적인 local collision 경로를 구성하기 어려운 점에 기인한다. 따라서 HIGHT에 최적 shuffle을 적용했을 때 local collision 기법을 이용한 전체 라운드 연관키 렉탱글 공격을 방지할 수 있어, 안전성이 향상될 수 있음을 알 수 있었다.

GFN 구조는 확산이 느리다는 단점이 있으나 이를 향상시키기 위한 연구는 지속해서 진행되고 있으며, 최신 관련 연구에서 branch 수에 따른 최적 shuffle을 제시하고 있다[18]. 본 논문에서 제시한 바와 같이 최적 shuffle을 암호 알고리즘에 적용했을 때 기존보다 향상된 안전성을 기대할 수 있고, 암호 설계자의 입장에서는 더 적은 라운드 수를 사용하여 암호의 성능을 향상시킬 수 있다. 따라서 GFN 구조를 갖는 기존 암호 알고리즘들에 대해 최적 shuffle을 적용한 후 차분, 선형 공격 등 여러 공격에 대한 안전성 연구를 수행할 필요가 있다.

표 5. 전체 요약
Table 5. Total summary

Cipher	Method	Probability bound	References
HIGHT	Rel.-Key Rec.	$\hat{p}^2 \cdot \hat{q}^2 > 2^{-53.678} > 2^{-64}$	[5]
HIGHT-variant	Type A	$\hat{p}^2 \cdot \hat{q}^2 < 2^{-70.19736} < 2^{-64}$	This paper
	Type B	$\hat{p}^2 \cdot \hat{q}^2 < 2^{-70.19736} < 2^{-64}$	
	Type C	$\hat{p}^2 \cdot \hat{q}^2 < 2^{-69.37858} < 2^{-64}$	
	Type D	$\hat{p}^2 \cdot \hat{q}^2 < 2^{-67.5535} < 2^{-64}$	

감사의 글

본 연구는 고려대 암호기술 특화연구센터(UD170109ED)를 통한 방위사업청과 국방과학연구소의 연구비 지원으로 수행되었습니다.

참고문헌

[1] NIST, “Advanced Encryption Standard,” FIPS-197, Nov. 2001.

[2] Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bon-Seok Koo, Changhoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim and Seongtaek Chee, “HIGHT: A New Block Cipher Suitable for Low-Resource Device,” CHES’06, LNCS 4249, pp. 46-59, Oct. 2006.

[3] Telecommunications Technology Association, “TTAK.KO-12.0040/R1(revised),” TTAS, 2008.

[4] International Organization for Standadization, “ISO/IEC 18033-3:2010,” ISO/IEC 18033-3, 2010.

[5] Peng Zhang, Bing sun and Chao Li, “Saturation Attack on the Block Cipher HIGHT,” CANS’09, LNCS 5888, pp. 76-86, Dec. 2009.

[6] Onur Özen, Kerem Varıcı, Cihangir Tezcan and Çelebi Kocair, “Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced Round PRESENT and HIGHT,” ACISP’09, LNCS 5594, pp. 90-107, Jul. 2009.

[7] Deukjo Hong, Bonwook Koo, Daesun kwon, “Biclique Attack on the Full HIGHT,” ICISC’11, pp. 365-374, Nov. 2011.

[8] Bonwook Koo, Deukjo Hong and Daesung Kwon, “Related-Key Attack on the Full HIGHT,” ICISC’10, pp. 49-67, Dec. 2010.

[9] Jongsung Kim, Guil Kim, Seokhie Hong, Sangjin Lee and Dowon Hong, “The Related-Key Rectangle Attack − Application to SHACAL-1,” ACISP’04, LNCS 3108, pp. 123-136, Jul. 2004.

[10] Alex Biryukov and Dmitry Khovratovich, “Related-key Cryptanalysis of the Full AES-192 and AES-256,” ASIACRYPT’09, LNCS 5912, pp. 1-18, 2009.

[11] Jiqiang Lu, “Cryptanalysis of reduced versions of the HIGHT block cipher from CHES 2006,” ICISC’07, pp. 11-26, 2007.

[12] Jiqiang Lu, “Cryptanalysis of Block Ciphers,” Ph.D. Thesis, Royal Holloway, University of London, Jul. 2008.

[13] Yuliang Zheng, Tsutomu Matsumoto and Hideki Imai, “On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses,” CRYPTO’89, LNCS 435, pp. 461-480, 1989.

[14] Kaisa Nyberg, “Generalized Feistel Networks,” ASIACRYPT’96, LNCS 1163, pp. 91-104, 1996.

[15] Tomoyasu Suzaki and Kazuhiko Minematsu, “Improving the Generalized Feistel,” FSE’10, LNCS 6147, pp. 19-39, 2010.

[16] Mate Soos, Karsten Nohl and Claude Castelluccia, “Extending SAT Solvers to Cryptographic Problems,” SAT’09, LNCS 5584, pp. 244-257, 2009.

[17] Robert K. Brayton, Gary D. Hachtel, Curtis T. Mcmullen and Alberto L. Sangiovanni-vincentelli, “Logic Minimization Algorithms for VLSI Synthesis,” *Springer Science & Business Media*, Vol. 2. 1984.

[18] Patrick Derbez, Pierre-Alain Fouque, Baptiste Lambin and Victor Mollimard, “Efficient Search for Optimal Diffusion Layers of Generalized Feistel Networks,” IACR Transactions on Symmetric Cryptology, pp. 218-240, 2019.

백승준(Seungjun Baek)

2019년 2월 : 국민대학교 수학과 졸업

2020년 3월~현 재 : 국민대학교 금융정보보안학과 석사과정
※관심분야 : 정보보호, 암호 알고리즘

김한기(Hangi Kim)

2016년 2월 : 국민대학교 수학과 졸업
2018년 2월 : 국민대학교 금융정보보안학과 석사

2018년 3월~현 재 : 국민대학교 금융정보보안학과 박사과정
※관심분야 : 정보보호, 암호 알고리즘

김종성(Jongsung Kim)

2000년 8월/2002년 8월 : 고려대학교 수학 전공 학사/이학석사
2006년 11월 : K.U.Leuven. ESAT/SCD-COSIC 정보보호 전공 공학박사
2007년 2월 : 고려대학교 정보보호대학원 공학박사

2007년 3월~2009년 8월 : 고려대학교 정보보호기술연구센터 연구교수
2009년 9월~2013년 2월 : 경남대학교 e-비즈니스학과 조교수
2013년 3월~2017년 2월 : 국민대학교 수학과 부교수
2014년 3월~현 재 : 국민대학교 일반대학원 금융정보보안학과 부교수
2017년 3월~현 재 : 국민대학교 정보보안암호수학과 부교수
※관심분야 : 정보보호, 암호 알고리즘, 디지털 포렌식