Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.25, NO.2, Apr. 2015 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2015.25.2.457

HIGHT에 대한 부채널 분석 및 대응 방법*

김 태 종,[1†] 원 유 승,[1] 박 진 학,[1] 안 현 진,[1] 한 동 국[1,2‡]
[1]국민대학교 금융정보보안학과, [2]국민대학교 수학과

Side Channel Attacks on HIGHT and Its Countermeasures*

Tae-jong Kim,[1†] Yoo-seung Won,[1] Jin-hak Park,[1] Hyun-jin An,[1] Dong-guk Han[1,2‡]
[1]Dept. of Financial Information Security, Kookmin University,
[2]Dept. of Mathematics, Kookmin University

요 약

사물인터넷은 다양한 장비에서 통신이 가능해야 한다. 사물인터넷 통신환경에서도 보안적인 요소를 고려해야하기 때문에 다양한 장비에 알맞은 암호 알고리즘이 필요하다. 그러므로 경량 블록 암호 알고리즘은 임베디드 플랫폼 사이에서 안전한 통신을 위하여 필수적이다. 그러나 이러한 환경에서 사용되는 경량블록알고리즘은 부채널 분석에 대한 취약점이 존재할 수 있다. 그렇기 때문에 부채널 대응기법을 고려하지 않을 수가 없다. 본 논문에서는 국산 경량암호 알고리즘인 ARX구조의 HIGHT 알고리즘에 대한 1차 전력분석 방법들을 제시하고 그 취약점을 확인한다. 또한, 1차 전력분석에 안전하도록 HIGHT를 설계하는 방법을 제안한다. 마지막으로, AES 와의 성능비교를 통하여 얼마나 효율성을 갖는지에 대해서 설명한다.

ABSTRACT

Internet of Things(IoT) technologies should be able to communication with various embedded platforms. We will need to select an appropriate cryptographic algorithm in various embedded environments because we should consider security elements in IoT communications. Therefore the lightweight block cryptographic algorithm is essential for secure communication between these kinds of embedded platforms. However, the lightweight block cryptographic algorithm has a vulnerability which can be leaked in side channel analysis. Thus we also have to consider side channel countermeasure. In this paper, we will propose the scenario of side channel analysis and confirm the vulnerability for HIGHT algorithm which is composed of ARX structure. Additionally, we will suggest countermeasure for HIGHT against side channel analysis. Finally, we will explain how much the effectiveness can be provided through comparison between countermeasure for AES and HIGHT.

Keywords: Side Channel Analysis, HIGHT, Masking Countermeasure, Light Weight Block Cipher

I. 서 론

이론적으로 안전하다고 증명된 암호 알고리즘이라 하더라도 장비에 탑재되어 구동될 때 발생하는 부가적인 정보로 비밀 키를 찾아내는 부채널 분석(Side Channel Analysis, SCA)에 취약점이 존재한다. 이러한 부채널 분석은 1996년 최초로 Paul Kocher에 의해 소개 되었다[1]. 부채널 분석기법 중 일반적으로 사용되는 상관전력분석(Correlation Power Analysis, CPA)은 전력과 추측 값 사이의 상관도를 이용하여 비밀 키를 찾아내는 방법이다[2]. 이러한 기법들의 발전으로 다양한 장비에서 부채널 취약

접수일(2015년 2월 12일), 수정일(2015년 3월 31일), 게재확정일(2015년 3월 31일)
* 본 연구는 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2013R1A1A2A10062137)
† 주저자, zasingam@kookmin.ac.kr
‡ 교신저자, christa@kookmin.ac.kr(Corresponding author)

점이 확인되고 있다.

한편 사물인터넷(Internet of Things, IoT)환경에서는 다양한 장비에서 인터넷 통신이 가능하게 한다. 그러므로 다양한 통신환경에서 보안요소와 성능을 충족시키기 위한 경량 블록 암호 알고리즘이 필요하다. 환경에 알맞은 알고리즘의 필요성으로 인하여 다양한 경량 블록 암호 알고리즘이 제안되고 있다. 이론적인 안정성이 증명된 국산 블록 암호 알고리즘으로는 SEED, ARIA, HIGHT, LEA가 있다. 이러한 블록 암호 알고리즘에도 부채널 분석에 취약점이 존재한다. 그럼에도 불구하고 현재에는 심각하게 고려되지 않고 있는 실정이다.

본 논문에서는 HIGHT 블록 암호 알고리즘[3,9]의 구조를 활용한 효율적인 1차 전력 분석 방법을 제안한다. 제안한 방법에 실험적인 검증을 위해 8비트 기반 ATmega128 보드에서 동작하는 HIGHT 알고리즘을 대상으로 1차 전력 분석을 수행하였다.

그리고 1차 전력분석에 안전한 HIGHT 암호 알고리즘 대응기법을 제안한다. 제안한 1차 부채널 대응기법의 효율성 비교를 위해 1차 부채널 대응기법이 적용된 AES를 활용하였다[4]. 속도 측정은 암호알고리즘을 ATmega128 보드에 탑재하여 실제 구동되는 시간을 측정하였다. 부채널 대응기법 적용하였을 때 속도의 경우 AES는 3.24배 정도의 추가 연산량이 필요하며, HIGHT는 55.88배 이상의 추가 연산량이 필요하였다. 한편 부채널 대응기법의 적용을 위하여 각 알고리즘의 RAM 소모량의 경우 AES는 약 266바이트가 필요하고 HIGHT는 약 23바이트가 필요하다. 이 결과를 통하여 속도가 우선되는 환경인지 메모리가 우선되는 환경인지에 따라 알고리즘을 선택할 수 있다.

본 논문의 구성은 2장에서 HIGHT 알고리즘 등 배경지식에 관해 서술하고 3장은 HIGHT 알고리즘의 1차 전력분석 방법을 제안한다. 4장은 HIGHT 1차 부채널 대응기법을 적용한 구조를 제안하고 AES와 성능을 비교한다. 5장에서 부채널 대응기법 적용에 있어 알고리즘 선택에서 고려되어야 할 사항을 서술한다.

II. 선행 연구

2.1 HIGHT 알고리즘

CHES 2006에 발표된 HIGHT(HIGh security and light weigHT) 알고리즘[3]은 적은 자원을 갖는 장비에 알맞게 설계되었다. HIGHT는 64비트의 블록 크기를 가지며 128비트의 비밀 키를 가진다. HIGHT에 대한 설명에 이해를 돕기 위하여 다음과 같은 표기법을 사용한다.

p_i : 평문에 상위 $i(0 \le i \le 7)$번째 바이트
s_i : 상태문에 상위 $i(0 \le i \le 7)$번째 바이트
s'_i : 연산 후 상태문에 상위 $i(0 \le i \le 7)$번째 바이트
c_i : 암호문에 상위 $i(0 \le i \le 7)$번째 바이트
k_i : 비밀 키에 상위 $i(0 \le i \le 15)$번째 바이트
WK_i : 초기변환과 최종변환에 쓰이는 화이트 키의 상위 $i(0 \le i \le 7)$번째 바이트
$RK_{i,j}$: $i(0 \le i \le 31)$번째 라운드 키의 상위 $j(0 \le j \le 4)$번째 바이트
$\oplus$: XOR(eXclusive-OR) 연산
$+$: 8비트 더하기 연산
$\lll i$: 왼쪽으로 i비트 로테이션 연산

2.1.1 키 스케줄

HIGHT 키 스케줄은 각 바이트 별로 키가 유도되는 특징을 갖는다. 화이트 키는 다음과 같은 식으로 표현된다.

$$WK_i = \begin{cases} k_{i+12} & 0 \le i \le 3 \\ k_{i-4} & 4 \le i \le 7 \end{cases}$$

라운드 키 유도를 위해 다음 알고리즘을 따른다.

Table 1. HIGHT Key Generation Algorithm

Input	$k_i,\ (0 \le i \le 15)$
Output	$RK_{i,j},\ (0 \le i \le 31), (0 \le j \le 4)$
1. For $i=0$ to 7 do 2. For $j=0$ to 7 do 3. $RK_{4i+(j/4),(j\%4)} \leftarrow k_{j-i \bmod 8} + \delta_{16i+j}$ 4. $RK_{4i+(j/4)+2,(j\%4)} \leftarrow k_{(j-i \bmod 8)+8} + \delta_{16i+j+8}$ 5. Return $RK_{i,j}$	

Table 1.에서 보듯이 각 라운드 키의 바이트는 특정 위치의 비밀 키 바이트에 의해서 유도된다. 특히 마지막 라운드 키 $RK_{31,0}$과 $RK_{31,2}$는 각각 비밀 키 k_{13}, k_{15}에서 유도된다. 따라서 $RK_{31,0}$과 $RK_{31,2}$를 찾는 것은 비밀 키 k_{13}, k_{15}를 찾는 것과 같다.

2.1.2 암호화 과정

HIGHT는 암호화 과정에 연산이 덧셈, 로테이션, XOR 연산으로 구성된 ARX 구조이다. 전체구조는 초기변환, 32번의 라운드 과정, 최종변환과 같이 3단계로 구성된다.

2.1.2.1 초기변환

HIGHT 전체 구조에서 초기변환은 다음과 같은 식으로 표현된다.

$$s_0 = p_0 + WK_0,\ s_2 = p_2 \oplus WK_1$$
$$s_4 = p_4 + WK_2,\ s_6 = p_6 \oplus WK_3$$
$$s_i = p_i,\ i = 1,3,5,7$$

2.1.2.2 라운드 과정

HIGHT 라운드 함수는 F_0, F_1가 있으며 다음과 같은 식으로 표현된다.

$$F_0(s_i) = (s_i \lll 1) \oplus (s_i \lll 2) \oplus (s_i \lll 7), i = 2,6$$
$$F_1(s_i) = (s_i \lll 3) \oplus (s_i \lll 4) \oplus (s_i \lll 6), i = 0,4$$

라운드 함수의 다른 연산은 8비트 덧셈연산과 비트단위 XOR 연산으로 이루어져 있으며 라운드의 전체 구조는 Fig. 1.과 같다.

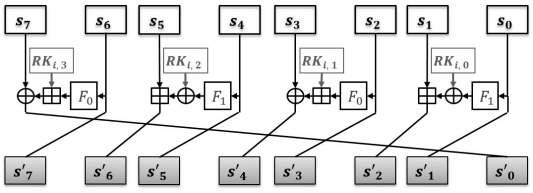


Fig. 1. The Structure of Round in HIGHT

2.1.2.3 최종변환

32번의 라운드 과정을 거친 상태문은 화이트 키를 이용한 최종변환 과정을 가지며 암호문이 생성된다. 이는 다음과 같은 식으로 표현된다.

$$c_0 = s_0 + WK_4,\ c_2 = s_2 \oplus WK_5$$
$$c_4 = s_4 + WK_6,\ c_6 = s_6 \oplus WK_7$$
$$c_i = s_i,\ i = 1,3,5,7$$

위에서 설명한 HIGHT 경량 블록 암호 알고리즘의 전체 구조는 Fig. 2.와 같다.

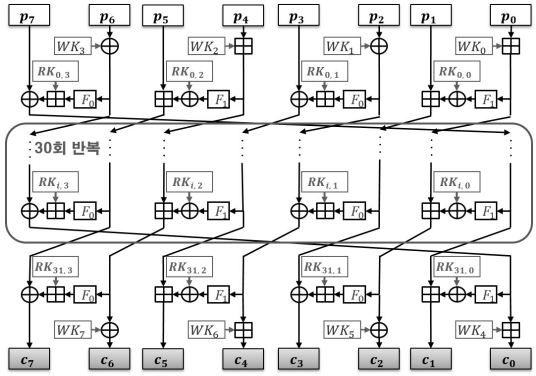


Fig. 2. The Structure of HIGHT

2.2 상관전력분석(Correlation Power Analysis)

상관전력분석[2]은 암호 알고리즘이 수행되는 동안 소비되는 전력이 중간 연산 결과 값에 의존한다는 것을 이용하여 비밀 키를 분석하는 방법으로 부채널 분석 중에 대표적으로 사용되는 방법이다. 상관전력 분석은 다음과 같이 5단계로 나눌 수 있다.

1단계는 알려진 값 d와 민감한 값의 일부인 k를 선택하여 $f(d,k)$를 택한다. 즉, d는 평문 또는 암호문이 될 수 있고 $f(d,k)$는 $d+k$, $d \oplus k$, $S-box(d+k)$등이 될 수 있다.

2단계는 랜덤한 데이터 $d_1,\cdots,d_D$ 를 이용하여 암호 알고리즘을 구동시키고 이의 D개의 소비전력을 측정한다. i번째 암호 알고리즘을 구동시킬 때 측정된 전력 소비량이 $t_i = (t_{i,1},\cdots,t_{i,T})$라 하자.(T:파형의 길이) 그러면 파형은 $D \times T$의 길이를 가진다.

3단계는 예상 가능한 중간값을 모든 키 후보군에 대하여 계산한다. 가능한 모든 키 후보군은 $k_1,\cdots,k_K$ (K:가능한 키 개수)라 하면 다음과 같이 데이터에 대한 중간값을 계산할 수 있다.

$$v_{i,j} = f(d_i,k_j),\ ((1 \le i \le D,\ 1 \le j \le K)$$

4단계는 계산된 중간값에 대응하는 전력 소비 값을 예측한다. 소프트웨어로 구현된 경우 비트가 '1'인 개수에 따라 전력 소비가 증가하는 모델인 해밍 웨이트 모델에 따라 예측한다.

5단계는 예측된 중간값에 대응하는 예상 전력 소비 값과 실제 전력 소비 값의 피어슨 상관계수를 길

이 T개만큼 계산한다.

여기서 옳은 키를 추측하였다면 높은 상관계수가 계산되어 질 것이다. 이러한 이론을 바탕으로 키를 찾아내는 것이 상관전력분석이다.

2.3 마스킹 대응 기법

이러한 상관전력분석에 대한 대응기법으로 마스킹 기법이 있다. 마스킹 대응기법은 암호연산의 중간값에 랜덤한 마스킹 값이 추가로 연산되어 있도록 하여 중간값을 바르게 추측할 수 없도록 만드는 기법이다.

AES 알고리즘은 [4]을 참조하여 1차 부울린 마스킹 대응기법을 적용하였다. 부울린 마스킹 대응기법에서 선형 연산인 경우 마스킹 값의 관리가 간단하지만, 비선형 연산의 경우 연산에 따라 마스킹 값의 관리가 다르게 고려되어야 한다.

2.3.1 선형연산에서의 마스킹 값의 관리

중간에 연산되는 값이 s라 하고 이 값에 랜덤한 마스킹 값 m이 XOR 연산 되어 있다고 하면 중간에 연산 되는 값은 $s \oplus m$가 된다. 이때 암호 알고리즘의 선형연산을 l라 하면 다음식과 같이 연산이 분리되어 진다.

$$l(s \oplus m) = l(s) \oplus l(m)$$

이때, $l(m) = m'$이라 하면 m'을 새로운 마스킹 값으로 관리하면 된다.

2.3.2 비선형연산에서의 마스킹 값의 관리

대표적인 비선형연산은 $S-box$연산과 덧셈연산이 있다. $S-box$연산의 경우 체 위에서의 연산을 이용하여 직접계산이 가능하지만, 이는 연산 속도가 현저히 떨어지므로 일반적으로 테이블 기반의 참조 연산을 하게 된다. 이와 같은 연산 형태에 부울린 마스킹 기법이 적용될 경우, 연산의 효율을 고려하여 암호 알고리즘이 매번 동작할 때마다 마스킹 테이블을 생성하여 $MS-box$연산을 수행한다. Table 2.은 2^n 크기의 $MS-box$를 생성하는 알고리즘이다.

부울린 마스킹이 적용되었을 때 덧셈연산을 바로 수행하면 마스킹 값을 관리하기 어렵다. 따라서 안전한 덧셈연산을 수행하기 위해 부울린 마스킹을 산술

Table 2. $MS-box$ Generation Algorithm

Input	Input Masking : IM Output Masking : OM
Output	$MS-box$
1. For $i=0$ to 2^n-1 do 2. $MS-box[i \oplus IM] = S-box \oplus OM$ 3. Return $MS-box$	

마스킹으로 변환(Boolean to Arithmetic Conversion, BtoA)하여 덧셈연산을 한다. 덧셈연산 후에는 산술 마스킹을 부울린 마스킹으로 변환(Arithmetic to Boolean Conversion, AtoB)하여 나머지 연산을 수행한다. 본 논문에서는 BtoA 알고리즘은 [5]를 참조하여 적용하였고 AtoB 알고리즘은 [6]을 참조하였다.

III. HIGHT에 대한 1차 전력 분석 방법들

3장에서는 HIGHT 알고리즘에 대해 평문을 이용하였을 경우 평문과 암호문을 이용할 경우를 나눠서 분석 시나리오를 제시한다. 그리고 실제 알고리즘을 보드에 탑재하여 바이트 단위의 분석을 시행한다. 이를 근거로 부채널 분석에 취약점을 확인한다.

3.1 공격 시나리오

3.1.1 평문을 이용한 1차 부채널 분석

평문을 이용한 HIGHT 부채널 분석은 2개의 선형연산을 분석해야 비밀 키를 복구하는 분석 시나리오를 가진다.

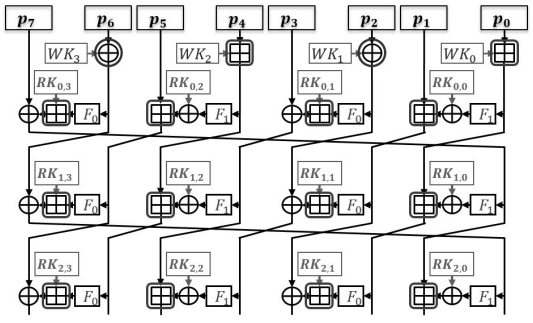


Fig. 3. The Attack Point of CPA

Fig. 3.에서 원으로 표시된 지점은 선형 연산 분

석지점이고 사각형으로 표시된 지점은 비선형 연산 분석 지점이다. 부채널 분석은 일반적으로 선형 연산 부분에서 보다 비선형 연산 부분에서 더 분석성능이 좋으므로 비선형 연산을 부채널 분석지점으로 선정하였다. 하지만 WK_1와 WK_3를 반드시 선형연산으로 분석하여야 한다. 이로 인해 WK_1와 WK_3를 분석 시 후보키가 발생하게 된다. WK_1의 경우 후보키 중에 옳은 키를 찾기 위해서는 $RK_{0,1}$의 분석을 활용한다. 활용방법은 모든 후보키에 대해 $RK_{0,1}$의 분석을 하여 구분되는 높은 상관 계수 값을 가지는 값의 WK_1의 후보키와 $RK_{0,1}$의 추측키의 쌍으로 결정할 수 있다. 비슷하게 WK_3를 비롯한 다른 키에 대해서 후보키를 줄이는 과정을 거치며 부채널 분석을 한다.

3.1.2 평문, 암호문을 이용한 1차 부채널 분석

평문, 암호문을 모두 이용할 수 있는 분석 환경의 경우 비선형연산이 연산된 부분으로 부채널 분석 지점을 선정할 수 있다. Fig. 3.에서 WK_1와 WK_3의 분석을 제외하면 비선형연산이 연산된 부분을 분석지점으로 선정할 수 있다. 여기서 WK_1와 WK_3은 비밀키 k_{13}과 k_{15}를 복구하는 것과 같다. 따라서 우리는 비밀 키 k_{13}과 k_{15}바이트를 암호문의 비선형연산을 분석하는 것으로 분석 성능을 높이고자 한다.

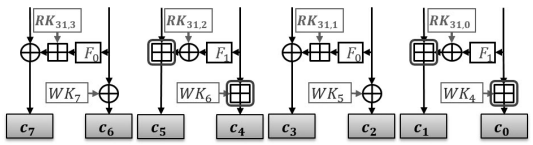


Fig. 4. Analysis Using Cipher Text

Fig. 4.에서 먼저 $WK_4(=k_0)$와 $WK_6(=k_2)$의 비선형연산의 분석을 시행한 후 $RK_{31,0}$과 $RK_{31,2}$를 사각형으로 표시한 부분에서 비선형연산으로 분석할 수 있다. 한편 $RK_{31,0}$과 $RK_{31,2}$는 k_{13}, k_{15}를 분석하는 것과 같다. 따라서 WK_1와 WK_3를 암호문의 비선형 연산으로 분석할 수 있다.

암호문과 평문을 이용하는 부채널 분석은 비선형연산의 분석만으로 비밀 키를 분석할 수 있다. 하지만 평문만을 이용하여 분석하는 방법은 선형 연산을 반드시 공격하여야 한다. 따라서 암호문과 평문을 모두 이용한다면 부채널 분석 성능을 높일 수 있다.

3.2 HIGHT 1차 부채널 분석 실험결과

HIGHT 알고리즘의 취약성을 검증하기 위한 실험을 시행하였다. 파형 수집 환경은 ATmega128 보드를 사용하여 8비트로 구현된 HIGHT 알고리즘을 구동하였다. 그리고 Teledyne LeCroy사 HDO6103 오실로스코프에서 500MS/s 샘플링으로 수집하였다. 수집한 파형을 연산 단위로 분석하여 공격 성공률(Success Rate)[7]과 추측 엔트로피(Guessing Entropy)[7]를 계산하고 비교한다. 공격 성공률과 추측 엔트로피는 모두 1,000번 분석하였을 때, 결과 값으로 나타낸다. 여기서 공격 성공률은 1,000번 분석하였을 때 옳은 키를 찾을 확률이고 추측 엔트로피는 1,000번 분석하였을 때 옳은 키의 평균 순위이다. 이에 대한 실험 결과 해석은 만약 분석성능이 우수하다면 공격 성공률이 100%에 가까워질 것이고 추측 엔트로피는 1에 가까워질 것이다. 다만 선형연산의 경우 후보키가 항상 존재할 수 있어서 추측 엔트로피는 2에 가까워진다.

바이트 키를 분석하기 위해서 같은 추측이라면 확산(diffusion)이나 혼돈(confusion)이 더 일어났을 경우 부채널 분석 성능이 좋다. 그러므로 WK_0를 분석하기 위해서는 $F_1(p_0+guessing\,key)$부분을 추측하였고 WK_1에 분석을 위해 $F_0(p_2\oplus guessing\,key)$ 부분을 추측하였다.

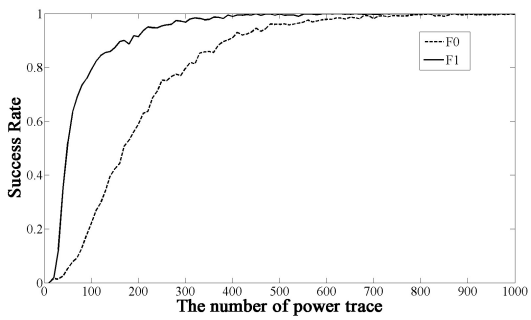


Fig. 5. Success Rate of CPA for HIGHT

Fig. 5.는 선형함수의 조합인 F_0부분의 분석과 비선형함수의 조합인 F_1부분의 분석의 추측 엔트로피를 계산하였다. 그 결과 공격 성공률이 99% 이상이기 위해서 필요한 파형 수는 F_0의 경우 730개 이상이고 F_1의 경우 400개 이상인 것을 확인하였다.

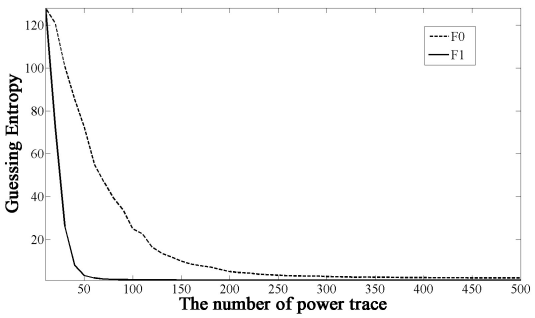


Fig. 6. Guessing Entropy of CPA for HIGHT

Fig. 6.은 마찬가지로 F_0부분과 F_1부분의 분석을 통하여 추측 복잡도를 계산하였다. 그 결과 추측 복잡도가 3 이하이기 위해서는 F_0의 경우 270개 이상의 파형이 필요하고 F_1의 경우 50개 이상의 파형이면 충분한 것을 확인할 수 있었다.

위의 실험결과를 통하여 선형연산들이 조합된 연산 부분의 분석보다 비선형연산과 조합된 연산 부분의 분석 성능이 뛰어난 것을 확인할 수 있다. 따라서 선형 연산을 분석해야 하는 평문만 이용한 분석보다 평문, 암호문을 이용한다면 앞에 설명과 같이 비선형연산 부분만 공격할 수 있으므로 더 좋은 분석성능을 가진다고 할 수 있다.

IV. 1차 부채널 분석에 안전한 HIGHT 마스킹 대응기법

3장에서 HIGHT 국산 경량 블록암호 알고리즘에서 부채널 분석 취약점을 검증하였다. 이에 따라 장비에 탑재될 HIGHT 알고리즘은 부채널 대응기법이 고려돼야 한다. 따라서 이번 장에서는 알려진 부울린 마스킹 대응기법을 HIGHT 알고리즘에 적용하고 그 구조에 대하여 서술한다. 또한, 적용결과를 8비트로 구현된 AES 알고리즘과 비교한다.

4.1 연산별 대응기법

HIGHT 알고리즘은 ARX 구조이다. 따라서 덧셈, 로테이션과 XOR 연산으로 구성되어 있다. 우리는 부채널 대응기법인 1차 부울린 마스킹을 적용할 것이다. 부울린 마스킹을 적용하기 위해서는 선형연산인 로테이션과 XOR 연산에서는 마스킹 값이 변화함에 따라 그 값을 저장하는 것으로 간단히 마스킹 값의 관리가 가능하다. 하지만 비선형연산인 덧셈연산에서는 다음과 같이 덧셈연산에 대한 마스킹 관리가 필요하다.

4.1.1 덧셈연산

1차 부울린 마스킹에서 덧셈연산을 하기 위해서는 BtoA 변환을 하여 안전한 덧셈연산을 한다. 덧셈연산을 한 이후에 다시 AtoB 변환을 시키는 것으로 나머지 연산을 수행한다. HIGHT에서 마스킹 덧셈연산(Masking Addition, MA)을 도식화하면 다음 Fig. 7.와 같다.

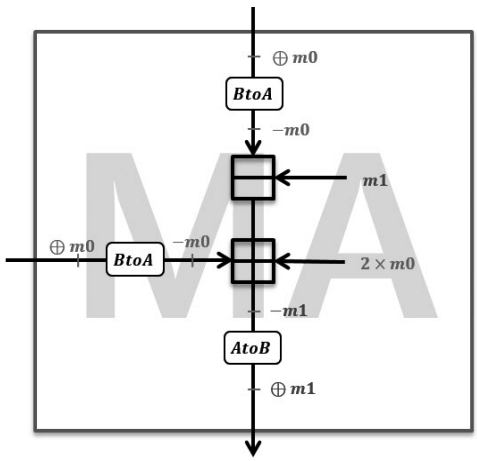


Fig. 7. The Structure Masking Addition

HIGHT의 MA 연산은 Fig. 7.과 같이 두 입력에 같은 8비트 마스킹 값 $m0$이 되도록 구성하였고 MA 연산 후 출력 마스킹 값은 8비트 $m1$이 되도록 구성하였다. Fig. 7.에서 BtoA 알고리즘은 [5]를 참조하였고 AtoB 알고리즘은 [6]을 참조하여 구성하였다.

4.2 HIGHT 마스킹 구조

HIGHT 마스킹 구조는 사전 연산 단계와 마스킹 적용 단계와 마스킹 라운드 과정 단계로 나누어 설명한다. 기본개념은 8비트 랜덤 값을 XOR 연산하여 마스킹을 적용하였다. 또한, 각 라운드가 시작할 때와 끝날 때 같은 마스킹 값을 가지도록 구성하여 라운드 함수를 반복적으로 사용 가능하게 하였다.

라운드 키에는 각 바이트에 랜덤 값 $m0$을 XOR 연산하여 사용한다.

4.2.1 사전 연산 단계

사전연산단계에서 8비트 랜덤한 값 $m0$이 생성되고 선형연산에서 마스킹 값 관리를 위해 $m1 = F_0[m0]$, $m3 = F_1[m0]$ 값을 저장한다. 또한, 중간에 마스킹 값의 변환을 위해 $m2 = m0 \oplus m1$을 저장한다.

4.2.2 마스킹 적용 단계

최초에는 평문 p_2, p_6을 제외한 각 바이트와 모든 라운드 키 각 바이트에 랜덤한 8비트 값 $m0$을 XOR 하고 암호화 과정이 시작된다. 평문 2바이트 p_2, p_6에 마스킹을 적용하지 않는 이유는 초기변환과정에서 각각 WK_1, WK_3가 XOR연산 되면서 자연스럽게 마스킹이 적용되기 때문이다. 또한 p_0, p_4는 화이트 키 덧셈 후 마스킹 변환 연산이 추가된다. Fig. 8.에서 보면 초기변환 후에 모든 바이트를 $m0$마스킹 값으로 유지하기 위하여 MA연산이 일어난 바이트는 $m2$를 추가적으로 XOR 연산한다.

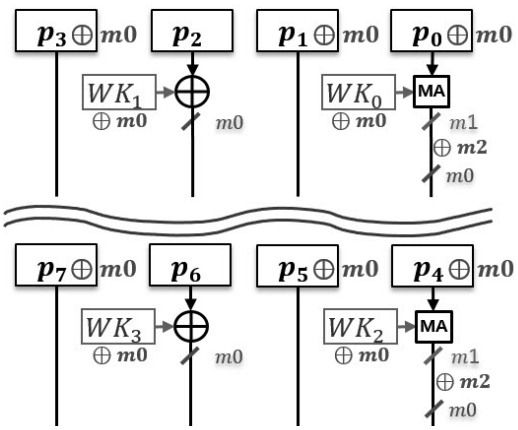


Fig. 8. Masking Initial Conversion

4.2.3 마스킹 라운드 단계

HIGHT 마스킹 라운드는 입력 마스킹과 출력 마스킹이 $m0$로 같게 구성한다. 선형연산 중에는 마스킹 값이 제거되지 않도록 한다. 비선형연산(덧셈)은 입력 마스킹 값을 같게 하여 구성한다. 이 논리를 바탕으로 마스킹 라운드의 구조를 설계한다.

Fig. 9.에서 보면 F_1을 거치며 변형된 마스킹 값 $m3$는 키 XOR 연산 후에 $m3$를 연산하여 MA입력

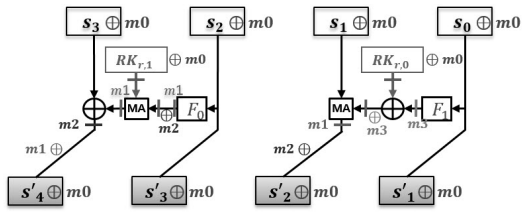


Fig. 9. The Structure of Masking Round

마스킹 값을 $m0$로 유지한다. MA연산 후에는 $m2$를 XOR 연산하여 $m0$로 마스킹 값을 유지한다. F_0을 거쳐 변형된 마스킹 값 $m1$은 $m2$를 XOR 연산하여 MA 입력 마스킹 값을 $m0$로 유지한다. 그 후 s_3와 연산하여 라운드 연산이 끝나면 $m1$을 XOR 연산하여 마스킹 보정 과정을 거친다.

Fig. 9.는 4개 바이트의 라운드 변환을 도식화하였으며 나머지 4개 바이트도 Fig. 9.과 같은 구조를 가진다. 다만 s_7의 결괏값이 s'_0로 이어진다.

4.3 마스킹을 적용한 AES와 HIGHT 성능비교

속도 측정을 위한 기준코드는 AES의 경우 성능개선의 구현을 하였고 HIGHT의 경우 한국인터넷진흥원에서 공개한 8비트 기반으로 구현된 코드[8]를 사용하였다. 속도측정은 Atmega128 보드에 탑재하여 암호 알고리즘이 실제 구동하는 시간을 3.2에서 사용된 오실로스코프를 이용하여 측정하였다.

속도를 측정한 보드 환경은 Table 3.과 같다.

Fig. 10.에서 Normal은 대응기법이 적용되지 않은 일반적인 알고리즘이고 1st-Order는 1차 부채널 대응기법이 적용된 알고리즘이다. AES의 경우 대응기법을 적용하였을 때 3.24배의 성능저하가 나타났고 HIGHT는 55.88배의 성능저하가 나타났다. 즉, 대응기법이 적용되지 않은 알고리즘의 속도를 비교했을 때 HIGHT가 다소 우수한 성능을 보였다. 그러나 1차 부채널 대응기법을 적용하였을 때에는 성능이 역전됨을 확인할 수 있다. 이런 현상이 나타나는 여러 이유 중 하나는 ARX 구조를 갖는 HIGHT 알고리즘에서 비선형연산인 덧셈에 대응기법을 적용할 때

Table 3. Board Specification

CPU	ATmega128
RAM	4KB
Processor	8 bit System

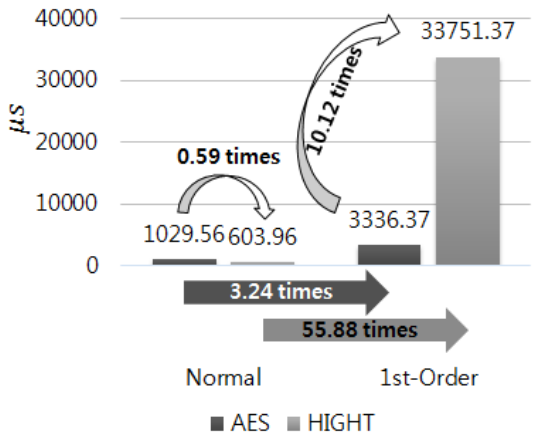


Fig. 10. Comparison Between the Performance of AES and HIGHT

연산의 횟수만큼 성능저하가 일어나는 반면에 AES 비선형 연산인 $S-box$연산은 테이블을 생성하여 성능저하를 최소화하기 때문이다.

한편 부채널 대응기법이 적용된 알고리즘의 RAM 소모량의 경우 AES는 $MS-box$ 테이블(256 바이트)과 마스킹 값을 포함하여 약 266바이트를 소모하고 HIGHT는 AtoB를 위한 보정 테이블(16바이트)과 마스킹 값을 포함하여 약 23바이트가 소모된다.

V. 결 론

본 논문에서는 HIGHT 부채널 취약점을 검증하였다. 그리고 이에 대한 대응기법으로 1차 부울린 마스킹이 적용된 HIGHT 알고리즘을 설계하였다. 또한, 마스킹이 적용된 AES와 HIGHT 알고리즘의 성능을 측정하여 효율성을 측정하였다. 그 결과, ARX 구조로 구성된 경량블록암호의 경우 SPN 구조의 블록암호보다 부채널 대응기법을 적용하였을 때, 연산의 부하가 더 발생하는 것을 확인할 수 있었다. 이는 덧셈연산에 대해 부채널 대응기법을 적용하였을 때 연산의 부하가 많이 발생하기 때문이다. 하지만 메모리가 부족한 환경의 경우에는 ARX구조를 선택하는 것이 $S-box$연산을 하는 SPN구조를 선택하는 것이 유리하다.

한편 사물인터넷 환경이 다가옴에 따라 각 환경에 맞는 다양한 암호 알고리즘이 제안되거나 기존 알고리즘이 사용될 것이다. 이에 따라 사용되는 알고리즘들의 부채널 대응기법을 고려하지 않을 수 없다. 따라서 앞으로 블록암호 알고리즘을 설계할 때에도 이러한 부채널 대응기법의 적용했을 경우 효율성에 대한 고려가 필요하다.

References

[1] P. Kocher, J. Jaffe and B.Jun, "Differential power analysis," Advances In Cryptology, CRYPTO′ 99, LNCS 1666, pp. 388-397, 999.

[2] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," Cryptographic Hardware and Embedded Systems, LNCS 3156, pp. 16-29, 2004.

[3] Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bonseok 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," Cryptographic Hardware and Embedded Systems, LNCS 4249, pp. 46-59, 2006.

[4] C. Herbst , E. Oswald and S. Mangard, "An AES Smart Card Implementation Resistant to Power Analysis Attacks," Applied Cryptography and Network Security, LNCS 3989, pp. 239-252, 2006

[5] L. Goubin, "A Sound Method for Switching between Boolean and Arithmetic Masking," Cryptographic Hardware and Embedded Systems, LNCS 2162, pp. 3-15, 2001.

[6] J.S. Coron and A. Tchulkine, "A New Algorithm for Switching from Arithmetic to Boolean Masking," Cryptographic Hardware and Embedded Systems, LNCS 2779, pp. 89-97, 2003.

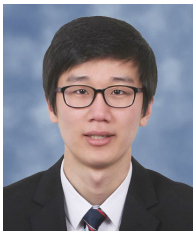
[7] F-X. Standaert, TG. Malkin, and M. Yung, "A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks," EUROCRYPT, LNCS 5479, pp. 443-461, 2009.

[8] KISA website, http://seed.kisa.or.kr/iwt/ko/sup/EgovHightInfo.do

〈저자소개〉

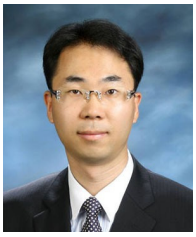
김 태 종 (Tae-jong Kim) 학생회원
2010년 2월: 국민대학교 수학과 학사
2014년 2월~현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 부채널 분석, 대칭키 암호알고리즘, 해쉬 함수

원 유 승 (Yoo-seung Won) 학생회원
2012년 2월: 국민대학교 수학과 학사
2014년 2월: 국민대학교 수학과 석사
2014년 2월~현재: 국민대학교 금융정보보안학과 박사과정
〈관심분야〉 정보보호, 부채널 분석, 스마트 카드 보안



박 진 학 (Jin-hak Park) 학생회원
2013년 2월: 국민대학교 수학과 학사
2014년 2월~현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 정보보호, 부채널 분석, 대칭키 암호 알고리즘

안 현 진 (Hyun-jin An) 학생회원
2012년 2월: 국민대학교 수학과 학사
2014년 2월: 국민대학교 수학과 석사
2014년 2월~현재: 국민대학교 금융정보보안학과 박사과정
〈관심분야〉 정보보호, 부채널 분석, RFID 정보보호기술



한 동 국 (Dong-guk Han) 종신회원
1999년 2월: 고려대학교 수학과 학사
2002년 2월: 고려대학교 수학과 석사
2005년 2월: 고려대학교 정보보호대학원 공학박사
2004년 4월~2005년 4월: 일본 Kyushu Univ. 방문연구원
2005년 4월~2006년 4월: 일본 Future Univ. -Hakodate, Post Doc.
2006년 6월~2009년 2월: 한국전자통신연구원 정보보호연구본부 선임연구원
2009년 3월~현재: 국민대학교 수학과 부교수
2012년 3월~현재: 국민대학교 일반대학원 금융정보보안학과 부교수
〈관심분야〉 공개키 암호 시스템 안전성 분석 및 고속 구현, 부채널 분석, RFID/USN 정보보호기술