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.2015.25.2.449

LEA에 대한 부채널 분석 및 대응 방법*

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

Side channel Attacks on LEA and Its Countermeasures*

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

요 약

최근 사물 인터넷에 대한 정보보호가 이슈화되면서 이에 적합한 알고리즘에 대한 연구가 활발히 진행되고 있다. 국내에서도 IoT환경에 적합한 경량 대칭키 암호 알고리즘인 LEA(Lightweight Encryption Algorithm)를 개발하였다. 본 논문에서는 LEA 암호에 대한 1차 전력 분석 방법들을 소개하고 이를 실험적으로 검증하였다. 그리고 1차 전력 분석에 안전하도록 LEA를 설계하는 방법을 제안한다. 설계된 LEA 부채널 분석대응법의 효율성을 비교하기 위해 동일한 안전도를 제공하는 AES 부채널 대응법과 비교하였다.

ABSTRACT

Recently, information security of IoT(Internet of Things) have been increasing to interest and many research groups have been studying for cryptographic algorithms, which are suitable for IoT environment. LEA(Lightweight Encryption Algorithm) developed by NSRI(National Security Research Institute) is commensurate with IoT. In this paper, we propose two first-order Correlation Power Analysis(CPA) attacks for LEA and experimentally demonstrate our attacks. Additionally, we suggest the mask countermeasure for LEA defeating our attacks. In order to estimate efficiency for the masked LEA, its operation cost is compared to operation time of masked AES.

Keywords: Side Channel Analysis, LEA, Masking Countermeasure

I. 서 론

IT 환경의 발달로 인해 사물인터넷(Internet of Things, IoT)의 보안이 이슈화되면서 이에 적합한 암호 알고리즘에 대한 연구가 활발히 진행되고 있다. 또한 이러한 환경에서 메모리, CPU 성능, 전력 등이 고려되어 최적화시키는 연구가 활발히 진행되고 있다. 따라서 기존에 사용되던 AES[1], ARIA[2], SEED[3] 등을 이용하는 것은 한계가 있다. 따라서 국내에서는 고속, 경량, 저전력을 요구하는 암호시스템을 필요로 하게 되었고 이에 따라 32비트 플랫폼에서 효율적으로 구동되는 LEA(Lightweight Encryption Algorithm)알고리즘[4]을 제안하였다.

LEA는 2013년 12월 한국정보통신기술협회(TTA)의 표준으로 지정되었으며, 128비트 블록 단위로 암·복호화를 수행하고 128, 192, 256비트의 비밀 키를 사용할 수 있으며 라운드함수는 Addition, Rotation, XOR의 연산만으로 구성되어있다. 따라

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, painstars@kookmin.ac.kr
‡ 교신저자, christa@kookmin.ac.kr(Corresponding author)

서 연산 속도가 다른 블록 암호에 비해 빠르고 키스케줄 과정이 간단하며 S-box를 사용하지 않으므로 경량 대칭키 암호를 필요로 하는 환경에서 적합하다.

부채널 분석(Side Channel Analysis)은 장비에서 암호알고리즘이 구동될 때 발생하는 전력신호, 전자파, 소리 등의 부가적인 정보를 이용하는 공격 방법이다. 대표적인 부채널 분석 방법은 전력 분석(Power Analysis), 시차 공격(Timing Attack), 오류 주입 공격(Fault Attack) 등이 있다. 따라서 이러한 부채널 분석에 대한 다양한 방법들이 존재하며, 이를 통해 알고리즘 내에 비밀 키를 추출할 수 있다. 이와 같은 부채널 분석에 대응하기 위하여 경량 대칭키 암호 알고리즘을 분석에 안전하기 위한 대응기법을 고려한 설계가 필요하다. 다시 말하면, 부채널 대응기법이 적용된 경량 대칭키 암호 알고리즘은 모듈 속도, 면적, 전력 소비 등과 같은 최초 설계 철학에 반하는 결과를 도출할 수 있다. 따라서 LEA의 부채널 분석 및 대응기법을 고려할 것이다.

본 논문에서는 LEA 암호 알고리즘의 구조를 활용한 효율적인 1차 전력 분석 방법을 제안한다. 제안한 방법을 실험적으로 검증하기 위해 32비트 기반 ARM 프로세서가 탑재된 Smart card에서 동작하는 LEA-128을 대상으로 1차 전력 분석을 수행하였다.

그리고 1차 전력분석에 안전한 LEA 암호 알고리즘 대응기법을 제안한다. 제안한 LEA 부채널 대응법의 효율성을 비교를 위해 1차 부채널 대응기법이 적용된 AES를 활용하였다. 예를 들어, AES-32비트 기반 구현에서는 부채널 대응법 적용으로 2배 정도의 추가 연산량이 필요했으며, LEA-32비트는 17배 이상이 소모되었다. 또한 일반적인 AES의 경우 S-box 테이블에 따른 ROM 256바이트의 메모리가 소요되고 대응기법이 적용된 경우 추가적으로 Masked S-box 테이블 생성에 필요한 RAM 256바이트와 마스크값 RAM 6바이트의 메모리가 소요된다. 반면 LEA의 경우 일반적인 경우 추가적인 ROM 소모량은 없고 대응기법이 적용된 경우 추가적으로 산술 마스킹에서 부울린 마스킹으로 변환에 필요한 테이블 생성에 필요한 RAM 32바이트와 마스크값 RAM 마스크값 RAM 6바이트의 메모리가 소요된다.

본 논문의 2장에서는 LEA 알고리즘의 서술과 부채널 분석에 안전한 대응기법 논리에 대해 간략히 서술하고 3장은 LEA에 대한 1차 부채널 분석을 서술한다. 4장은 LEA 1차 대응기법을 제안함으로써 안전한 LEA 설계 방법을 제시하고 마지막으로 5장에서 결론을 맺는다.

II. 선행 연구

2.1 LEA 알고리즘

ARX(modular Addition, bitwise Rotation, bitwise XOR) 구조로 이루어진 LEA 알고리즘은 128비트 블록 사이즈를 가지며, 3가지 키 사이즈(128, 192, 256)로 구현 가능하다. Table 1.은 키 사이즈에 따른 LEA의 설명이다.

Table 1. Specification of LEA

LEA-k	Size of block	Key length	Number of rounds
LEA-128	128	128	24
LEA-192	128	192	28
LEA-256	128	256	32

2.1.1 기호와 표기

본 논문에서 사용하는 기호는 아래와 같다

p_i : 128비트 평문에서 i번째($i=0,1,2,3$) 한 워드 (32비트)

c_i : 128비트 암호문에서 i번째($i=0,1,2,3$) 한 워드 (32비트)

k_i : 128비트 비밀 키에서 i번째($i=0,1,2,3$) 한 워드 (32비트)

$RK_r[i]$: r번째 라운드 키에서 i번째($0 \le i \le 5$) 한 워드 (32비트)

$T_i[j]$: i번째($0 \le i \le 24$)라운드 키를 유도하기 위한 임시 변수 한 워드(32비트)($0 \le j \le 3$)

$\oplus$: XOR(eXclusive-OR) 연산

$\boxplus$: 32비트 Addition 연산

$\boxminus$: 32비트 Subtraction 연산

ROL_i : 왼쪽으로 i비트 Rotation 연산($1 \le i \le 31$)

ROR_i : 오른쪽으로 i비트 Rotation 연산($1 \le i \le 31$)

2.1.2 LEA 키스케줄

LEA 암호 알고리즘이 수행되기 위하여 128비트 비밀 키로부터 암호화 과정에 필요한 24개의 라운드 키 $RK_r(0 \le r \le 23)$들을 생성하는 키 스케줄 과정

은 Fig. 1.과 같다.

LEA-128은 $RK_r[1]$, $RK_r[3]$, $RK_r[5]$가 동일한 키를 사용하는 특징이 있다. 또한 $T_{i+1}[j]$를 안다면 $T_i[j]$를 유추할 수 있고 역도 성립한다. 이러한 특징으로 인해 j가 동일한 모든 라운드 키를 유추할 수 있다. 예를 들면 $RK_{23}[0]$를 안다면 $RK_0[0]$를 유추할 수 있다.

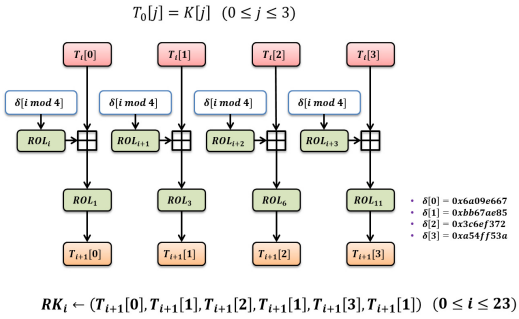


Fig. 1. Keyschedule of LEA

2.1.3 암호화 라운드

LEA-128의 암호화 라운드 과정은 키 스케줄링을 통하여 얻은 192비트 라운드 키를 이용한다. 앞서 언급한 바와 같이 $RK_r[1]$, $RK_r[3]$, $RK_r[5]$라운드 키가 동일하게 사용된다. 평문 128비트 블록을 32비트 4워드로 나누어 Addition, Rotation, XOR 연산 후 다음 라운드의 입력 값이 된다. 암호화 과정을 그림으로 나타내면 Fig. 2.와 같다.

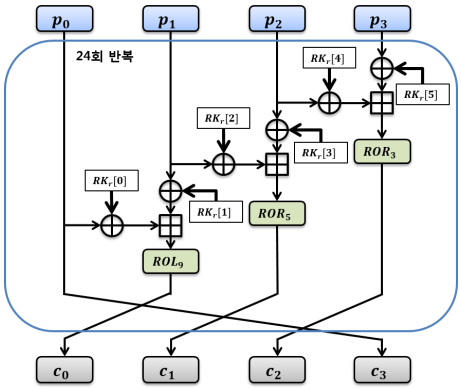


Fig. 2. Encryption rounds of LEA

2.2 1차 부채널 분석에 안전한 마스킹 대응기법

부채널 대응기법은 대표적으로 예상 가능한 중간 값을 랜덤하게 만드는 마스킹 기법이 쓰인다. 경량 대칭키 암호 알고리즘이 사용되는 디바이스에서 요구되는 저면적, 저전력 등을 만족시키기 위해서는 고차 마스킹을 쓰는 것이 현실적이지 않기 때문에 1차 마스킹 기법이 주로 쓰인다.

일반적으로 예상되는 중간 값 v에 대하여 매번 랜덤 값을 생성하여 다음과 같은 방법을 통해 중간 값을 유지한다.

$$v_m = v \perp m$$

v_m은 랜덤 값 m에 의하여 마스킹 기법이 적용된 중간 값이다. $\perp$으로 어떠한 연산을 선택하느냐에 따라 마스킹 기법이 다르게 적용된다. 또한 암호 알고리즘은 선형연산과 비선형 연산으로 구성되며, 각각은 연산 특성으로 인해 서로 다른 마스킹 논리가 적용된다. 선형 연산이 일어나는 부분의 경우 $f(x \triangle m) = f(x) \triangle f(m)$의 형태로 마스킹 값을 관리할 수 있고 대표적으로 부울린 마스킹 기법인 $\oplus$연산이 있다. 이에 비해 비선형 연산의 경우, $f(x \circledcirc m) \neq f(x) \circledcirc f(m)$이기 때문에 소프트웨어 구현 관점에서 비선형 연산의 경우 선형 연산에 비해 사전 연산 테이블을 이용하거나 추가적인 연산이 필요하므로 속도나 메모리 소모가 많아질 수 있다. 대표적으로 산술 마스킹 기법인 $GF(2^n)$ 위에서 $\boxplus$연산이 있다.

III. LEA에 대한 1차 전력 분석 방법들

본 장에서는 LEA에 대한 전력분석에 대한 이론적인 공격 효율성을 비교하고 2가지 시나리오를 소개한다. 또한, 가장 효율적인 시나리오의 실제 실험결과를 보인다.

3.1 LEA 1차 부채널 분석 효율성 비교

우선 LEA-128의 이론적인 분석지점은 크게 $\oplus$연산과 $\boxplus$연산으로 구분된다. 두 연산의 분석 효율성을 비교하기 위해 Relative distinguishing margin[5]인 아래와 같은 식을 이용한다.(추측 키: $\hat{k}$, 옳은 키: k^*)

$$RelMarg(D) = \frac{Corr_{\max}(k^*) - \left\{Corr_{\max}(\hat{k}) \mid \hat{k} \neq k^*\right\}}{\sqrt{Var[Corr(\hat{k})] \mid \hat{k} \in K}} \quad (1)$$

이 때 옳은 키의 최대 상관계수를 $Corr_{\max}(k^*)$이라 하고 옳은 키를 제외한 키 집합 중 최대 상관계수를 $\{Corr_{\max}(\hat{k}) | \hat{k} \neq k^*\}$라 하자. $\oplus$연산과 $\boxplus$연산에서 발생 가능한 비밀 키들에 대한 8비트 시뮬레이션 파형을 만들고 CPA를 수행한 결과, $\oplus$연산의 경우 $RelMarg(\oplus) = 0$이다. 따라서 모든 경우에서 옳은 키의 상관계수의 절대 값과 동일한 틀린 키 1개가 존재한다는 것이다. 반면 +연산의 경우 $RelMarg(\boxplus) = 1.7$이고 옳은 키가 유일하게 분석 가능하다는 것을 의미한다. 그러므로 8비트의 비밀 키를 추측한다면, $\oplus$연산은 2^8의 키 공간을 2로 축소시킬 수 있다. 그러나 $\boxplus$연산의 경우, 유일한 키 분석이 가능하다. 따라서 $\boxplus$연산이 사용되는 부분을 분석지점으로 선택하는 것이 옳은 키를 추측하는데 효율적이다.

3.2 공격 시나리오

3.2.1 평문을 이용한 1차 부채널 분석

평문을 이용한 LEA 1차 부채널 분석은 Fig. 3.과 같이 나타낼 수 있다.

평문을 이용한 1차 부채널 분석의 경우 공격지점으로 활용될 수 있는 부분은 평문 $X_0[i]\{i \in 1,2,3,4\}$와 라운드 키 $RK_0[j]\{j \in 1,2,3,4,5,6\}$가 $\oplus$연산이 일어나는 부분을 선택할 수 있고, 두 가지 키가 사용된 $\boxplus$연산이 일어나는 부분을 선택할 수 있다.

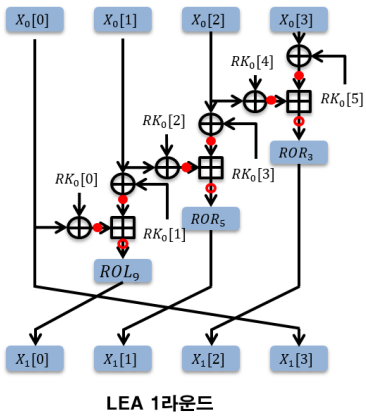


Fig. 3. Side channel analysis for Using plaintext of LEA

따라서 평문을 이용하여 $\boxplus$연산이 사용되는 부분을 8비트씩 추측할 때, 가장 효율적인 방법은 Fig. 3.에서 $\oplus$연산이 일어난 • 지점을 중간 값으로 하여 분석한 뒤 $\boxplus$연산이 일어난 ◦ 지점을 중간 값으로 하여 분석하는 것이다. 이 때, 2^2 • 2^8 키 공간을 추측해야한다.

3.2.2 평/암호문을 이용한 1차 부채널 분석

평/암호문을 이용한 LEA 1차 부채널 분석은 Fig. 4.와 같이 나타낼 수 있다.

LEA의 구조 특성으로 인해 암호문을 이용하여 중간 값 $ROR_9(X_{24}[0]) \boxminus (X_{24}[3] \oplus \hat{k})$을 통해 23라운드 키를 찾을 수 있고 키 스케줄을 역으로 연산하여 해당 워드의 1라운드 키를 찾을 수 있다. 따라서 이 정보를 통해 $(X_0[0] \oplus RK_0[0]) \boxplus (X_0[1] \oplus \hat{k})$을 통해 공통으로 사용된 키를 도출한 후 나머지 키를 분석하는데 활용할 수 있다.

따라서 암호문을 이용할 경우 LEA의 구조적 특성으로 인해 $\boxplus$연산에서 한 가지 키에 대한 2^8의 키 공간을 추측하여 옳은 키를 분석할 수 있다. 그러므로 평문을 이용한 분석의 경우 적은 정보를 이용하여 키를 분석할 수 있다는 장점이 있지만, 추측하는 키 공간이 크기 때문에 소요시간이 많이 걸리는 단점이 있다. 평문과 암호문을 모두 이용할 경우 추측하는 키 공간이 줄어들어 소요시간이 보다 적은 장점이 있지만, 공격자가 보다 많은 정보를 알고 있어야하는 단점을 가지고 있다. 따라서 실제 실험에서는 평문과 암호문을 이용하여 $\boxplus$연산이 사용되는 부분에 대한 분석을 실시하고 이상적인 컴퓨팅 속도를 고려하여 8비트 단위로 분석한다.

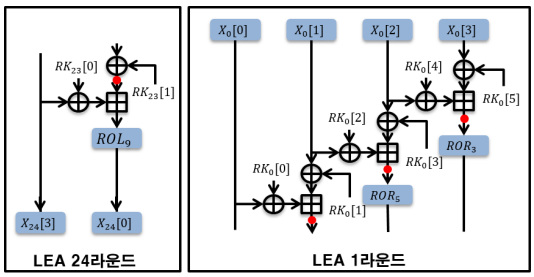


Fig. 4. Side channel analysis of Using plaintext and ciphertext of LEA

3.3 LEA 1차 부채널 분석 실험결과

3.3.1 실험환경

32비트 기반 ARM 프로세서가 탑재된 Smart card에서 동작하는 LEA-128을 대상으로 전체 라운드에 대한 전력파형을 수집하였고 Fig. 5.와 같다. 칩의 특성에 의해 자세한 라운드 구분은 어렵다. X축은 시간(point)을 나타내며 Y축은 해당 시간에 소모된 전력을 나타낸다.

- ◉ 프로세서 : 32비트 ARM
- ◉ 오실로스코프 : Lecroy WaveRunner HDO6104
- ◉ Sampling Rate : 250 Msample/sec
- ◉ 구현 알고리즘 : LEA-128 Encryption
- ◉ 파형수집 개수 : 10,000개

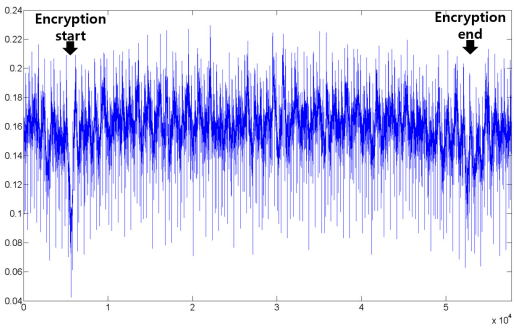


Fig. 5. A measured trace (X-axis : time(point), Y-axis : power consumption)

3.3.2 실험과정

평/암호문을 이용한 1차 부채널 분석을 확인하기 위해, 우선 암호문을 이용하여 $RK_{23}[0]$의 LSB 1바이트를 대상으로 Addition 연산을 수행하는 $ROR_9(X_{24}[0]) \boxminus (X_{24}[3] \oplus \hat{k})$를 중간 값으로 설정하여 분석하였다. 이를 통해 $RK_{23}[0]$를 이용하여 키스케줄을 역으로 연산하여 $RK_0[0]$를 도출하였다. 이후 평문을 이용하여 $(X_0[0] \oplus RK_0[0]) \boxplus (X_0[1] \oplus \hat{k})$를 중간 값을 이용하여 공통으로 쓰이는 키인 $RK_0[1]$의 LSB 1바이트를 도출하였다.

3.3.3 실험결과

옳은 키는 $RK_{23}[0]$의 LSB 1바이트는 0x14, $RK_0[1]$의 LSB 1바이트는 0x87이다. 암호문을 이용하여 $RK_{23}[0]$의 1바이트를 추측한 결과는 Fig. 7.과 같고 $RelMarg(D) = 0.22$이다. 또한 평문을 이용하여 $RK_0[1]$의 1바이트를 추측한 결과는 Fig. 8.과 같고 $RelMarg(D) = 1.69$이다. 시뮬레이션 결과와 다른 이유는 노이즈에 대한 영향과 암호 연산 앞쪽에 정렬을 맞춰 마지막 암호문을 이용하는 부분의 정렬이 제대로 이루어지지 않을 수 있다. 이 때의 CPA 결과는 절대값을 취한 것을 의미하고 X축은 $\hat{k}$를 나타내며 Y축은 추측 키에 대한 최대 상관계수를 나타낸다.

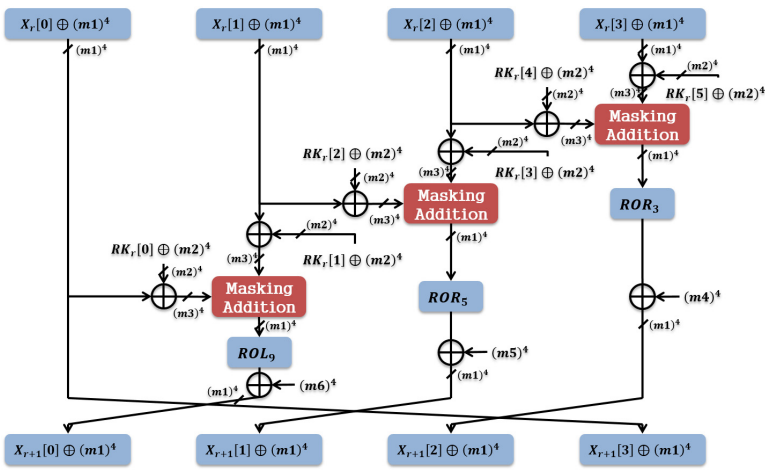


Fig. 6. An encryption round of masked LEA

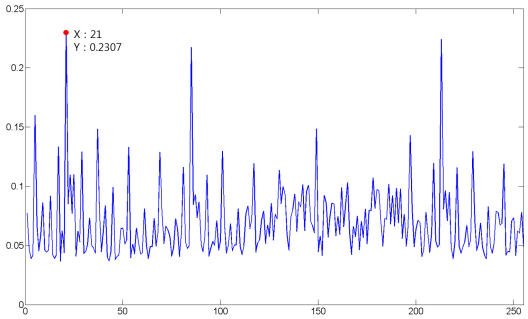


Fig. 7. CPA Result for ciphertext(using 10,000 trace) (X-axis : key candidates, Y-axis : correlation coefficient)

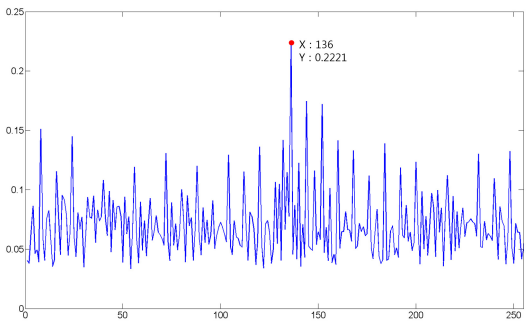


Fig. 8. CPA Result for plaintext(using 10,000 trace) (X-axis : key candidates, Y-axis : correlation coefficient)

IV. LEA 1차 대응기법 제안

본 장에서는 LEA 1차 대응기법을 제안하고 AES와 마스킹 라운드에 따른 속도를 비교하여 설명한다.

4.1 마스킹 라운드 구조

우리가 구현한 LEA에 대한 1차 부울린 마스킹 대응기법 적용방법은 Fig. 6.과 같다.

각 라운드의 입력 마스크는 $m1$이고, 라운드 키 마스크 값으로 쓰이는 값은 $m2$이다. 라운드 내에서 데이터에 대한 마스크 값을 일관성 있게 가져가므로, 첫 라운드에서 마스크 값을 연산해주고 마지막 라운드에서 마스크 값을 제거하는 방법으로 대응기법을 설계하였다. 이 때 쓰이는 마스크 값은 Table 2. 와 같다.

Addition 연산은 비선형 연산에 속하는 연산으로써 1차 부울린 마스킹 기법을 적용한다면, 덧셈 연산을 위하여 $'\oplus' \rightarrow '-'$로 바꾸어 주는 함수인 BtoA[6](Boolean to Arithmetic) 변환 방법과 $'+' \rightarrow '\oplus'$로 바꾸어 주는 함수인 AtoB[7](Arithmetic to Boolean) 변환 방법이 필요하다. 본 논문에서는 BtoA, AtoB의 변환 방법에 대한 자세한 방법은 생략한다.(x_1,x_2 :중간 값)

Table 2. Masking value of LEA

Masking value	
$m1$	$rand(8bit)$
$m2$	$rand(8bit)$
$m3$	$m1 \oplus m2$
$m4$	$ROR_3(m1) \oplus m1$
$m5$	$ROR_5(m1) \oplus m1$
$m6$	$ROL_9(m1) \oplus m1$

Table 3. Masking scheme for Addition

Input	$x_1 \oplus m_1, x_2 \oplus m_2, m_1, m_2, m'$
Output	$(x_1 + x_2) \oplus m'$
1. $BtoA(x_1 \oplus m_1)(= x_1 - m_1)$ 2. $BtoA(x_2 \oplus m_2)(= x_2 - m_2)$ 3. $(x_1 - m_1) + (x_2 - m_2)(= (x_1 + x_2) - (m_1 + m_2))$ 4. $[\{(x_1 + x_2) - (m_1 + m_2)\} - m'] + (m_1 + m_2)$ $(= (x_1 + x_2) - m')$ 5. Return $AtoB((x_1 + x_2) - m')(= (x_1 + x_2) \oplus m')$	

4.2 마스킹 라운드에 따른 AES와 LEA 속도비교

1차 부채널 분석에 안전하게 설계된 LEA의 효율을 판단하기 위해 AES-8bit, AES-32bit와 Fig. 9.를 통해 속도 비교를 하였다. 이 때, LEA 경진대회 및 KECCAK 속도 측정 방법[8]을 이용하였다.

AES-8bit의 경우 일반적인 방법과 1차 대응기법[9]이 적용된 경우 1,953cc → 3,256cc로 약 1.67배의 성능 차이를 보였고 AES-32bit의 경우 857cc → 1,776cc로 약 2.07배의 성능 차이를 보였다. 또한 LEA-32bit의 경우 528cc → 9,368cc로 약 17.74배의 성능 차이를 보였다. 따라서 대응기법이 적용된 LEA의 경우 속도의 효율이 현저히 저하되는 것을 알 수 있고 Addition 연산을 통해 비선형성을 유지하는 ARX 구조의 알고리즘에서 부채널 대응기법을 적용할 때 큰 속도저하를 나타내는 결과를 얻을 수 있다. 즉,

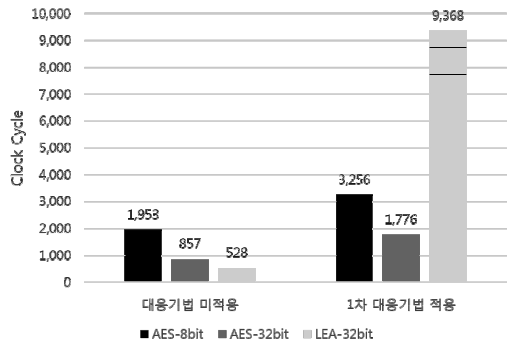


Fig. 9. Performance comparisons for AES and LEA

AES의 경우 단순히 부울린 마스킹에 대한 논리와 사전연산 테이블을 이용한 논리가 적용되지만 LEA의 경우 부울린 마스킹과 덧셈 마스킹의 변환과정 때문에 대응기법 적용에 대한 효율성이 현저히 떨어지는 것을 알 수 있다.

V. 결 론

본 논문에서는 1차 부채널 분석에 안전한 LEA 마스킹 대응기법에 대해 제안하였다. LEA 구조 특성을 이용하여 효율적으로 부채널 분석을 하는 방법과 국제 표준 암호 알고리즘인 AES와 비교를 통해 1차 부채널 분석에 안전하게 설계된 LEA의 효율성을 판단하였다. 일반적인 방법의 경우 AES에 비해 효율적인 속도를 확인할 수 있었지만 1차 부채널 분석에 안전한 대응기법이 적용된 경우 효율이 현저하게 떨어지는 것을 확인하였다. 따라서 향후 효율성이 향상된 LEA 부채널 대응법 설계 방법에 대한 연구가 필요할 것으로 사료된다.

References

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

[2] D. Kwon, J. Kim, S. Park, S. Sung, Y. Sohn, J. Song, Y. Yeom, E. Yoon, S. Lee, J. Lee, S. chee, D. Han and J. Hong, "New Block Cipher: ARIA," Proceedings of ICISC 2003, LNCS 2971, pp. 432-445, Nov, 2004.

[3] J. Park, S. Lee, J. Kim, and J. Lee, "The SEED Encryption Algorithm," RFC 4009, Dec, 2005

[4] J. Park, D. Hong, D. Kim, D. Kwon and H. Park, "128-Bit Block Cipher LEA," TTAK.KO-12.0223, Dec, 2013.

[5] C. Whitnall and E. Oswald, "A fair evaluation framework for comparing side-channel distinguishers," Journal of Cryptographic Engineering, vol. 1, no. 2, pp. 145-160, Aug, 2011.

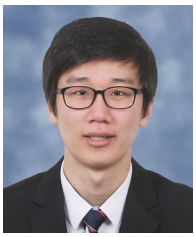
[6] L. Goubin, "A Sound Method for Switching between Boolean and Arithmetic Masking," CHES 2001, LNCS 2162, pp. 3-15, Sep, 2001.

[7] B. Debraize, "Efficient and Provably Secure Methods for Switching from Arithmetic to Boolean Masking," CHES 2012, LNCS 7428, pp. 107-121, Sep, 2012.

[8] http://keccak.noekeon.org, The source of this code is from the keccak(SHA-3) code, released in keccak homepage, 2013.

[9] C. Herbst, E. Oswald and S. Mangard, "An AES Smart Card Implementation Resistant to Power Analysis Attacks," ACNS 2006, LNCS 3989, pp. 239-252, June, 2006.

〈저자소개〉

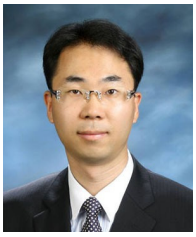


박 진 학 (Jin-hak Park) 학생회원
2013년 2월: 국민대학교 수학과 학사
2014년 2월~현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 정보보호, 부채널 분석, 대칭키 암호 알고리즘

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

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

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



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: Future Univ. -Hakodate, Post Doc.
2006년 6월~2009년 2월: 한국전자통신연구원 정보보호연구본부 선임연구원
2009년 3월~현재: 국민대학교 수학과 부교수
2009년 3월~현재: 국민대학교 일반대학원 금융정보보안학과 부교수
〈관심분야〉 공개키 암호 시스템 안전성 분석 및 고속 구현, 부채널 분석, RFID/USN 정보보호기술