Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.5, Oct. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.5.993

SEED 블록 암호 알고리즘 확산계층에서 낮은 복잡도를 갖는 부채널 분석*

원 유 승,[1†] 박 애 선,[1] 한 동 국[1,2‡]
[1]국민대학교 금융정보보안학과, [2]국민대학교 정보보안암호수학과

Side Channel Analysis with Low Complexity in the Diffusion Layer of Block Cipher Algorithm SEED*

Yoo-Seung Won,[1†] Aesun Park,[1] Dong-Guk Han[1,2‡]
[1]Dept. of Financial Information Security, Kookmin University,
[2]Dept. of Information Security, Cryptology, and Mathematics, Kookmin University

요 약

임베디드 장비의 가용성을 고려했을 때, 안전성과 효율성이 동시에 제공될 수 있는 1차 마스킹과 하이딩 대응기법과 같이 조합된 대응기법은 꽤 매력적이다. 특히, 효율성을 제공하기 위하여 첫 번째와 마지막 라운드의 혼돈 및 확산 계층에 조합된 대응기법을 적용할 수 있다. 또한, 중간 라운드에는 1차 마스킹 또는 대응기법이 없게 구성한다. 본 논문에서, 확산 계층의 출력에서 낮은 복잡도를 갖는 최신 부채널 분석을 제안한다. 일반적으로, 공격자는 높은 공격 복잡도 때문에 확산 계층의 출력을 공격 타겟으로 설정할 수 없다. 블록 암호의 확산 계층이 AND 연산들로 구성되어있을 때, 공격 복잡도를 줄일 수 있다는 것을 보인다. 여기서, 우리는 주 알고리즘을 SEED로 간주한다. 그러면, S-box 출력과 확산 계층 출력과의 상관관계에 의해 2^{32}를 갖는 공격 복잡도는 2^{16}으로 줄일 수 있다. 더욱이, 일반적으로 주 타겟이 S-box 출력이라는 사실과 비교하였을 때, 시뮬레이션 파형에서 요구되는 파형 수가 43~98%가 감소할 수 있다는 것을 입증한다. 게다가, 실제 장비에서 100,000개 파형에 대해 일반적인 방법으로 옳은 키를 추출하는 것을 실패하였음에도, 제안된 방법에 의해 옳은 키를 찾는데 8,000개의 파형이면 충분하다는 것을 보인다.

ABSTRACT

When the availability of embedded device is considered, combined countermeasure such as first-order masking and hiding countermeasures is quite attractive because the security and efficiency can be provided at the same time. Especially, combined countermeasure can be applied to the confusion and diffusion layers of the first and last rounds in order to provide the efficiency. Also, the middle rounds only employs first-order masking countermeasure or no countermeasure. In this paper, we suggest a novel side channel analysis with low complexity in the output of diffusion layer. In general, the attack target cannot be set to the output of diffusion layer owing to the high complexity. When the diffusion layer of block cipher is composed of AND operations, we show that the attack complexity can be reduced. Here, we consider that the main algorithm is SEED. Then, the attack complexity with 2^{32} can be reduced by 2^{16} according to the fact that the

Received(06. 29. 2017), Modified(08. 08. 2017), Accepted(08. 08. 2017)
* 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. 20170005200011001, SCR-Friendly 대칭키 암호 및 응용모드 개발)
† 주저자, mathwys87@kookmin.ac.kr
‡ 교신저자, christa@kookmin.ac.kr(Corresponding author)

correlation between the combination of S-box outputs and that of the outputs of diffusion layer. Moreover, compared to the fact that the main target is the output of S-box in general, we demonstrate that the required number of traces can be reduced by 43~98% in terms of simulated traces. Additionally, we show that only 8,000 traces are enough to retrieve the correct key by suggested scheme, although it fails to reveal the correct key when performing the general approach on 100,000 traces in realistic device.

Keywords: Second-Order CPA, Windowing, Diffusion Layer, SEED

I. 서 론

Paul Kocher에 의해 제안된 부채널 분석(Side Channel Analysis, SCA)[1]은 공통평가기준(Common Criteria, CC), 암호모듈 검증제도(Cryptographic Module Validation Program, CMVP) 등과 같이 암호 알고리즘이 사용되는 표준에 영향을 끼치고 있다. 특히, 특정등급의 정보보호 제품으로서 가치를 인정받기 위해서는 부채널 분석에 대한 대응기법은 필수 불가결한 사항이다.

오랜 기간 연구된 부채널 대응기법은 이론적 완성도가 이뤄진 상태이다. 하지만 정보보호 제품에 대한 가용성을 제공하기 위해서는 연구된 부채널 대응기법을 적절히 사용해야 한다. 블록 암호 알고리즘에서 이러한 타협점은 1차 부울린 마스킹 기법과 하이딩 기법을 혼용하여 활용하는 방안이다.

하지만, 이를 전체 라운드에 적용한다면, 가용성을 충분히 제공할 수 없는 경우가 대부분이다 [13]. 이를 보완하기 위하여, 부채널 공격의 주 공격 대상인 일부 라운드에만 대응기법을 적용한다.

주 공격 대상이란 부채널 분석에 추측하는 중간값에 대한 키 추측 복잡도가 낮으면서 confusion coefficient[2]가 높은 부분을 일컫는다. 따라서 이와 같은 부분에 대응기법이 적용된다. 예를 들어, 블록 암호 알고리즘 SEED의 경우 기지 평문에 대해서 방어하기 위하여 1 또는 2라운드의 G함수까지만 부울린 마스킹 및 셔플링 대응기법을 적용한다. 가용성의 범위에 따라 3라운드 이상에 대하여 부울린 마스킹 기법을 선택적으로 적용 가능하다.

정리하자면 가용성 제공을 위하여 키 추측 공격 복잡도가 낮은 부분에 대해서는 보다 안전한 부채널 대응기법을 적용하고, 나머지 부분에 대해서는 키 추측 공격 복잡도가 높아지기 때문에 상대적으로 낮은 대응기법을 적용한다.

본 논문에서는 앞서 설명한 현실적인 대응기법을 준수하였을 때, 블록 암호 알고리즘의 새로운 취약점을 서술한다. 비선형 연산인 S-box를 연산한 후, AND 연산이 포함된 블록 암호 알고리즘이 공격 대상이 된다. SEED[3], PRINCE[4] 등과 같은 블록 암호 알고리즘이 대상이 될 수 있다. 본 논문에서는 국내 블록 암호 알고리즘인 SEED를 그 대상으로 선정한다. 또한, 그 대응기법으로 SEED의 1, 16라운드에 4개 더미 연산, 셔플링 및 1차 마스킹 대응기법이 적용된 것으로 간주한다. 특히, 그 가용성을 위해 SEED의 G-함수 내의 처음 3개의 Xor과 1개의 G함수에만 더미연산 & 셔플링 대응기법 및 1차 마스킹 기법을 적용하고, 나머지 연산에는 1차 마스킹 기법만 적용되었다고 가정한다. 최종적으로, 부분적으로 1/8 셔플링 및 마스킹 대응기법을 제공한다.

이에 대해 일반적인 이차 상관 전력 분석 방법[5]을 수행한다면, 1차 마스킹 기법만 적용된 부분에 대해서는 공격 복잡도가 2^{32}으로 너무 높거나, 2^{16} 공격 복잡도를 갖는 S-box가 공격 대상일 경우 셔플링 대응기법으로 인해 본래의 공격법보다 그 파형 수가 28~784배가 더 필요하다.

본 논문에서의 제안 방법은 블록 암호의 확산계층에서의 출력 값인 SEED의 G-함수 출력 값이 연산되는 위치에서 S-box 출력을 중간 값으로 설정한다. 이는 S-box 출력 값과 G-함수 출력의 상관관계 증명을 통해, 공격 복잡도를 2^{32}에서 2^{16}으로 줄인다. 파형 수는 이론적으로 16배 증가하지만, 같은 복잡도를 제공하는 기존 공격법보다 43~98% 파형 수를 감소시키는 효과가 나타났다.

XMEGA 보드를 활용하여 실험한 결과, 본 논문에서 제안한 방법은 약 8,000개의 파형으로 공격이 가능하지만 기존의 공격법은 100,000개의 파형으로도 모두 공격이 불가능하였다.

II. 사전 지식

본 장에서는 공격 대상이 되는 블록 암호 알고리

즘 SEED와 부채널 분석 공격 결과 평가 방법론에 대해서 서술한다.

2.1 블록 암호 알고리즘 SEED와 Masked SEED

128비트 키를 사용하는 SEED-128은 총 16라운드를 거쳐 128비트 메시지를 암호화한다[3].

SEED-128은 128비트 메시지를 각각 64비트 블록 L_0과 R_0으로 나눈 뒤, Feistal 구조 내의 F-함수 16번 거쳐 암호문인 (L_{16},R_{16})을 출력한다. F-함수 내에는 S-box와 AND 연산이 포함된 G-함수와 덧셈 연산이 수행된다.

2.2 더미 연산 & 셔플링 및 1차 마스킹이 적용된 SEED

2.2.1 1차 마스킹이 적용된 SEED

블록 암호 알고리즘 SEED는 대부분 부울린 마스킹을 활용하지만, F-함수 내에 덧셈을 포함하고 있기 때문에 부분적으로 산술 마스킹으로 변환하여 사용한다. F-함수 내에 적용된 마스킹의 일부 구조[6]를 살펴보면 Fig. 1.과 같다.

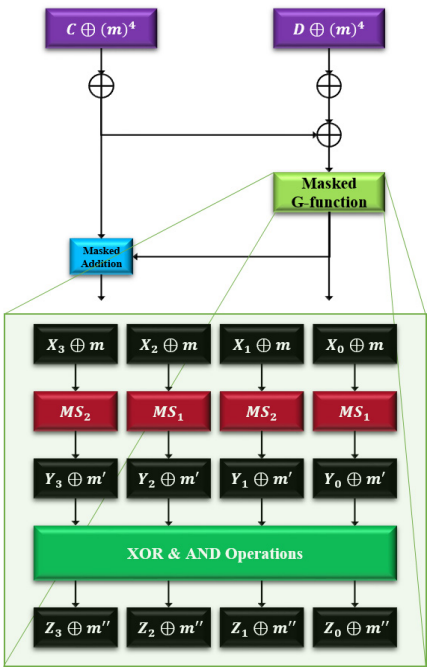


Fig. 1. G-function of Masked SEED

S-box 연산 후에, 각 바이트에 상수 값에 대한 AND 연산과 XOR 연산으로 구성된 선형 연산이다. 대응기법 관점으로 살펴보면 마스킹 값이 각 4비트에 균등하게 연산되므로 후처리가 필요하지 않다. 다음 연산을 위해 각 4바이트에 대하여 마스킹 m을 연산함으로써, 최종 출력 마스킹은 m''이 된다.

2.2.2 더미 연산, 셔플링 및 1차 마스킹이 적용된 SEED

현재 컴퓨팅 환경을 고려했을 때, 1차 마스킹 대응기법은 어렵지 않게 분석이 가능하다. 따라서 현실적으로 공격을 어렵게 하는 더미 연산 및 셔플링 대응기법을 추가적으로 삽입한다. 하지만, 이에 대한 대응기법도 연산 소비가 적게 일어나지 않기 때문에 부분 라운드에만 적용된다. 즉, 셔플링 및 고정 더미 연산을 1번째 라운드와 마지막 라운드의 1번째 G함수까지만 적용한다. 이를 더 자세히 언급하면, 셔플링 적용된 부분에 한하여 4바이트 연산에 대하여 4바이트 더미 연산을 추가한 총 1/8의 셔플링 복잡도를 제공하는 것으로 수행한다.

2.3 Guessing Entropy

Guessing Entropy[7]는 블록 암호 알고리즘에 대한 부채널 분석에서 Security Metric으로 대부분 활용되는 측정치 중 하나이다. 명확한 식은 [7]을 참조하되, 직관적인 의미는 다음과 같다. 부채널 공격법을 설정한 후, 키 분석을 실행하는데 있어 수집된 파형 중 일부를 랜덤하게 뽑아 수행한다. 이를 공격자가 설정한 수만큼 반복 공격을 하게 되어, 옳은 키의 평균 키 등수를 나타내는 것이 Guessing Entropy이다. 일정 파형 개수가 된다면, 파형을 랜덤하게 추출해도 옳은 키의 등수가 항상 1등으로 나오게 된다. 이는 우리가 흔히 말하는 부채널 분석에서 키가 도출되었다고 할 수 있다.

Guessing Entropy 값을 추출하기 위한 공격법은 이차 상관 전력 분석(Second-Order CPA, SOCPA)[5] 및 Windowing 분석[10]에 적용한다.

2.3.1 이차 상관 전력 분석(Second-Order CPA, SOCPA)

1차 부울린 마스킹이 적용된 대응기법에 대하여 부채널 분석을 수행하기 위해서는 대부분 이차 전력

분석이 사용된다. 이는 논리적으로 중간 값으로 추측이 되는 두 지점에 대한 시점을 활용한다. 예를 들어, 논리적으로 추측 가능한 두 지점을 $S(x\oplus k)\oplus m$과 m을 연산하면, $S(x\oplus k)\oplus m\oplus m = S(x\oplus k)$와 같다. (단, S:S-box, x: 평문 데이터, m: 마스킹 값)

이와 같은 두 지점이 전력 소비는 일반적으로 해밍웨이트(Hamming-Weight, HW) 모델[8]에 준하여 발생한다. 즉, 이 시점의 소비전력을 각각 t_1, t_2라 하면, $t_1 = HW(S(x\oplus k)\oplus m)$, $t_2 = HW(m)$이다. 이와 같은 가정을 기반으로 실제 전력 파형과 중간 값의 논리적인 상관관계를 활용한다. 대부분 Absolute-Difference(AD)[5] 또는 Product-Combining(PC)[9]이 사용된다. 본 논문에서 활용하는 PC는 다음과 같다. (단, $E[X]$은 확률 변수 X에 대한 기댓값을 일컫는다.)

$$(t_1 - E[t_1]) \times (t_2 - E[t_2]) \approx HW(S(x\oplus k)) \quad (1)$$

식 (1)의 좌변에서 파형의 두 지점을 전처리(pre-processing)하여 상관 전력 분석을 하는 것을 이차 전력 분석이라 일컫는다. 중간 값 8비트를 기준으로, 전력 소비가 해밍웨이트를 명확히 준수하면, 그 상관도는 최대 0.35이다.

2.3.2 Windowing 분석

Windowing 분석[10]은 셔플링 또는 더미 연산 대응기법이 적용되었을 때, 효과적으로 분석하는 방법이다. 마스킹 대응기법 적용여부와 관계없이, Windowing을 수행한 후, 이에 합당한 공격을 수행한다. Windowing은 셔플링 또는 더미 연산이 수행되는 각 연산을 명확히 구분해 내어 모두 합하여 파형을 재생성 한다. 즉, 더미 연산과 실제 연산이 총 8가지라면, 연산 시점을 명확히 구분하여 포인트 합을 구한 후, 1가지 연산의 파형으로 생성한다. 이후, 마스킹이 적용되어 있다면 이차 전력 분석을 수행한다. 따라서 본 논문에서는 수행하는 Windowing 분석은 Windowing SOCPA로 표기한다.

III. 확산 계층에서 낮은 복잡도를 갖는 부채널 분석

본 장에서는 더미 연산 & 셔플링 및 1차 마스킹이 적용된 SEED의 G-함수 출력으로의 SOCPA 공격법을 설명한다. 이를 설명하기에 앞서, 본 논문에서 제안한 방법과 기존의 방법들 간의 정량적인 평가를 위하여 몇 가지를 정의한다. 이는 다음과 같다.

Proposition 1. [공격 복잡도(Attack Complexity)] 부채널 공격 수행을 위해서 중간 값을 추측할 때, 키 추측을 위한 최소 경우의 수

예를 들어, 1차 상관 전력 분석으로 S-box 출력을 추측하기 위한 공격 복잡도는 2^8이다. S-box 출력의 1비트라도 추측하기 위해서는 키 추측을 최소 2^8 경우의 수 만큼 수행해야 되기 때문이다.

Proposition 2. [셔플링에 의한 필요 파형 수(The required number of traces for shuffling countermeasure)] 셔플링이 적용되지 않았을 때 키가 추출되는 최소 파형 수를 α라 하자. 특정 연산에 셔플링이 $1/n$만큼 적용되었을 때, 키가 추출되기 위한 최소 필요 파형 수

Proposition 2를 활용하여 SOCPA를 수행했을 때, 셔플링에서 연산 두 지점을 명확히 추출하기 위한 확률은 $\frac{1}{{}_nC_2} = \frac{2}{n(n-1)}$이다. 즉, 일반 SOCPA로 공격하는 경우, 셔플링 복잡도 역수의 제곱인 $\alpha \times \left(\frac{n(n-1)}{2}\right)^2$의 파형 수가 필요하고, windowing SOCPA 기법을 적용하였을 때 셔플링 복잡도의 역수 만큼인 $\alpha \times \frac{n(n-1)}{2}$의 파형이 필요하다[11].

3.1 일반적인 접근의 공격 방법

앞서 설명된 공격 타겟에 대하여 일반적인 접근의 공격 방법을 설명한다. 그 방법은 지금까지의 문헌에서 제안된 현실적인 공격인 SOCPA와 Windowing SOCPA를 의미한다. 또한, SOCPA에 대한 공격 지점은 더미 연산 및 셔플링 기법이 적용된 S-box 출력과 G-함수 출력을 일컫는다. 특히, G-함수 출

Table 1. Summary for general approach

Intermediate variable	Attack Scheme	Attack Complexity	The required number of traces
The output of S-box	SOCPA	2^{16}	$\alpha \times ({}_8C_2)^2$
The output of S-box	Windowing SOCPA	2^{16}	$\alpha \times {}_8C_2$
The output of G-function	SOCPA	2^{32}	α

력은 SEED의 F-함수 구성을 고려했을 때, 덧셈 연산의 입력할 때의 시점을 의미한다. 앞서 설명한 것과 같이 실제 연산 4바이트와 더미 연산 4바이트를 적용하였으므로, 이를 정리하면 다음과 같다.

Proposition 2에 의하여 S-box 출력이 타겟인 경우 각각의 요구되는 파형 수는 셔플링이 1/8로 발생하므로, 필요 파형 수는 각각 $({}_8C_2)^2$, ${}_8C_2$로 요구된다. 하지만, G-함수 출력은 셔플링이 적용되지 않으므로 요구되는 파형 수는 1배인 α이다.

3.2 G-함수 출력에서 낮은 복잡도를 갖는 이차 상관 전력 분석

본 절에서는 앞선 3가지 분석과는 달리 G-함수 출력 파형 정보를 활용한 새로운 이차 상관 전력 분석을 소개한다. 이에 앞서, 보조정리와 정리를 활용하여 G-함수 출력과 S-box의 출력의 상관관계를 증명한다.

보조정리 1. $(A \wedge B) \oplus (A \wedge C) \Leftrightarrow A \wedge (B \oplus C)$

증명. $A \oplus B$는 xor 연산의 대수적 표현 또는 진리표에 의하여 $A \oplus B \Leftrightarrow (A \vee B) \wedge (A \wedge B)^c$ 임이 밝혀져 있다.

$$
\begin{aligned}
(A\wedge B)\oplus(A\wedge C) &\Leftrightarrow \{(A\wedge B)\vee(A\wedge C)\}\wedge\{(A\wedge B)\wedge(A\wedge C)\}^c\\
&\Leftrightarrow \{A\wedge(B\vee C)\}\wedge\{A\wedge B\wedge C\}^c\\
&\Leftrightarrow \{A\wedge(B\vee C)\}\wedge\{A\wedge(B\wedge C)\}^c\\
&\Leftrightarrow \{A\wedge(B\vee C)\}\wedge\{A^c\vee(B\wedge C)^c\}\\
&\Leftrightarrow [\{A\wedge(B\vee C)\}\wedge A^c]\vee[\{A\wedge(B\vee C)\}\wedge(B\wedge C)^c]\\
&\Leftrightarrow A\wedge(B\vee C)\wedge(B\wedge C)^c\\
&\Leftrightarrow A\wedge(B\oplus C) \qquad \square
\end{aligned}
$$

보조정리 1을 활용하여 G-함수 출력의 두 중간 값의 조합을 유도한다. 이를 위해서 기호 표기를 다음과 같이 정의한다.

Table 2. Notations

Notation	Description
i	i^{th} byte $(0 \leqq i \leqq 3)$
Y_i	The output of S-box
Z_i	The output of G-function
m''	Masking value for the output byte of G-function

G-함수 출력의 각 바이트 Z_i의 두 조합은 다음 정리 1에 의하여 S-box출력 바이트 Y_i의 두 바이트로 표현이 가능하다. 이는 G-함수 출력 값 두 바이트의 조합으로 SOCPA를 수행 가능함을 의미한다.

정리 1. $(Z_3 \oplus Z_2) = \{(Y_1 \oplus Y_0\} \wedge 11000000_2\} \oplus \{(Y_3 \oplus Y_0\} \wedge 00110000_2\} \oplus \{(Y_3 \oplus Y_2\} \wedge 00001100_2\} \oplus \{(Y_2 \oplus Y_1\} \wedge 00000011_2\}$

증명. 마스킹 된 G-함수 출력의 두 바이트는 다음과 같다.

$Z_3 \oplus m'' = (Y_3 \wedge 11001111_2) \oplus (Y_2 \wedge 11110011_2) \oplus (Y_1 \wedge 11111100_2) \oplus (Y_0 \wedge 00111111_2) \oplus m''$

$Z_2 \oplus m'' = (Y_3 \wedge 11110011_2) \oplus (Y_2 \wedge 11111100_2) \oplus (Y_1 \wedge 00111111_2) \oplus (Y_0 \wedge 11001111_2) \oplus m''$

보조정리 1에 의하여

$$
\begin{aligned}
&(Z_3 \oplus m'') \oplus (Z_2 \oplus m'')\\
&= Z_3 \oplus Z_2 = \{Y_3 \wedge (11001111_2 \oplus 11110011_2)\} \oplus \{Y_2 \wedge (11110011_2 \oplus 11111100_2)\} \oplus \{Y_1 \wedge (11111100_2 \oplus 00111111_2)\} \oplus \{Y_0 \wedge (00111111_2 \oplus 11001111_2)\}\\
&= (Y_3 \wedge 00111100_2) \oplus (Y_2 \wedge 00001111_2) \oplus (Y_1 \wedge 11000011_2) \oplus (Y_0 \wedge 11110000_2)
\end{aligned}
$$

$\therefore (Z_3 \oplus Z_2) = \{(Y_1 \oplus Y_0\} \wedge 11000000_2\} \oplus \{(Y_3 \oplus Y_0\} \wedge 00110000_2\} \oplus \{(Y_3 \oplus Y_2\} \wedge 00001100_2\} \oplus \{(Y_2 \oplus Y_1\} \wedge 00000011_2\}$ $\square$

정리 1을 활용하여 S-box 출력 2바이트의 부분 정보의 논리 값과 G-함수 출력 2바이트 결과에 해당하는 전력 파형 정보 사이의 상관도가 있음을 보였다. 본 논문에서는 이를 이용하여 낮은 복잡도를 갖는 새로운 이차 상관 전력 분석을 제안한다. 또한,

Table 3. Summary for our suggestion

Intermediate variable	Attack Scheme	Attack Complexity	The required number of traces
The output of G-function	SOCPA	2^{16}	16α

정리 1 이외에도 G-함수 출력의 여러 가지 조합이 S-box 출력의 조합으로 표현 가능하다. 즉, 같은 방법으로 5가지 조합이 존재한다.

이를 기반으로 본 논문에서 제안하는 방법을 요약하면 다음 표와 같다.

위의 표에서 16배가 증가하는 이유는 8비트 중 2비트 상관도만 있기 때문이다. 이는 Noise-free 환경에서 상관도를 입력 가능한 값에 대한 전수 조사하였을 때, 8비트의 값과 2비트의 상관도는 0.25이다. [11]에 의하여 필요파형 수는 상관도의 역수의 제곱에 비례한다. 따라서 $\left(\frac{1}{0.25}\right)^2$배의 파형 수가 필요하다. 따라서 이론적으로 2^{16} 공격복잡도 내에서 $28\alpha \sim 784\alpha$배의 파형에서 16α의 파형이 필요한 것으로 판단된다. 즉, 43~98%의 파형을 감소시켜 분석이 가능하다는 것을 알 수 있다.

요컨대, G-함수 출력의 조합 $Z_3 \oplus Z_2$에 대한 공격을 수행하기 위하여, $(Y_1 \oplus Y_0) \wedge 11000000_2$} 또는 $(Y_3 \oplus Y_0) \wedge 00110000_2$} 또는 $(Y_3 \oplus Y_2) \wedge 00001100_2$} 또는 $\{(Y_2 \oplus Y_1) \wedge 00000011_2$}를 중간 값으로 설정하면 된다.

IV. 실험 결과

3장에서의 기존 방법 중 현실적인 공격이 가능한 공격복잡도 2^{16}을 갖는 2가지와 본 논문에서 제안한 방법을 서로 비교한다. 즉, 총 3가지에 대한 공격을 시뮬레이션 파형과 실제 파형에서 수행한다.

4.1 시뮬레이션 파형에 대한 실험 결과

시뮬레이션을 수행하기 위해서는 전력 소비 모델의 정의가 필요하다. 본 논문에서는 잡음이 전혀 없는 환경에서 해밍웨이트 모델을 따르는 것으로 설정한다. 즉, 공격 중간 값에 대한 해밍웨이트 값을 파형으로 생성한다. 이에 대한 실험 결과는 다음과 같다.

Fig. 2.는 3가지 공격에 대한 Guessing Entropy 결과를 나타낸다. Guessing Entropy를

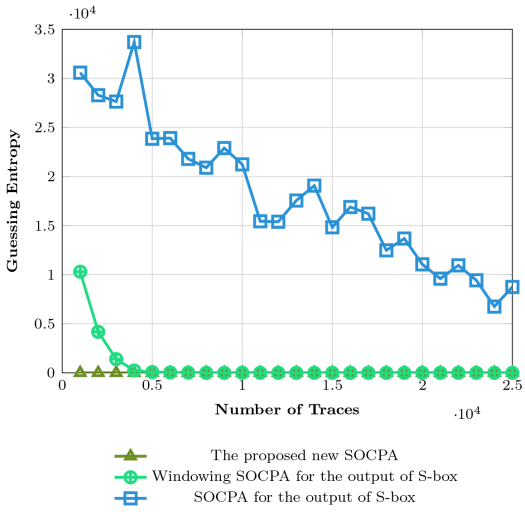


Fig. 2. Experimental Result for Simulated Traces

계산하기 위하여 100,000개 시뮬레이션 된 파형을 생성하여 1,000개부터 25,000개 파형을 1,000개씩 증가시켜 가면서 분석하였다. 또한, 같은 파형 수로 100,000개 내에 랜덤 선택하여 분석을 50번 반복하였다. SOCPA는 공격복잡도 2^{16}을 가지므로, Guessing Entropy 값은 1부터 65,536 사이 값을 갖는다.

더미 연산과 셔플링 대응기법이 적용된 S-box 위치에서 공격할 경우, 초록색 선을 나타내는 Windowing SOCPA는 약 12,000개에서 Guessing Entropy가 1로 수렴하고 SOCPA의 경우에는 그래프 상에서 나타나지 않는 범위인 100,000개 이상의 파형이 필요하였다. 또한, 본 논문에서 제안된 방법을 G-함수 출력부분에 대해 사용한 경우는 약 2,000개 파형만으로 Guessing Entropy가 1로 수렴하는 것을 확인할 수 있었다. 따라서 시뮬레이션 상에서는 약 5~50배 이상의 파형 수를 줄여 공격을 수행할 수 있다.

4.2 실제 파형에 대한 실험 결과

본 절에서는 실제 수집된 파형에 대하여 실험을 수행한다. XMEGA128D4-U를 탑재한 Chipwhisperer-Lite(CW1173)[12]로 측정하여 실험을 진행하였다. 실험 결과는 다음과 같다.

Fig. 3.는 Fig. 2.와 실험 변수는 동일하다. 다른 점은 수집된 분석 파형이 다르다고 할 수 있다.

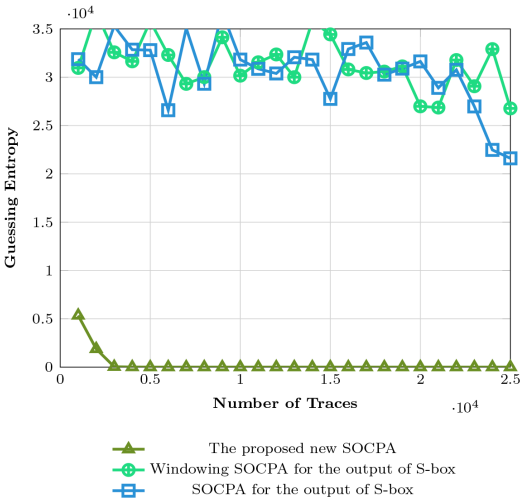


Fig. 3. Experimental Result for Real Traces

본 논문에서 제안한 공격을 수행하기 위하여 시뮬레이션 상으로는 G-함수 출력 값의 해밍웨이트 값으로 생성하였다. 실제 파형에서는 셔플링 대응기법을 우회하기 위하여 G-함수 출력 값을 Load하는 부분인 G-함수 내의 덧셈 부분에서 파형을 수집하였고, 기존 방법들에 대한 공격을 수행하기 위하여 G-함수 연산 부분을 따로 수집하였다.

본 논문에서 제안한 방법은 약 8,000개 파형으로 정확히 Guessing Entropy가 1로 수렴한다. 하지만, 기존에 존재하는 공격법들은 25,000개뿐만 아니라 총 100,000개에서 모두 분석이 불가능하였다. 이는 실험보드의 잡음이 적은 것으로 알려져 있으나 [12], 기존에 존재하는 2가지 공격법 모두 100,000개 이상의 파형이 필요할 것으로 판단된다.

V. 결 론

확산계층에서 AND연산을 갖는 블록 암호 알고리즘에 대하여 새로운 공격법을 본 논문에서 제안하였다. 그 예를 블록 암호 알고리즘 SEED에 현실적인 대응기법을 적용하였을 때, 이론적으로는 43~98% 파형 수 감소를 예상할 수 있었다. 또한, 실험적으로 기존 공격법으로는 100,000개에서 공격이 불가능하였지만, 제안된 방법으로는 약 8,000개에서 공격이 가능함을 보였다.

차후, 본 논문에서 제안된 공격에 대하여 다른 암호 알고리즘 및 잡음이 심한 경우에 대한 추가적인 실험과 현실적인 대응기법에 대한 연구가 필요하다.

References

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis", CRYPTO' 99, LNCS 1666, pp. 388-397, 1999.

[2] Y. Fei, A.A. Ding, J. Lao, and L. Zhang, "A Statistics-based Fundamental Model for Side-channel Attack Analysis", IACR ePrint 2014-152, Feb. 2011.

[3] Korea Internet and Security Agency, Block Cipher Algorithm SEED, Available from https://seed.kisa.or.kr/html/egovframework/iwt/ds/ko/ref/[2]_SEED+128_Specification_english_M.pdf;jsessionid=303F153DDB99E8847A8CC919C4E4BAFE

[4] J. Borghoff, T. Güneysu, E.B. Kavun, M. Knezevic, L.R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S.S. Thomsen, and T. Yalcin, "PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications", ASIACRYPT 2012, LNCS 7658, pp. 208-225, 2012.

[5] M. Joye, P. Paillier, and B. Schoenmakers, "On Second-Order Differential Power Analysis", CHES 2005, LNCS 3659, pp. 293-308, 2005.

[6] H. Kim, Y.I. Cho, D. Choi, D.-G. Han, and S. Hong, "Efficient Masked Implementation for SEED Based on Combined Masking", ETRI Journal, vol. 33, no. 2, pp. 267-274, Apr. 2011.

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

[8] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model", CHES 2004, LNCS 3156, pp. 16-29, 2004.

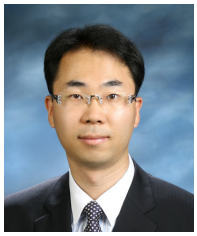
[9] E. Prouff, M. Rivain, and R. Bevan,

"Statistical Analysis of Second Order Differential Power Analysis", IEEE Transactions on Computers, vol. 58, no. 6, pp. 799-811, June 2009.
[10] J.-W. Cho and D.-G. Han, "Security Analysis of the Masking-Shuffling based Side Channel Attack Countermeasures", International Journal of Security and Its Applications, Vol. 6, no. 4, pp. 207-214, Jan. 2012.
[11] S. Mangard, E. Oswald, and T. Popp: Power Analysis Attacks - Revealing the Secrets of Smart Cards, Springer US, New York, 2007.
[12] C. O'Flynn and Z. Chen, "ChipWhisperer: An Open-Source Platform for Hardware Embedded Security Research", COSADE 2014, LNCS 8622, pp. 243-260, 2014.
[13] D. Goudarzi and M. Rivain, "How Fast Can Higher-Order Masking Be in Software?", EUROCRYPT 2017, LNCS 10210, pp. 567-597, 2017.

〈저자소개〉

원 유 승 (Yoo-Seung Won) 학생회원
2012년 2월: 국민대학교 수학과 학사
2014년 2월: 국민대학교 수학과 석사
2014년 3월~현재: 국민대학교 금융정보보안학과 박사과정
〈관심분야〉 정보보호, 부채널 분석, 대칭키 암호 알고리즘, 스마트 카드 보안

박 애 선 (Aesun Park) 학생회원
2011년 2월: 국민대학교 수학과 학사
2013년 2월: 국민대학교 수학과 석사
2014년 3월~현재: 국민대학교 금융정보보안학과 박사과정
〈관심분야〉 부채널 분석 및 대응법, 스마트 카드 평가, Post-quantum cryptography 등



한 동 국 (Dong-Guk Han) 종신회원
1999년 2월: 고려대학교 수학과 졸업(학사)
2002년 2월: 고려대학교 수학과 석사 (이학석사)
2005년 2월: 고려대학교 정보보호대학원 박사 (공학박사)
2004년 4월~2005년 4월: 일본 Kyushu Univ., 방문연구원
2005년 4월~2006년 4월: 일본 Future Univ.-Hakodate, Post.Doc.
2006년 6월~2009년 2월: 한국전자통신연구원 정보보호연구단 선임연구원
2009년 3월~현재: 국민대학교 정보보안암호수학과 교수
〈관심분야〉 공개키 암호시스템 안전성 분석 및 고속 구현, 부채널 분석 및 대응법 설계, IoT 정보보호 기술