Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.35, NO.1, Feb. 2025 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2025.35.1.37

HW 구현 대칭키 암호에 대한 범용적 딥러닝 기반 프로파일링 부채널 분석 방안*

최 기 훈,[1†] 오 충 연,[1] 김 주 환,[2] 박 혜 진,[2] 한 동 국[3‡]
[1,2,3]국민대학교 (학생, 대학원생, 교수)

Universal Deep Learning-Based Profiling Side-Channel Analysis Method on HW Implementation Symmetric Key Algorithm*

Ki-Hoon Choi,[1†] Chung-Yeon Oh,[1] Ju-Hwan Kim,[2] Hye-Jin Park,[2] Dong-Guk Han[3‡]
[1,2,3]Kookmin University (Undergraduate, Graduate student, Professor)

요 약

딥러닝 기반 부채널 분석의 성능은 중간값에 의해 크게 좌우된다. 분석자는 비밀키와 관련되며 부채널 정보와 관련성이 높은 중간값을 딥러닝 모델의 목푯값으로 활용해야 한다. 암호 알고리즘의 하드웨어 구현물은 일반적으로 레지스터에 저장된 데이터와 저장할 데이터의 차분을 중간값으로 사용하여 분석한다. 이를 위해서는 공격자가 레지스터에 저장되는 데이터의 흐름을 알고 있어야 한다. 즉, 효율적인 분석 방법이 하드웨어 구조에 따라 달라질 수 있다. 본 논문에서는 이를 극복하고자 하드웨어 암호를 대상으로 범용적으로 적용할 수 있는 딥러닝 기반 프로파일링 분석 방안을 제안한다. 제안한 방안은 대표적인 하드웨어 구현물 구조를 분석 후 효과적인 중간값을 조합해 비밀키를 판정한다. 제안한 방법을 실험적으로 검증하기 위해 두 종의 대표적 하드웨어 구현물을 분석했다. 단일 중간값 활용 시 하드웨어 구현 방식에 따라 비밀키 분석에 실패한 경우가 있었으나, 제안한 방안은 두 구현물 모두에서 비밀키 분석에 성공했다. 이는 제안한 분석 방안을 범용적으로 활용할 수 있음을 시사한다.

ABSTRACT

The performance of deep learning-based side-channel analysis is highly affected by the target intermediate value. The security analyzer should choose an intermediate value related to the side-channel information as a label. In general, the difference between the value stored in a register and the value to be stored is chosen as an intermediate value for a hardware cryptographic implementation. It requires a data flow of the data register; the efficient analysis method could be different by the hardware architecture. This paper proposes a universal deep learning-based profiling attack method for hardware cryptographic implementation. The proposed method reveals the secret key by combining several intermediate values related to the typical hardware implementations. We demonstrate the proposed method by performing the proposed and existing deep learning-based profiling attacks. The existing method, which utilizes a single intermediate value, failed to reveal the secret key for a specific implementation. Besides, the proposed method discloses each key of two implementations. This result implies that the proposed method is universally utilizable.

Keywords: Deep Learning, Side-Channel Analysis, Template Attack, FPGA

Received(10. 17. 2024), Modified(12. 10. 2024), Accepted(12. 11. 2024)

* 이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. RS-2024-00394190, 온-디바이스 자율보호가 내재화된 개방형 영상보안플랫폼 기술 개발)

† 주저자, weeikm@kookmin.ac.kr
‡ 교신저자, christa@kookmin.ac.kr(Corresponding author)

I. 서 론

부채널 분석(Side-Channel Analysis, SCA)은 암호 장비에서 알고리즘 동작 시 발생하는 소비전력, 전자기파 등의 부채널 정보를 이용해 비밀 정보를 획득하는 분석 기법이다[1]. 대표적인 통계기법 기반 부채널 분석 기법인 상관 전력 분석(Correlation Power Analysis, CPA)은 암호 장비의 소비전력과 암호 알고리즘의 중간 연산값 사이의 상관관계를 이용하여 비밀키를 분석한다[2]. 이를 위해 공격자는 소비전력과 중간값의 관계를 나타내는 모델인 소비전력 모델을 선정해야 한다. 대표적인 소비전력 모델로는 해밍 웨이트(Hamming weight) 모델과 해밍 디스턴스(Hamming distance) 모델이 있다[3]. 중간값과 부채널 정보의 관계는 암호 장비의 특성에 따라 달라진다. 소프트웨어 구현물은 중간값의 비트 1의 개수인 해밍 웨이트가 소비전력과 선형 관계임이 알려져 있다. 하드웨어 구현물은 레지스터의 상태 전이 횟수가 소비전력과 선형 관계임이 알려져 있다. 따라서 하드웨어 대상 상관 전력 분석은 일반적으로 레지스터에 저장된 중간값과 저장할 중간값의 해밍 디스턴스를 소비전력 모델로 이용한다. 이를 위해 분석자는 레지스터에 저장되는 값을 알고 있어야 한다. 즉, 세부 하드웨어 구조가 기지여야 한다. 한편, 딥러닝 기반 부채널 분석은 부채널 정보와 중간값의 관계를 직접 모델링하므로 별도 소비전력 모델을 상정하지 않아도 된다[4]. 하지만 이는 딥러닝 모델을 암호 장비의 특성과 무관하게 설계해도 됨을 의미하지 않는다. 딥러닝 기반 분석에서도 소비전력에 주요한 영향을 미치는 레지스터의 상태 전이와 관련된 값을 목푯값(label)으로 설정하는 것이 효율적이다[5]. 따라서 딥러닝 기반 분석 역시 효율적 분석을 위해서는 세부 하드웨어 구조가 기지여야 한다.

하드웨어 구조가 미지인 블랙박스 환경에서 단일 중간값을 선정하기 어렵지만, [6]은 암호 알고리즘 AES(Advanced Encryption Standard) 마지막 라운드의 비선형 연산 계층인 SubBytes 입력을 중간값으로 활용해 비밀키 복구에 성공했다. 하지만 해당 연구는 구현물 한 종에 대해서 분석했으므로 활용한 중간값은 [6]에서 분석한 구현물에서만 효율적일 가능성이 있다. 즉, 구현에 따라 효율적인 중간값이 상이하므로 기존 방법은 다른 하드웨어 구조에 적용하기 어려울 수 있다. 따라서 블랙박스 환경에서 범용적으로 활용할 수 있는 분석 방안이 필요하다.

〔contribution〕

- 하드웨어 구현 대칭키 암호 대상 범용적으로 사용할 수 있는 분석 방안을 제안한다. 기존 연구는 하드웨어 구조를 기반으로 적합한 중간값을 선정한다. 반면, 제안 방안은 다양한 중간값을 조합해 비밀키를 분석하므로 하드웨어 구조가 미지인 경우에도 활용할 수 있다.
- 제안한 방안을 두 가지 공개 하드웨어 암호 모듈에 대해 검증한다. 기존 방안은 하드웨어 구조에 적합한 중간값에서만 효율적인 분석이 가능하다. 반면, 제안한 방안은 두 구조 모두에서 높은 분석 성능을 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 딥러닝 기반 프로파일링 분석과 하드웨어 대상 분석 연구 동향을 기술한다. 3장에서는 하드웨어 구현물 분석에 적합한 중간값을 제안하고 이를 활용해 블랙박스 환경에 적합한 분석 방안을 제안한다. 4장에서는 제안한 분석 방안을 검증하기 위해 일반적인 AES 하드웨어 구현물 두 종에 대해 기존 방안과 제안 방안의 성능을 비교한다. 마지막으로 5장에서는 본 논문의 결론을 제시한다.

II. 관련 연구

2.1 딥러닝 기반 프로파일링 분석

딥러닝 기반 프로파일링 분석은 완전한 제어권을 가진 장비로 생성한 딥러닝 모델 기반 프로파일을 분석 대상 장비에서 수집한 부채널 정보와 대응하여 비밀키를 복구하는 방법이다[7]. 딥러닝 기반 프로파일링 분석은 공격자가 공격 대상 기기와 동일하고 평문과 암호문, 키 제어가 가능한 프로파일링 기기에 접근 가능하다는 공격자 가정을 갖는다. 분석 과정은 프로파일링 단계와 공격 단계로 구성된다.

- 프로파일링 단계
  프로파일링 기기에서 암호 알고리즘이 동작할 때의 부채널 정보를 다량 수집한다. 입력값, 목푯값을 각각 부채널 정보와 중간값으로 설정하여 신경망에게 중간값과 부채널 정보 간 관계를 학습시킨다.
- 공격 단계
  공격 대상 기기의 부채널 정보를 학습된 딥러닝 모델에 입력해 중간값을 예측한다. 이를 평문 또

는 암호문과 조합해 비밀키를 복구한다.

일반적으로 딥러닝 모델은 부채널 정보를 중간값별로 분류하는 문제를 해결하기 위해 활용된다[7, 8]. 이때 딥러닝 모델의 출력은 각 중간값에 대응되는 확률로 취급된다. 이를 키별 확률로 변환하기 위해 평문 또는 암호문을 활용할 수 있다. 예컨대 AES 1라운드 SubBytes 출력을 중간값으로 사용할 경우, 딥러닝 모델은 부채널 정보 T_t에 대해 해당 중간값 $v = SBox[x_t \oplus k]$가 0부터 255까지일 확률 $p_{t,0}$부터 $p_{t,255}$를 출력한다. x_t, k는 각각 한 바이트 평문과 키를 의미하고, t는 부채널 정보의 색인, $p_{t,i}$는 중간값이 i일 확률을 의미한다. 중간값별 확률은 평문을 활용해 키별 확률 $s_{t,0}$부터 $s_{t,255}$로 변환할 수 있다. 즉, 키가 k일 확률 $s_{t,k}$를 수식 (1)과 같이 계산할 수 있다. 정확도를 높이기 위해 수식 (2)와 같이 다수의 부채널 정보에 대한 각 확률을 곱하여 최종 확률 S_k를 도출할 수 있다.

$$s_{t,k} = p_{t,v} = p_{t,SBox[x_t \oplus k]} \tag{1}$$

$$S_k = \prod_{t=0}^{n-1} s_{t,k} \tag{2}$$

부채널 분석의 성능을 평가하기 위해 일반적으로 분석을 다회 수행했을 때 비밀키의 평균 등수인 게싱 엔트로피(guessing entropy)를 활용한다[9]. 프로파일링 분석에서 비밀키의 등수는 앞서 구한 키별 확률을 내림차순으로 정렬했을 때 순위로 계산된다. 게싱 엔트로피가 0으로 수렴한 것은 모든 공격에서 비밀키 분석이 성공했음을 의미한다. 본 논문에서는 게싱 엔트로피가 0으로 수렴하기 위한 부채널 정보의 개수인 최소 분석 파형 수(Minimum number of Traces to Disclosure, MTD)를 성능 지표로 활용한다.

2.2 하드웨어 대상 딥러닝 기반 프로파일링 분석 동향

딥러닝 기반 프로파일링 분석의 성능은 중간값에 크게 의존한다. 분석자는 암호 알고리즘의 구현에 따라 비밀키와 관련 있으며 부채널 정보와 연관성이 높은 중간값을 선정해야 한다. 하드웨어 대상 암호 구현물의 부채널 정보는 일반적으로 레지스터의 상태 전이에 큰 영향을 받는다[3]. 따라서 기존 연구는 각 중간값이 딥러닝 기반 부채널 분석 성능에 미치는 영향을 주로 분석했다.

[10]은 하드웨어 암호 구현물 대상 부채널 분석에 딥러닝을 최초로 적용한 연구이다. 해당 연구에서는 중간값으로 1라운드 SubBytes 출력을 활용했다. 해당 연구에서는 레지스터 상태 전이와 관련된 중간값이 아닌 레지스터에 저장된 값 자체를 중간값으로 사용했다. 이는 CPA와 같은 통계기법 기반 분석에서는 일반적으로 사용하지 않는 중간값도 딥러닝 기반 분석에 활용할 수 있음을 시사한다. [11]은 CNN(Convolutional Neural Network)의 유효성을 검증한 연구이다. 해당 연구에서는 일반적으로 하드웨어 분석 시 사용하는 중간값인 10라운드 입출력 차분을 활용했다. [6]은 하드웨어 세부 구조가 미지인 블랙박스 환경에서 활용할 수 있는 중간값을 제안했다. 블랙박스 환경에서는 레지스터 상태 전이와 관련된 중간값을 선정할 수 없다. 따라서 그들은 치환계층인 SubBytes 입력을 중간값으로 활용해 비밀키를 복구했다. 입력을 고려한 이유는 마지막 라운드를 대상으로 하기 때문이다.

한편, [5]는 레지스터 상태 전이와 관련된 중간값뿐만 아니라, 다양한 중간값을 활용하는 방안인 tandem 기법을 제안했다. 3개의 딥러닝 모델에 목푯값을 각각 10라운드의 입력, SubBytes 출력, 라운드 입출력의 차분으로 하여 학습시킨 후 각 모델이 도출한 키별 확률을 곱하여 최종 확률 $S_{tandem,k}$를 계산한다. 이는 아래 수식 (3)으로 나타낼 수 있다. $S_{m,k}$는 모델 m의 키가 k일 확률을 의미한다. 해당 연구에서는 전통적 중간값인 레지스터 차분 이외의 값을 활용하여 부채널 분석의 성능을 높였다.

$$S_{tandem,k} = \prod_{m=0}^{2} S_{m,k} \tag{3}$$

선행 연구에서 딥러닝 기반 프로파일링 분석은 다양한 중간값을 활용할 수 있음을 보였다. 그러나, 대부분의 기존 연구는 분석자가 하드웨어 구조가 기지임을 상정하여 중간값을 선정하거나, 블랙박스 환경에서 범용적으로 적용하기 어려운 단일 중간값을 활용했다.

III. 블랙박스 하드웨어 구현물에 대한 딥러닝 기반 부채널 분석 방안

3.1 하드웨어 구현에 따른 중간값

암호 알고리즘을 하드웨어로 구현할 때는 속도와 면적을 고려해야 한다. 대칭키 암호 알고리즘은 라운드 함수를 반복 연산한다[12]. 따라서 속도를 고려하는 경우 라운드 함수 전체에 대한 하드웨어 모듈을 구성하고, 해당 모듈에 데이터를 반복 입력하도록 설계한다. 반면, 면적을 고려하는 경우 라운드 함수를 나누어 작은 모듈로 구성한다. 라운드 함수 내부 계층은 여러 바이트에 동일 연산을 수행한다. 따라서 내부 계층을 작은 모듈로 설계한 후, 해당 모듈에 데이터를 반복 입력하도록 구현할 수 있다. 일반적으로 하드웨어 구현 시 비선형 함수가 가장 많은 면적을 차지한다[13, 14]. 따라서 면적을 고려하는 경우 비선형 함수를 나누어 연산하도록 설계한다. 예컨대, 암호 알고리즘 AES의 비선형 계층인 SubBytes는 바이트 단위로 동일한 연산을 수행한다. 따라서 4바이트의 SubBytes 연산을 수행하는 모듈을 구성하고, 해당 모듈에 데이터를 4번 반복 입력하여 SubBytes를 구현할 수 있다.

Fig. 1.은 두 구현에서 레지스터에 저장되는 데이터의 흐름을 도식한 그림이다. 좌측 그림은 속도를 고려한 구현물, 우측 그림은 면적을 고려한 구현물의 데이터 흐름이다. 그림의 각 기호의 의미는 Table 1.과 같다.

하드웨어 구현물의 부채널 정보는 레지스터 상태 전이와 연관성이 크다. 따라서 부채널 분석 시 적합한 중간값은 구현에 따라 다르다. 속도를 고려한 구현은 라운드 함수의 출력이 레지스터에 저장된다. 면적을 고려한 구현은 라운드 출력 혹은 비선형 연산 출력이 레지스터에 저장된다. 따라서 첫 번째 구현은 라운드 입출력의 차분, 즉 Table 1.의 $\Delta RD_{j,i}$이, 두 번째 구현은 비선형 연산 입출력의 차분 $\Delta SB_{j,i}$, 비선형 연산 출력과 라운드 출력의 차분 $\Delta AR_{j,i}$이 중간값으로 적합하다. 한편, 면적 최적화를 위해 SubBytes를 4 바이트 단위가 아닌 1 바이트 단위로 구현한 경우에도 중간값은 동일하다.

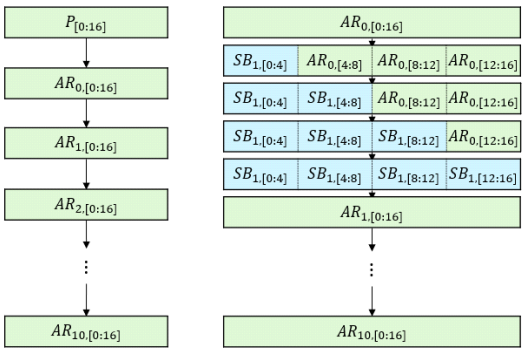


Fig. 1. Data stored in the register at each clock cycle (left and right are hardware architecture optimizing time and area, respectively)

AES의 마지막 라운드는 MixColumns 계층을 포함하지 않으므로 차분 계산에 유리하다. 따라서 마지막 라운드와 관련된 중간값을 선정하는 것이 적합하다. 분석에 활용할 수 있는 추가적인 마지막 라운드 관련 중간값은 다음과 같다. 마지막 라운드 계층 중 ShiftRows는 단순히 바이트 위치만 변경하므로, AddRoundKey와 SubBytes 출력을 중간값으로 활용할 수 있다. 종합적으로 두 구현에서 분석한 세 개의 레지스터 차분 및 마지막 라운드의 계층 출력 두 개를 합해 총 다섯 개의 중간값을 활용한다. 각 중간값은 수식 (4)~(8)과 같이 계산할 수 있다. 수식의 C_i는 i번째 바이트 암호문, $RK_{j,i}$는 j번째 라운드 키의 i번째 바이트를 의미한다. $InvSBox$는 SubBytes 함수에서 쓰이는 SBox의 역 테이블, SR^{-1}은 ShiftRows의 역함수를 의미한다. Table 1.은 분석에 활용할 다섯 개의 중간값에 대한 기호이며, 색인은 각 수식과 대응 관계를 나타낸다.

Table 1. Symbols of the intermediate values

Index	Symbol	Meaning
(4)	$AR_{j,i}$	jth round, ith byte AddRoundKey output
(5)	$SB_{j,i}$	jth round, ith byte SubBytes output
(6)	$\Delta RD_{j,i}$	ith byte of difference between jth round input and output
(7)	$\Delta SB_{j,i}$	ith byte of difference between jth SubBytes input and output
(8)	$\Delta AR_{j,i}$	ith byte of difference between jth round SubBytes output and AddRoundKey output

$$SR^{-1}(InvSBox[C_i \oplus RK_{10,i}]) \quad (4)$$

$$C_i \oplus RK_{10,i} \quad (5)$$

$$SR^{-1}(InvSBox[C_i \oplus RK_{10,i}]) \oplus C_i \quad (6)$$

$$SR^{-1}(InvSBox[C_i \oplus RK_{10,i}] \oplus C_i \oplus RK_{10,i}) \quad (7)$$

$$SR^{-1}(C_i \oplus RK_{10,i}) \oplus C_i \quad (8)$$

3.2 범용적 딥러닝 기반 프로파일링 분석 방안

하드웨어 세부 구조가 미지이면 적합한 중간값을 선정하기 어렵다. 따라서 선행 연구[6]와 같이 비선형 연산 계층의 입력 혹은 출력을 활용하거나, 역공학 등을 통해 하드웨어 구조를 사전에 분석해야 한다. 우리는 tandem[5] 모델을 활용하여 블랙박스 환경에서 효율적인 딥러닝 기반 부채널 분석 방안을 제안한다. 데이터와 목푯값의 관계가 불명확하여 딥러닝 모델이 둘의 관계를 학습하지 못하면 일반적으로 딥러닝 모델은 임의의 값을 출력한다[15]. 즉, 부적합한 중간값을 학습한 모델은 모든 값의 추정 확률이 균등하다. 반면, 적합한 중간값을 학습한 모델은 옳은 중간값의 추정 확률만 높다.

따라서 3.1절에서 제안한 중간값들을 Fig. 2.와 같이 각각 별도의 딥러닝 모델에 학습시킨 후 이를 수식 (3)과 같이 곱하여 최종 확률을 도출하는 방안을 범용적 분석 방안으로 제안한다. 제안한 방안은 적합한 중간값을 학습한 모델이 최종 확률에 주요한 영향을 미치므로 범용적으로 활용할 수 있다.

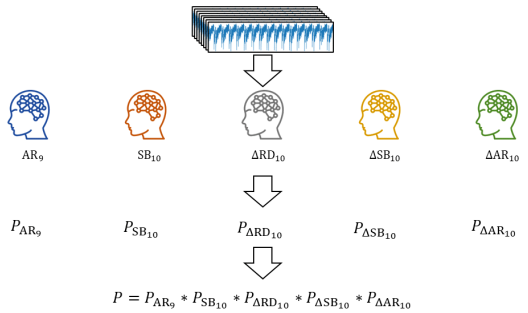


Fig. 2. Overall structure of model training

IV. 중간값별 딥러닝 기반 프로파일링 분석 결과

4.1 실험 환경

본 논문에서는 속도를 고려한 구현물인 Google사 Project Vault AES[16]와 면적을 고려한 구현물인 Joachim Strömbergson의 Project secworks[17]를 분석한다. 본 논문에서는 Google사 Project Vault AES를 target 1로 정의하고, Joachim Strömbergson의 Project secworks를 target 2로 정의한다. 각 하드웨어 구현물을 Xilinx Artix-7 기반 FPGA(Field-Programmable Gate Array)를 사용하는 부채널 분석 테스트보드 CW305에 업로드하고, 암호화 시의 소비전력을 CW-Lite로 측정했다. 세부 환경은 Table 2.와 같다. 학습과 분석을 위해 각각 1,000,000번 암호화할 때의 소비전력을 수집했다. 학습 데이터는 임의의 평문과 키로, 분석 데이터는 임의의 평문과 고정된 키로 암호화할 때의 소비전력을 수집했다.

신경망은 MLP(Multi-Layer Perceptron)와 CNN(Convolutional Neural Network) 두 가지로 Table 3.과 같이 구성했다. 출력층의 활성화 함수로는 softmax를 사용해 신경망의 출력을 확률로 표현했다. 충분히 많은 신경망 구조로 검증 실험을 수행하기 위해 구현물별 총 450개의 모델을 학습시켜 0바이트를 분석했다. Table 3.에서 hidden layers는 MLP의 각 은닉층의 노드 수를 의미하며, convolution layer는 CNN의 각 은닉층의 필터 수를 의미한다. 각 신경망은 독립적으로 동일한 초기 가중치를 사용했다. 본 논문에서는 분석 데이터를 1

Table 2. Specification of the target device and the measurement environment

Device	Feature	Notes / Range
Capture board (CW-Lite)	ADC specs	10-bit 105MS/s
	Sample Buffer Size	24,573 samples
	Sampling rate	40MHz
Target board (CW-305)	Clock freq	10MHz
	Target Device	Xilinx Artix-7
	Target Architecture	Xilinx 7 Series

Table 3. Hyperparameters

Field	Parameter	Value		
Common	Batch size	32		
	Number of epochs	10, 20, 30, 40, 50		
	Learning rate	1e-3, 1e-4, 1e-5		
	Activation function	ReLU		
	Loss function	Cross-entropy		
	Optimization	Adam		
MLP	Hidden layers	256	128-256	64-128-256
CNN	Convolution layer	32	16-32	8-16-32
	Kernel size	9		
	Pooling	Average pooling		
	Pooling size	2		
	Dense layer	64-8		

00,000개씩 10개로 나누어 프로파일링 공격을 10번 독립적으로 수행하여 성능 지표를 계산했다.

4.2 중간값별 학습 지표 분석

Target 1과 target 2의 소비전력에 대한 중간값별 학습 지표를 각각 Fig. 3.과 Fig. 4.에 도식했다. 그림의 가로축은 에포크, 세로축은 손실 혹은 정확도를 의미한다. 중간값별 대푯값을 표현하기 위해 18개의 신경망 중 최종 검증 손실이 가장 작은 모델의 학습 지표를 도식했다. Table 4.는 각 중간값별 마지막 epoch의 검증 정확도를 나타낸 것이다.

Target 1의 소비전력을 학습시킨 결과, 하드웨어 소비전력 특성을 반영한 전통적인 중간값인 10라운드 입출력 차분이 다른 중간값들에 비해 손실은 작고, 정확도는 높은 것을 볼 수 있다. 중간값으로 가능한 256개의 값 중 1개를 임의로 선택했을 때의 확률은 1/256로 약 0.4%이다. 3.1절에서 분석한 10라운드 입출력 차분을 제외한 나머지 중간값들은 학습 지표가 개선되나 그 폭이 미흡하여 임의로 옳은 중간값을 선택할 확률과 유사하다. 하지만, 10라운드 입출력 차분은 그로부터 약 0.2% 정도 높게 도출되며 상대적으로 학습 지표가 좋은 것을 볼 수 있다. 10라운드 SubBytes 입출력 간의 차분에 대한 성능이 좋게 도출된 이유는 SubBytes 입출력 차분으로 가능한 경우의 수가 제한적이기 때문이다. 해당 경우의 수는 256보다 작으며 균등하지 않으므로 신경망이 SubBytes 입출력 차분 1개를 임의로 선택했을 때 옳은 중간값을 선택할 확률이 다른 중간값보다 높아 정확도가 높다.

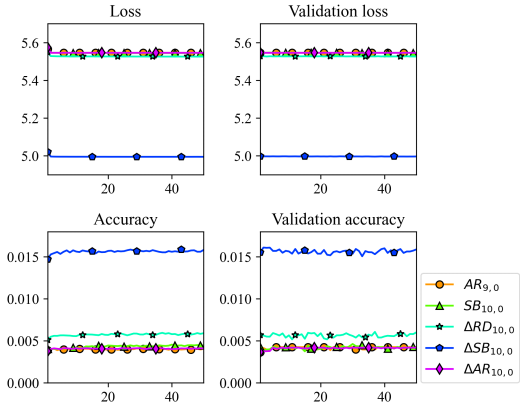


Fig. 3. Loss and accuracy for 1-50 epochs by intermediate values of target 1

Target 2의 소비전력을 학습시킨 결과, target 1의 결과와 달리 10라운드 입출력 차분의 정확도가 0.4%로 소비전력과의 상관성이 적어 학습 지표가 개선되지 않은 것을 볼 수 있다. 그에 반해 소비전력과 상관성이 상대적으로 높은 다른 중간값들은 학습 지표가 개선됨을 볼 수 있다.

Target 1과 target 2의 중간값별 검증 정확도가 가장 높은 모델에 대한 키별 평균 확률을 상자수염그

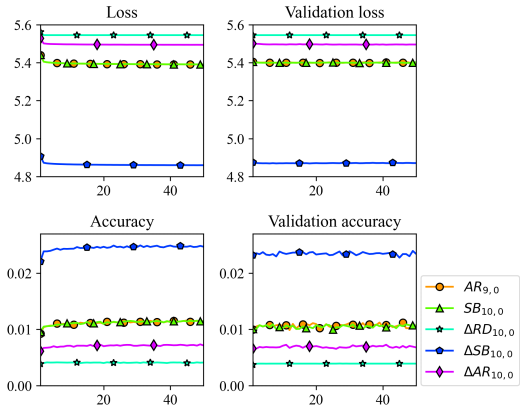


Fig. 4. Loss and accuracy for 1-50 epochs by intermediate values of target 2

Table 4. Validation accuracy of last epoch

Target	Intermediate value	Validation accuracy (%)
Target 1	$AR_{j,i}$	0.425
	$SB_{j,i}$	0.416
	$\Delta RD_{j,i}$	0.600
	$\Delta SB_{j,i}$	1.589
	$\Delta AR_{j,i}$	0.418
Target 2	$AR_{j,i}$	1.038
	$SB_{j,i}$	1.072
	$\Delta RD_{j,i}$	0.391
	$\Delta SB_{j,i}$	2.343
	$\Delta AR_{j,i}$	0.679

림으로 도식한 것이 Fig. 5., Fig. 6.과 같다. 즉, 수식 (2)의 표기에서 후보키 k에 따른 $\frac{1}{n}\times\sum_{t=0}^{n-1}s_{t,k}$의 분포를 나타낸 것이다. 상자는 제 1사분위수와 제 3사분위수 사이의 범위를 의미하며, 원은 이상치를 의미한다. 빨간색 선은 옳은 키의 확률이며 상자 안에 주황색 선은 중앙값을 의미한다. Fig. 5.에서 target 1의 구조적 특성을 반영한 중간값인 10라운드 입출력 차분만 옳은 키 확률과 다른 키 확률이 균등하지 않은 것을 볼 수 있다. Fig. 3.에서 SubBytes 입출력 차분의 경우의 수가 256보다 적어 다른 중간값들에 비해 높은 정확도가 도출됐지만 Fig. 5.에서 옳은 키의 확률과 다른 키의 확률이 균등한 것을 보인다. 이는 3.2절에서 설명한 것과 같이 모델이 데이터와 중간값의 관계를 학습하지 못한 것이며,

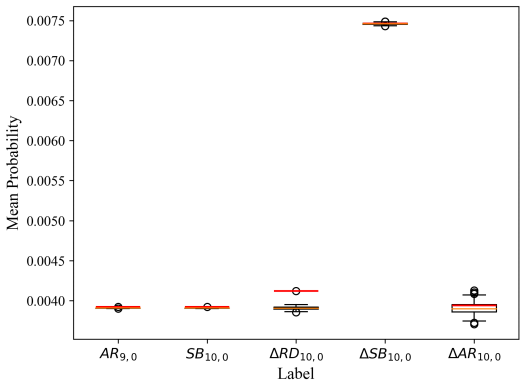


Fig. 5. Average probability by key of target 1

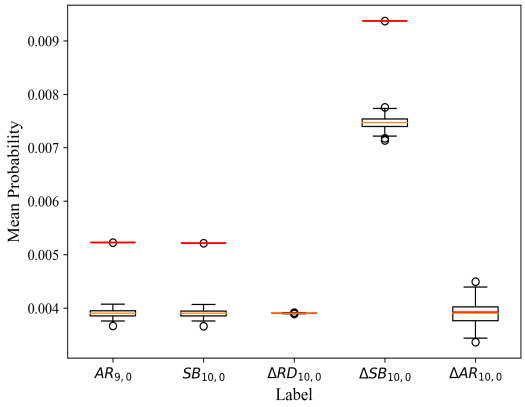


Fig. 6. Average probability by key of target 2

target 1 분석에 부적합한 중간값임을 의미한다.

Fig. 6.에서 10라운드 입출력 차분은 데이터와 목푯값의 관계가 가장 불명확하기 때문에 모델이 임의의 값을 출력하여 모든 키별 평균 확률이 균등하다. 이는 제안한 분석 방법에서 분석 대상 하드웨어에 적합하지 않은 중간값은 비밀키 선정에 미미한 영향을 미치고, 적합한 중간값만 주요한 영향을 미침을 실증한다.

4.3 중간값별 성능 지표 분석

Target 1의 중간값별 게싱 엔트로피, 비밀키 분석 성공률, 최소 분석 파형 수를 Fig. 7.과 Fig. 8.에 나타냈다. 중간값별 대푯값을 표현하기 위해 실험한 신경망 중 최종 최소 분석 파형 수가 가장 작은 모델의 결과를 도식했다. 분석 결과, Fig. 7.과 Fig. 8.에서 구현물의 특성을 반영한 중간값인 10라운드 입출력 차분의 성능이 가장 좋았고, 이외에 부채널 정보와 상관성이 적은 중간값들은 성능이 좋지 않았다.

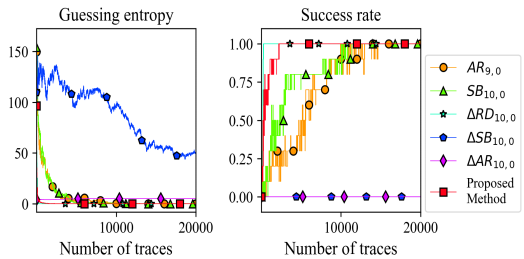


Fig. 7. Guessing entropy and success rate of target 1

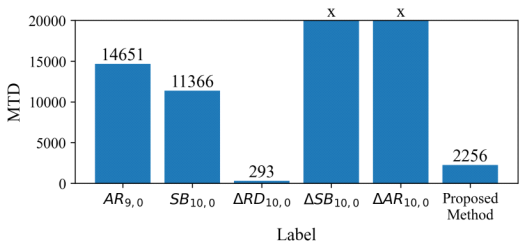


Fig. 8. Minimum number of traces to disclosure of target 1

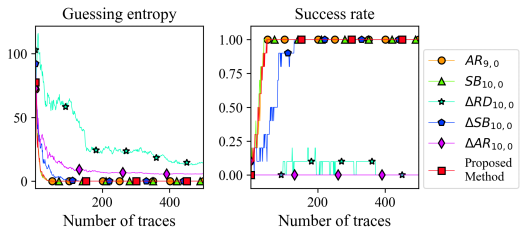


Fig. 9. Guessing entropy and success rate of target 2

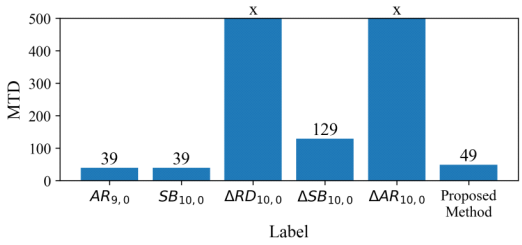


Fig. 10. Minimum number of traces to disclosure of target 2

Fig. 9.와 Fig. 10.에서 target 2는 매 라운드의 SubBytes 연산 과정을 저장해 10라운드 입출력 차분의 성능이 좋지 않았다. 이는 3.1절에서 기술한 것과 같이 구현물의 구조와 관계성이 적은 중간값은 딥러닝 분석의 성능이 좋지 않음을 실증한다. 기존 블랙박스 환경에서 활용했던 중간값인 SubBytes 입력도 활용할 수 있지만, 특정 구현에서 굉장히 낮은 분석 성능을 보였다. 반면, 제안한 방안은 두 구현물 모두 분석 성공했으며, 최적의 중간값보다는 분석 성능이 낮지만 범용적으로 활용할 수 있었다.

두 구현물 모두 각 하드웨어의 소비전력 특성과 상관성이 높은 중간값의 성능이 우수했다. 그러나 이는 하드웨어 구조가 기지인 경우에만 활용할 수 있다. 제안한 방안은 하드웨어 구조에 적합한 중간값만 비밀키 선정에 주요한 역할을 하므로 두 구현물 모두에서 높은 성능을 가졌다. 따라서 제안한 방안은 블랙박스 환경에서 범용적으로 활용할 수 있다.

V. 결 론

본 논문에서는 블랙박스 환경에서 하드웨어 구현물 대상 딥러닝 분석 시 범용적으로 활용할 수 있는 분석 방안을 제안했다. 하드웨어 분석을 위해서는 구현에 적합한 중간값을 선정해야 한다. 이는 하드웨어 구조가 기지인 화이트박스 환경에서는 가능하나, 블랙박스 환경에서는 어렵다. 본 논문에서는 속도와 면적을 각각 고려하여 구현된 대표적인 하드웨어 구현물 두 종을 효과적으로 분석할 수 있는 중간값을 탐색한 후 이를 모두 조합하는 방안을 제안했다. 제안한 방안은 하드웨어 특성에 적합한 중간값만 비밀키 선정에 주요한 영향을 미치므로 범용적으로 활용할 수 있다. 검증 실험 결과, 제안한 방안은 일반적인 하드웨어 구현물의 다양한 중간값에 대한 결과를 포함하고 있어 면적과 속도를 고려한 각 하드웨어 구현물에 대해 모두 분석에 성공한 것을 보였다.

본 연구에서는 FPGA에 구현된 AES를 대상으로 제안한 방안의 검증 실험을 수행했다. 향후 연구에서는 다양한 암호 알고리즘을 FPGA와 ASIC (Application-Specific Integrated Circuit)에 구현하여 검증 실험하겠다.

References

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," Advances in Cryptology, CRYPTO'99, LNCS 1666, pp. 388-397, Aug. 1999.

[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, Aug. 2004.

[3] S. Mangard, E. Oswald, and T. Popp, Power analysis attacks: Revealing the secrets of smart cards, Springer New York, USA, pp. 38-43, 2007.

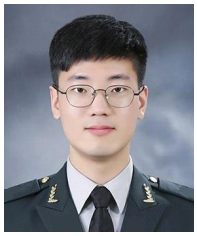
[4] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networ

ks are universal approximators," NeuralNetworks, vol. 2, no. 5, pp. 359-366, Mar. 1989.
[5] H. Wang and E. Dubrova, "Tandem deep learning side-channel attack on FPGA implementation of AES," SN Computer Science, vol. 2, no. 373, pp. 1-12, Jul. 2021.
[6] Y.S. Won and S. Bhasin, "On use of deep learning for side channel evaluation of black box hardware AES engine," Proceedings of 7th EAI International Conference on Industrial Networks and Intelligent Systems, pp. 185-194, Apr. 2021.
[7] Z. Martinasek and V. Zeman, "Innovative method of the power analysis," Radioengineering, vol. 22, no. 2, pp. 586-594, Jun. 2013.
[8] G. Hospodar, B. Gierlichs, E. D. Mulder, I. Verbaushede, and J. Vandewalle, "Machine learning in side-channel analysis: a first study," Journal of Cryptographic Engineering, vol. 1, no. 4 pp. 293-302, Oct. 2011.
[9] B. Köpf, D. Basin, "An information theoretic model for adaptive side-channel attacks," Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 286-296, Oct. 2007.
[10] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," International Conference on Security, Privacy, and Applied Cryptography Engineering, LNCS 10076, pp. 3-26, Dec. 2016.
[11] S. Picek, I.P. Samiotis, A. Heuser, J. Kim, S. Bhasin, and A. Legay, "On the performance of convolutional neural networks for side-channel analysis," International Conference on Security, Privacy, and Applied Cryptography Engineering, LNCS 11348, pp. 157-176, Dec. 2018.
[12] H. Delfs and H. Knebl, Introduction to cryptography, 3rd Ed., Springer, Berlin, Heidelberg, pp. 11-31, Aug. 2015.
[13] E.N.C. Mui, "Practical Implementation of Rijndael S-Box Using Combinational Logic", Custom R&D Engineer Texco Enterprise Pvt.Ltd., 2007.
[14] M. Shanthini, P. Rajasekar, and H. Mangalam, "Design of low power S-Box in Architecture Level using GF", International journal of engineering research and general science, vol. 2, no. 3, pp. 268-276, May. 2014.
[15] C. Zhang, S. Bengio, M. Hardt, B. Recht and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," Communications of the ACM vol. 64. no. 3, pp. 107-115, Mar. 2021.
[16] GitHub, "ProjectVault github," https://github.com/ProjectVault/orp/tree/master/hardware/mselSoC/src/systems/gophyte/rtl/verilog/crypto_aes/rtl/verilog, Jun. 2024.
[17] GitHub, "verilog aes github," https://github.com/secworks/aes/tree/master/src/rtl, Jul. 2024.

〈저자소개〉

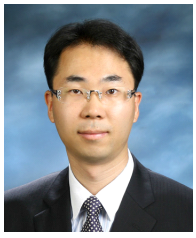
최 기 훈 (Ki-Hoon Choi) 학생회원
2019년 3월~현재: 국민대학교 정보보안암호수학과 학사과정
〈관심분야〉 부채널 분석 및 대응법 설계, 딥러닝

오 충 연 (Chung-Yeon Oh) 학생회원
2020년 3월~현재: 국민대학교 정보보안암호수학과 학사과정
〈관심분야〉 부채널 분석 및 대응법 설계, 딥러닝



김 주 환 (Ju-Hwan Kim) 학생회원
2021년 2월: 국민대학교 수학과 학사
2023년 9월~현재: 국민대학교 금융정보보안학과 석박사통합과정
〈관심분야〉 부채널 분석 및 대응법 설계, 딥러닝, 오류 주입 공격

박 혜 진 (Hye-Jin Park) 학생회원
2023년 2월: 국민대학교 정보보안암호수학과 학사
2023년 3월~현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 부채널 분석 및 대응법 설계, 딥러닝, 오류 주입 공격



한 동 국 (Dong-Guk Han) 종신회원
1999년 2월: 고려대학교 수학과 학사
2002년 2월: 고려대학교 수학과 이학석사
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 정보보호연구단 선임연구원
2009년 3월~현재: 국민대학교 정보보안암호수학과 정교수
〈관심분야〉 공개키 암호시스템 안전성 분석 및 고속 구현, 부채널 분석 및 대응법 설계, IoT 정보보호 기술