Non-Profiled Deep Learning-based Side-Channel attacks with Sensitivity Analysis

https://youtu.be/bcTbzgfnBPE

HSU 한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

Non-Profiled Deep Learning-based Side-Channel attacks

Non Profiled attacks

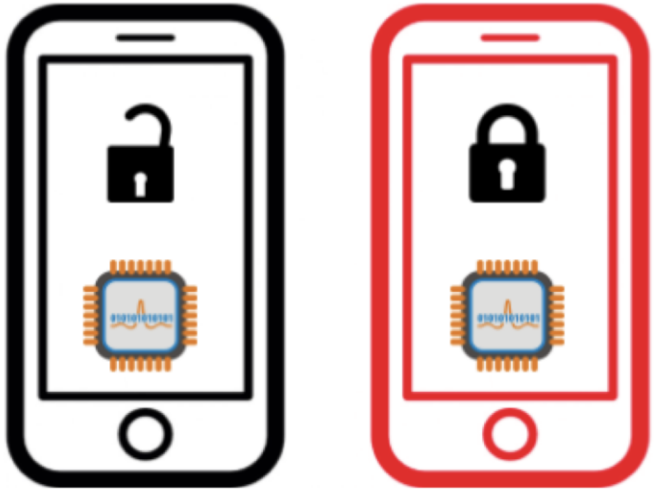
Target device
(closed)

- Differential Power Analysis (DPA)
- Correlation Power Analysis (CPA)
- Mutual Information Analysis (MIA)

Profiled attacks

Profiling device
(open)

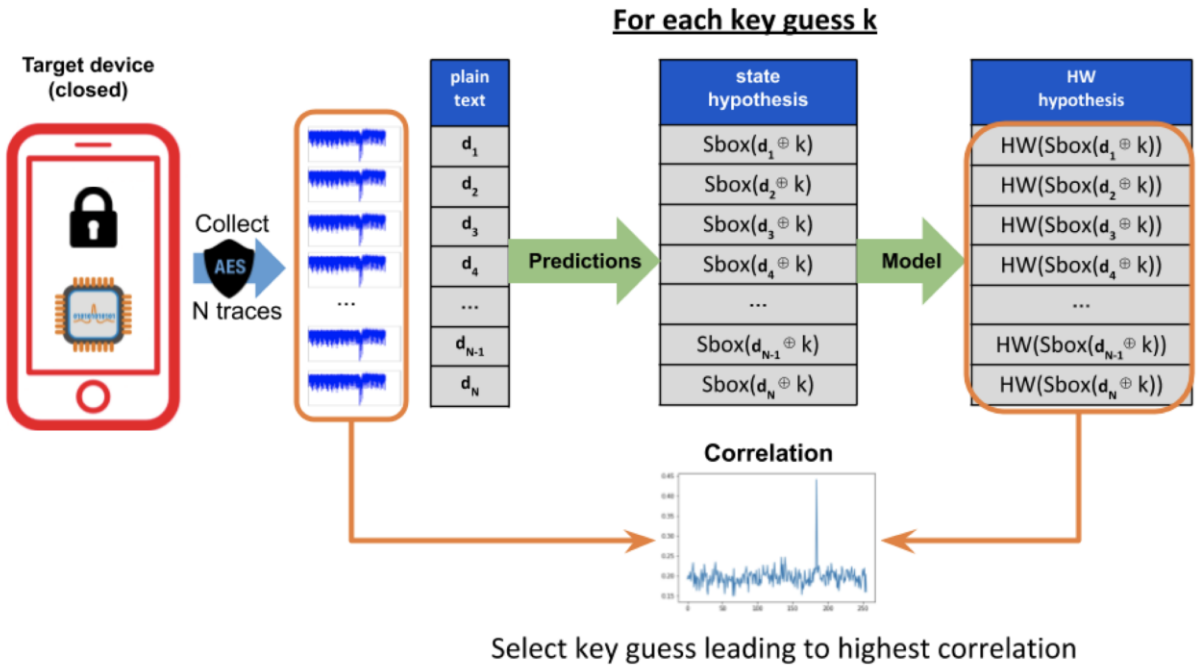
Target device
(closed)



- Template attacks
- Support Vector Machine
- Random Forests
- Deep Learning

Differential Deep Learning Analysis (DDLA)

- Non-Profiled 컨텍스트에서 딥 러닝 기술을 적용하는 새로운 방법



Non-Profiled Deep Learning Side-Channel attacks

- DDLA (Differential Deep Learning Analysis)
  - 프로파일링되지 않은 컨텍스트에서 딥 러닝 기술을 적용하는 새로운 공격 방법

Algorithm 1 Differential Deep Learning Analysis (DDLA)

Inputs: N traces $(T_i)_{1 \leq i \leq N}$ and corresponding plaintexts $(d_i)_{1 \leq i \leq N}$. A network `Net` and number of epochs n_e.

1: Set training data as $X = (T_i)_{1 \leq i \leq N}$.
2: **for** $k \in \mathcal{K}$ **do**
3: Re-initialize trainable parameters of `Net`.
4: Compute the series of hypothetical values $(H_{i,k})_{1 \leq i \leq N}$.
5: Set training labels as $Y_k = (H_{i,k})_{1 \leq i \leq N}$.
6: Perform Deep Learning training: $\mathbf{DL}(\texttt{Net}, X, Y_k, n_e)$.
7: **end for**
8: **return** key k which leads to the best DL training metrics

- Network architecture
  - MLP 및 CNN 아키텍처를 사용하는 DDLA의 두 가지 변형에 중점
  - 동기화 된 트레이스를 대상으로 할 때 MLP를 사용(MLP-DDLA).
  - 비 동기화 된 트레이스를 대상으로 할 때 CNN을 사용 (CNN-DDLA).

- 다른 추측보다 올바른 키 값에 대해 더 효율적인 훈련을 관찰 할 수 있어야함

- 최상의 교육 지표로 이어지는 키를 선택하여 다른 후보와 올바른 키 값을 구별 해야함

Metrics

- Sensitivity analysis based on MLP first layer weights
- Sensitivity analysis based on network inputs
- Loss and accuracy metrics
- 시뮬레이션 데이터를 사용하여 알고리즘 정의 된대로 공격을 수행하고 메트릭 관찰

  N = 5,000 개의 시뮬레이션 된 트레이스를 생성

  n = 50 samples per trace.

  Sbox 누설은 time sample t = 25에서 설정

  Sbox (di⊕k ∗) + N (0,1)로 정의, di는 알려진 무작위 바이트와 k ∗ a 고정 키 바이트

  N (0,1) 평균 μ = 0 및 표준 편차 σ = 1.의 가우시안 노이즈

Sensitivity analysis based on MLP first layer weights

- DDLA 훈련 중 첫 번째 계층 가중치와 관련된 네트워크의 민감도 분석을 기반
- MLP와 같은 아키텍처에서만 의미

$$S_{weights}[t] = \sum_{j=1}^{R} |\nabla W_{t,j}|$$

$$\nabla W_{i,j} = \frac{\partial \mathcal{L}}{\partial W_{i,j}}$$

- 트레이스의 샘플 R
- 첫 번째 은닉층 가중치 행렬 W
- Wi, j에 대한 손실의 미분의 절대값 합산
- 해당 가중치에 대한 손실의 감도를 측정
- 미분의 절대 값이 높을수록 해당 가중치가 손실 최소화에 더 많이 기여

Sensitivity analysis based on MLP first layer weights

- 키 추측에 대해 훈련이 끝날 때 미분의 누적 합계를 비교
- 올바른 키 추측이 Sbox 누출 위치에 해당하는 정확히 25에서 더 높은 값

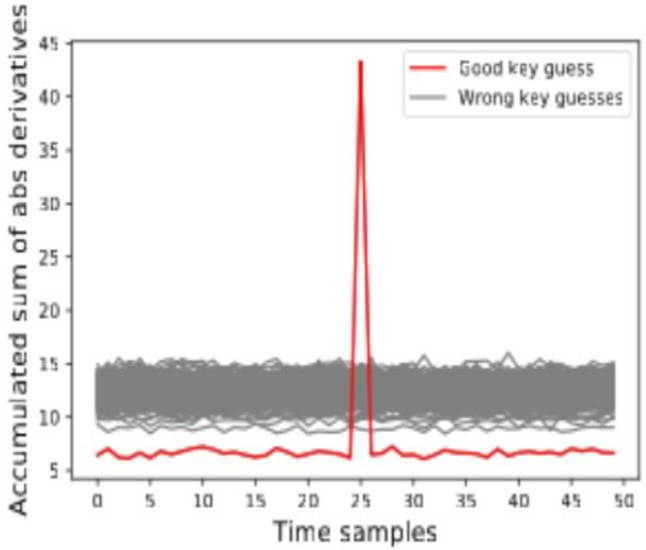


Figure 4: Sum of absolute derivatives accumulated over 250 iterations of SGD (50 epochs)

(Stochastic Gradient Descent)

Sensitivity analysis based on network inputs

입력(trace T)에 대한 손실의 편도 함수 사용

모든 네트워크 아키텍처에 적용 가능

$$\nabla T_{i,j} = \frac{\partial \mathcal{L}_{T_i}}{\partial x_j}, \quad \text{for } i \in \{1, ..., N\} \text{ and } j \in \{1, ..., n\},$$

$$S_{input}[t] = \sum_{i=1}^{N} |\nabla T_{i,t}|, \quad \text{for } t \in \{1, ..., n\}.$$

$$S_{input}[t] = \sum_{i=1}^{N} (\nabla T_{i,t} \times T_{i,t}),$$

일반적으로 더 나은 결과

Sensitivity analysis based on network inputs

- MLPsim 및 CNNsim 아키텍처를 모두 사용하여 적용했습니다
- 이전과 유사한 결과를 관찰, 좋은 키 추측은 훨씬 더 높은 민감도

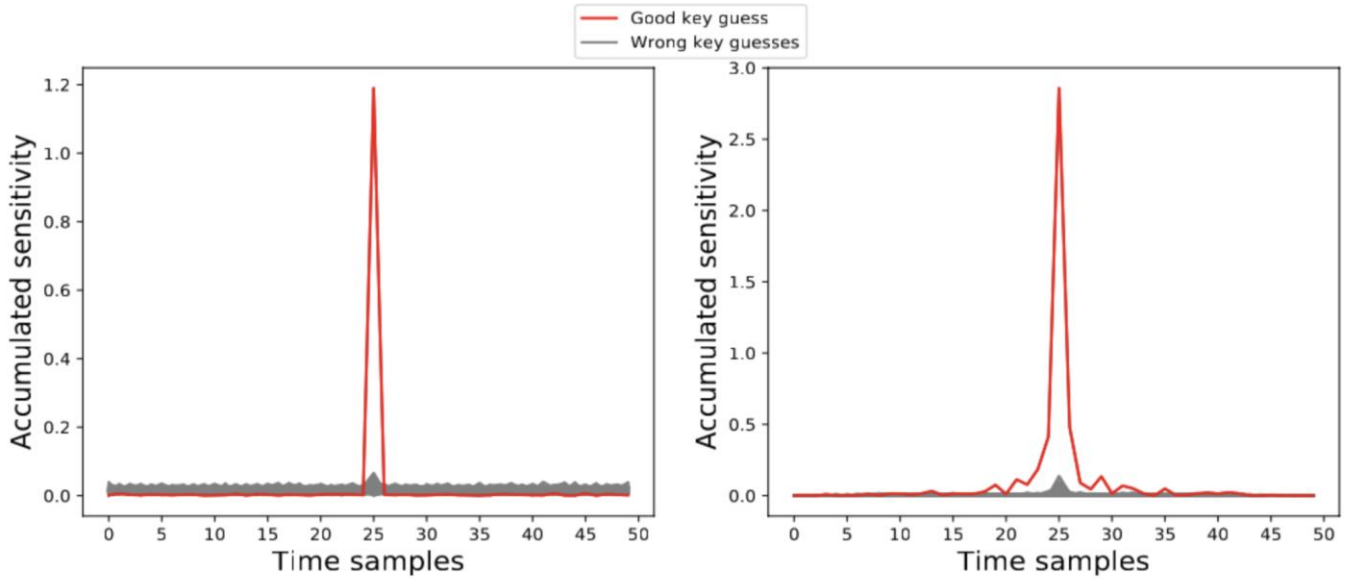


Figure 5: Inputs-based sensitivity accumulated over 50 epochs for all key guesses. Left: with MLP_{sim} network. Right: with CNN_{sim} network.

Loss and accuracy metrics

- 시뮬레이션 데이터 세트를 사용
  DDLA 공격을 수행 할 때 50epochs에서 모든 키 게스트에 대해 얻은 손실과 정확도

- 가장 높은 정확도 또는 가장 낮은 손실 값으로 이어지는 추측을 선택하여 올바른 키를 표시 가능

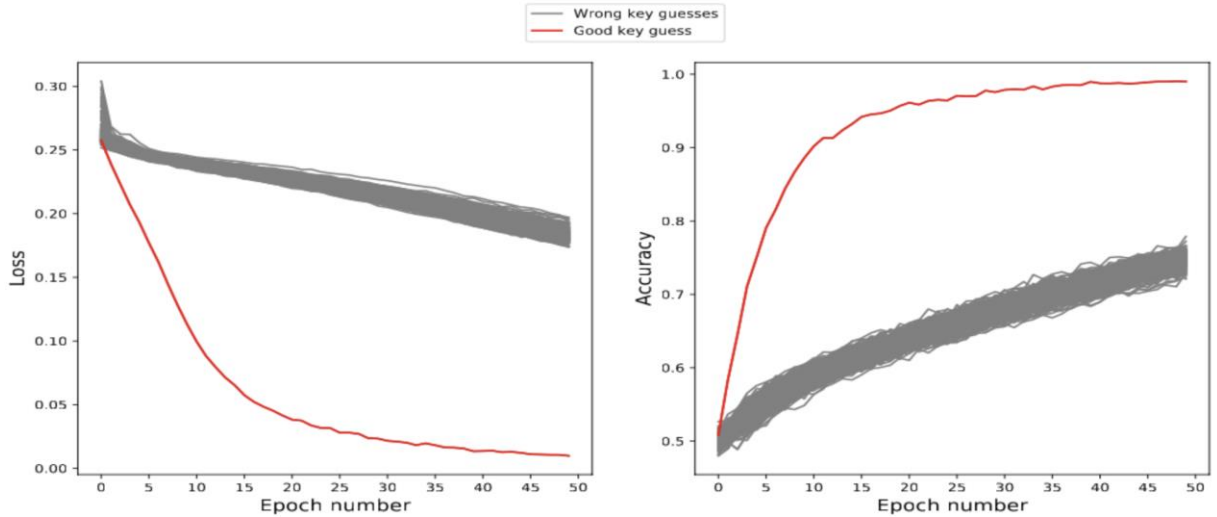


Figure 6: Loss (left) and accuracy (right) over the training epochs for all the key guesses when applying MLP-DDLA.

Labels

- identity labeling (Sbox(di⊕k))
  - 항상 유사한 정확도로 인해 올바른 키 값을 구별 불가
- Hamming Weight labeling
- Binary labeling

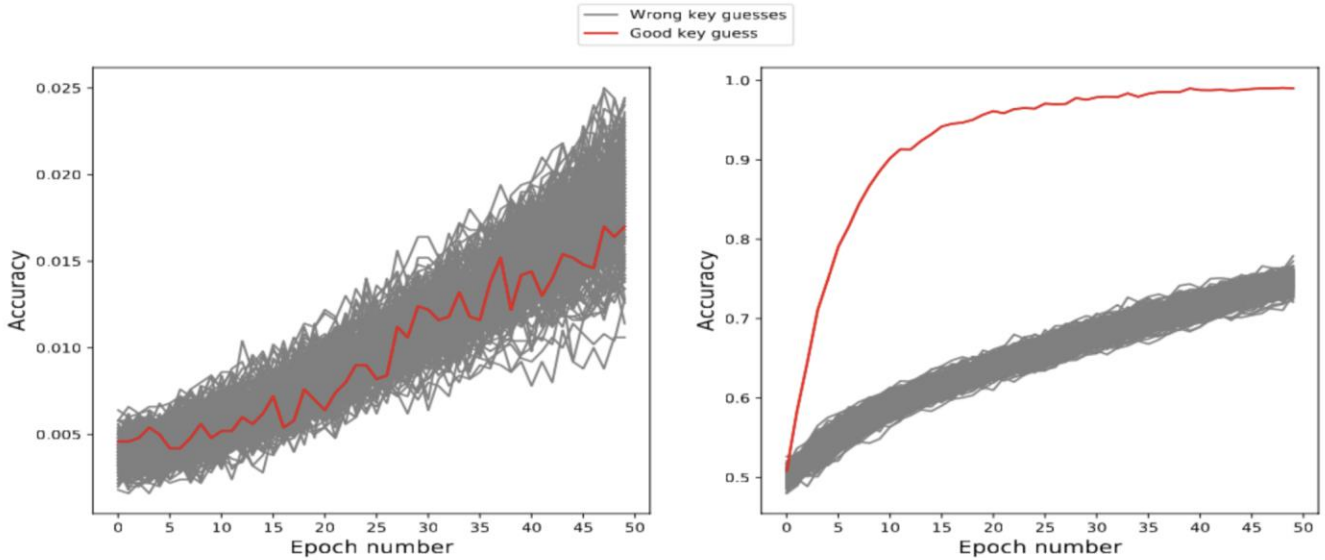


Figure 7: MLP-DDLA accuracies using two different labeling methods. Left: Identity labeling. Right: Binary labeling (MSB).

High-Order DDLA

- Profiled 및 Non-Profiled 공격으로부터 암호화 구현을 보호하기위한 일반적인 대책
  민감한 중간 값을 마스크로 숨기는 것

$$S = Sbox(d \oplus k) \oplus m_1 \oplus \cdots \oplus m_s$$

- 전처리 단계 : 절대값의 차이 같은 조합 함수를 사용하여 마스크값과 마스크 된 값의 누출 값을 결합
- 공격 단계 : 결합 된 누출값을 통해 정보를 추출 , Pearson 's Correlation 사용

- 마스크의 위치와 마스킹 된 값을 알고 있는 경우 : 두 누출 위치를 함께 결합
- 마스크의 위치와 마스킹 된 값 누출을 알 수 없는 경우 : 추적에서 가능한 모든 포인트 쌍을 함께 결합
  - ➢ 트레이스가 크면 이러한 처리가 너무 복잡하고 실용적이지 않을 수 있음

Experiments

- 비 동기화 된 트레이스에 대해 프로파일 링 되지 않은 컨텍스트에서 CNN을 사용하고 CPA와 비교

- DDLA가 블랙 박스에서 마스크 된 구현을 중단하고 trace에서 마스크 위치를 표시하는 방법

- ChipWhisperer-Lite (CW) 및 공용 데이터베이스 ASCAD 에서 수집 된 시뮬레이션 된 trace사용
- CW를 통해 Atmel XMEGA128 칩에서 실행되는 구현의 전력 trace을 수집
- ASCAD의 trace은 8 비트 ATMega8515 보드에서 수집

- MLP, CNN 사용
- CNN 는 크기가 32 인 필터 4 개와 크기 16 인 필터 4 개로 구성된 컨볼루션 레이어 2 개로 구성
- MLP 는 20 개와 10 개의 뉴런으로 구성된 두 개의 은닉 계층으로 구성

동기화 된 트레이스

- 3,000 개의 트레이스로 성공 민감도 메트릭은 CPA와 동일한 누출 위치를 보임

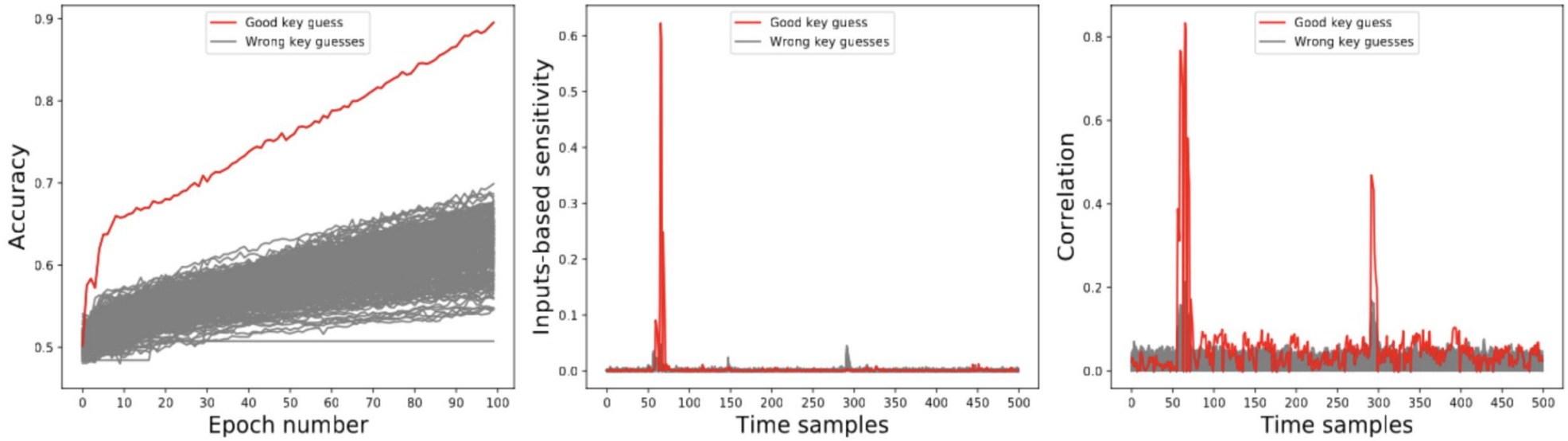


Figure 8: Attack on CW unprotected implementation without de-synchronization. Left: MLP-DDLA accuracies. Center: MLP-DDLA inputs-based sensitivity Right: CPA.

비 동기화 된 트레이스

- CPA와 MLP-DDLA가 트레이스의 동기화 해제로 키 복구 실패
- CNN-DDLA는 성공
- CNN의 번역 불변 속성이 비 동기화 된 트레이스에 대한 비 프로파일 공격 중에 성공적으로 사용될 수 있음
- 추적을 완벽하게 재 동기화 할 수 없는 경우

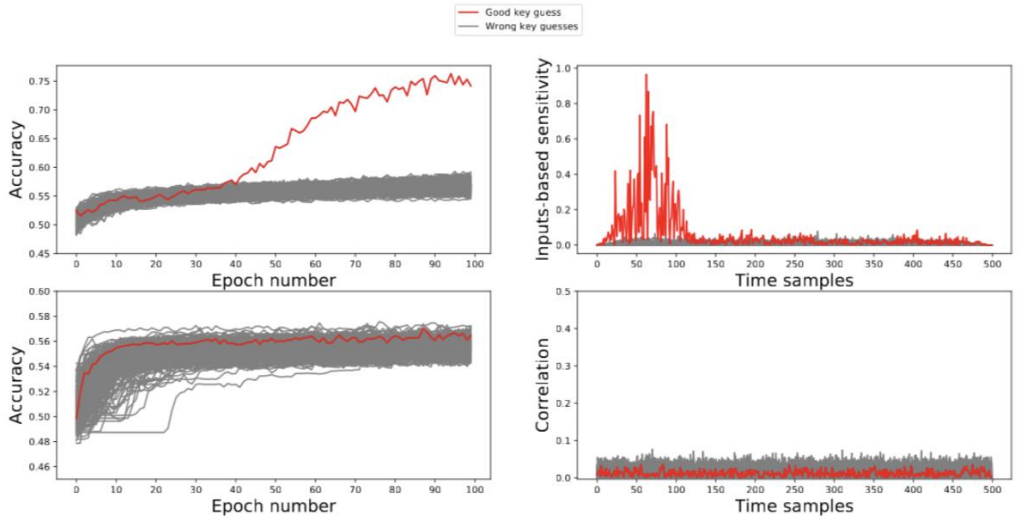


Figure 9: Attack on CW unprotected implementation with de-synchronization. Top-left: CNN-DDLA accuracies. Top-right: CNN-DDLA inputs-based sensitivity. Bottom-left: MLP-DDLA accuracies. Bottom-right: CPA.

Second order DDLA on ASCAD

- 20,000 개의 트레이스
- epoch 후에 DDLA 공격 성공
- 민감도 분석 값은 CPA 리버스 엔지니어링을 통해 얻은 마스크 및 마스킹 된 Sbox 영역과 일치
- **DDLA에 의해 강조 표시된 위치는 마스크 또는 키 값에 대한 지식없이 획득**

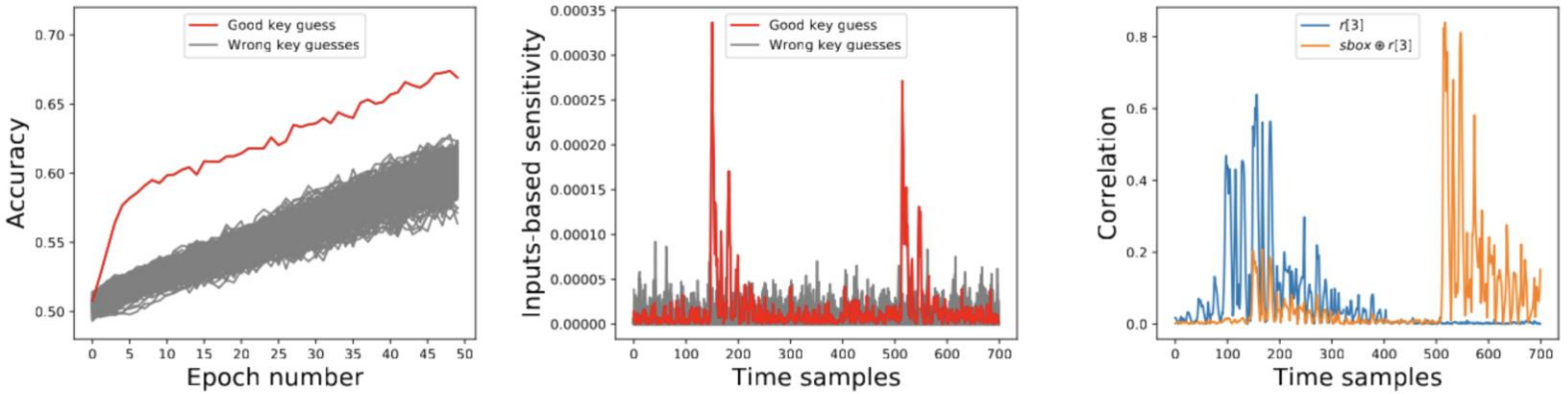


Figure 11: MLP-DDLA attack on ASCAD. Left: Accuracy. Center: Inputs-based sensitivity. Right: CPA reverse engineering.

Third order DDLA on ChipWhisperer

- 50,000 트레이스
- 약 20 epoch에 DDLA 공격 성공

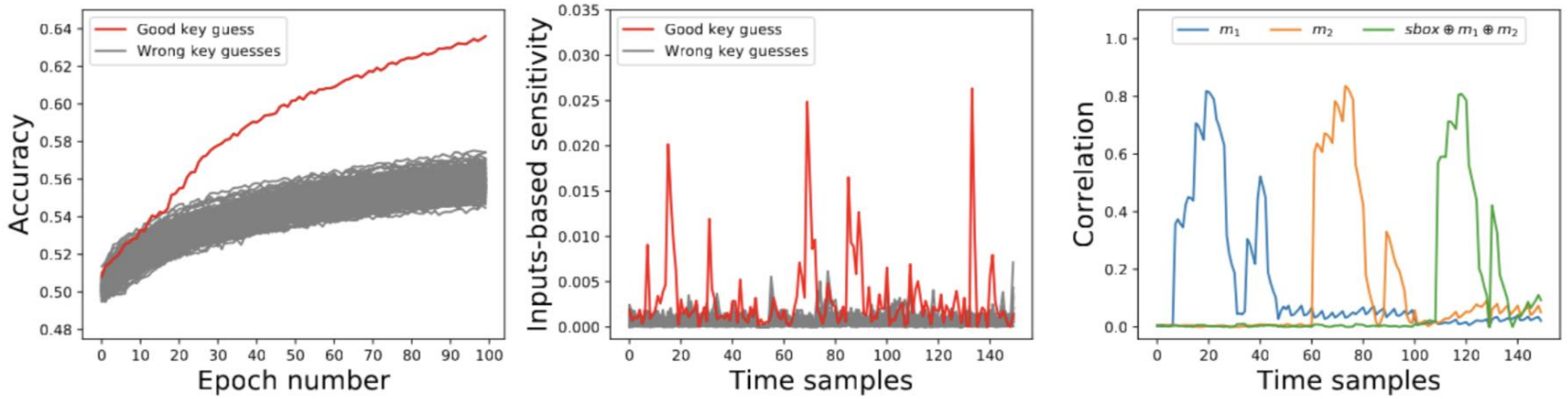


Figure 12: DDLA on CW 2-masks protected implementation. Left: Accuracy. Center: Inputs-based sensitivity. Right: CPA-based reverse engineering.

Complexity

- 올바른 키 값을 표시하는 데 몇 에포크 만 필요
- Epoch 수를 줄임으로써 이러한 공격을 더 빠르게 수행
- 실험에 사용 된 신경망은 공격의 복잡성을 제한하기 위해 의도적으로 작게 유지
- 이 접근 방식은 공격에 사용되는 총 epoch 수를 제어 할 수 있으므로 복잡성을 줄이는 데 사용할 수 있
- 사용 된 아키텍처는 확실히 최적이 아니며 더 복잡한 네트워크는 더 나은 결과를 가져올 수 있

Table 1: Execution times comparison for 1 key byte attacks.

Target	Architecture	Nb traces	Nb samples	Nb epochs	Time
CW no mask	MLP_{exp}	3,000	500	100	4min20s
CW no mask	CNN_{exp}	3,000	500	100	33min17s
CW 2-masks	MLP_{exp}	50,000	150	100	1h03min16s
ASCAD	MLP_{exp}	20,000	700	50	14min12s
ASCAD	MLP_{exp}	20,000	700	5	1min54s

Conclusion

- 프로파일 링되지 않은 컨텍스트에서 딥 러닝 기술을 적용하기위한 새로운 부 채널 공격 방법 인 DDLA (Differential Deep Learning Analysis)를 소개
- 딥 러닝 훈련을 사용하여 비밀 키 값을 공개
- 비밀 키와 추적의 누출 및 마스크 위치와 같은 관심 지점을 모두 표시 할 수있는 민감도 분석을 기반으로하는 메트릭을 도입
- 구현 된 보호에 대한 누수 조합 전처리 또는 가정없이 블랙 박스에서 마스킹 된 구현을 중단하는 데 사용가능
- 보호되지 않은 구현과 마스킹 된 구현 모두에 적용될 수 있음

Q & A