A Comprehensive Study of Deep Learning for Side-Channel Analysis

Loïc Masure\textsuperscript{1,3}, Cécile Dumas\textsuperscript{1}, Emmanuel Prouff\textsuperscript{2,3}

\textsuperscript{1}Univ. Grenoble Alpes, CEA, LETI, DSYS, CESTI, F-38000 Grenoble
loic.masure@cea.fr
\textsuperscript{2}ANSSI, France
\textsuperscript{3}Sorbonne Université, UPMC Univ Paris 06, POLSYS, UMR 7606, LIP6, F-75005, Paris, France

17/09/2020, CHES
Outline

1. Context

2. SCA Optimization Problem versus Deep Learning Based SCA

3. NLL Minimization is PI Maximization

4. Simulation results

5. Experimental results
Who am I

- PhD student, studying Deep Learning (DL) for Side-Channel Analysis (SCA)

- Conceives a component
- Evaluates Security Claims
- Delivers a Security Certification
- Commercialises the certified product

 Developers
- Loïc
- Cécile
- Emmanuel

French Certification Scheme

Institutions
- ITSEF
- ANSSI
What is SCA?

```
LOAD X;       LOAD B;          MV B;             ...
```

Encryption Sensitive operation

```
Z = C(P, K)
```

Profiling Attack

Attack using open samples similar to the target device – same code, same chip, etc. – with full knowledge of the secret key.

Two steps:

1. Profiling phase: $P$, $K$ known $\Rightarrow Z$ known, $X$ acquired on an open sample
2. Attack phase: $P$ known, $X$ acquired on the target device, $K$ guessed

17/09/2020, CHES | Loïc Masure, Cécile Dumas, Emmanuel Prouff | 4/18
What is SCA?

Encryption Sensitive operation

\[
\text{LOAD X ; } \quad \text{LOAD B ; } \quad \text{MV B ;} \quad \ldots \quad \text{Plaintext P} \quad \text{Secret K} \\
\text{Measure trace X} \quad \text{Z = C(P, K)}
\]

Profiling Attack

Attack using open samples similar to the target device – same code, same chip, etc.

Two steps:

- Profiling phase: \( P, K \) known \( \Rightarrow \) \( Z \) known, \( X \) acquired on an open sample
- Attack phase: \( P \) known, \( X \) acquired on the target device, \( K \) guessed

17/09/2020, CHES | Loïc Masure, Cécile Dumas, Emmanuel Prouff | 4/18
What is SCA?

Profiling Attack

Attack using open samples similar to the target device – same code, same chip, etc. – with full knowledge of the secret key

Two steps:
- Profiling phase: \( P, K \) known \( \Rightarrow \) \( Z \) known, \( X \) acquired on an open sample
- Attack phase: \( P \) known, \( X \) acquired on the target device, \( K \) guessed
Outline

1. Context

2. SCA Optimization Problem versus Deep Learning Based SCA

3. NLL Minimization is PI Maximization

4. Simulation results

5. Experimental results
Profiling Attacks

Key Recovery (i.e. attack step)

Given $N_a$ attack traces $x_i$ with plaintext $p_i$, calculate scores $y_i = F(x_i)$

$y_0$

$Z_i = C(p_i, k^*)$

$0 1 \cdots 0 1 \cdots K$
Profiling Attacks

Key Recovery (i.e. attack step)

Given \( N_a \) attack traces \( \mathbf{x}_i \) with plaintext \( p_i \), calculate scores \( y_i = F(x_i) \)

\[
0 \quad 1 \quad \cdots \quad y_0 \quad y_1
\]

\[
Z_i = C(p_i, k^*)
\]

\[
0 \quad 1 \quad \cdots \quad K
\]
Profiling Attacks

Key Recovery (i.e. attack step)

Given $N_a$ attack traces $x_i$ with plaintext $p_i$, calculate scores $y_i = F(x_i)$

$$y_0 \quad y_1 \quad y_2$$

$$0 \quad 1 \quad \ldots$$

$$Z_i = C(p_i, k^*)$$

$$0 \quad 1 \quad \ldots \quad K$$
Profiling Attacks

Key Recovery (i.e. attack step)

Given $N_a$ attack traces $x_i$ with plaintext $p_i$, calculate scores $y_i = F(x_i)$

$$y_0, y_1, y_2$$

$$Z_i = C(p_i, k^*)$$

$$0 1 \cdots \hat{k} K$$
Profiling Attacks

Key Recovery (i.e. attack step)

Given $N_a$ attack traces $x_i$ with plaintext $p_i$, calculate scores $y_i = F(x_i)$

Goal: find $F$ that minimizes $N_a$ s.t. $\hat{k} = k^*$ with probability $\geq \beta$ (e.g. 0.9)
Profiling Attacks

Key Recovery (i.e. attack step)

Given $N_a$ attack traces $x_i$ with plaintext $p_i$, calculate scores $y_i = F(x_i)$

Goal: find $F$ that minimizes $N_a$ s.t. $\hat{k} = k^*$ with probability $\geq \beta$ (e.g. 0.9)

Optimal model: $F^*$, with $N_a^*$ traces
Key Recovery (i.e. attack step)

Given $N_a$ attack traces $x_i$ with plaintext $p_i$, calculate scores $y_i = F(x_i)$

Goal: find $F$ that minimizes $N_a$ s.t. $\hat{k} = k^*$ with probability $\geq \beta$ (e.g. 0.9)

Optimal model: $F^*$, with $N_a^*$ traces

How to find $F^*$ $\implies$ profiling step

Requires to know the probability distribution $F^* = \Pr[Z|X]$
A Comprehensive Study of Deep Learning for Side-Channel Analysis

Profiling Attacks

Key Recovery (i.e. attack step)

Given $N_a$ attack traces $x_i$ with plaintext $p_i$, calculate scores $y_i = F(x_i)$

Goal: find $F$ that minimizes $N_a$ s.t. $\hat{k} = k^*$ with probability $\geq \beta$ (e.g. 0.9)
Optimal model: $F^*$, with $N_a^*$ traces

How to find $F^*$ $\implies$ profiling step

Requires to know the probability distribution $F^* = \Pr[Z|X]$
Reality: unknown for the evaluator/attacker. Estimation with parametric models $F(., \theta)$:
Deep Learning (DL) based SCA is a hot topic currently

Recent milestones about its effectiveness: more robust against counter-measures like masking [MPP16], jitter (misalignment) [CDP17], whether on software or FPGA [Kim+19]
Deep Learning (DL) based SCA is a hot topic currently

Recent milestones about its effectiveness: more robust against counter-measures like masking [MPP16], jitter (misalignment) [CDP17], whether on software or FPGA [Kim+19]

Training a Neural Network

\[ z = C(p, k^*) \]

\[ F(x, \theta) \]

\[ \mathcal{L}(y, z) \]

Parameters \( \theta \)
Deep Learning (DL) based SCA is a hot topic currently

Recent milestones about its effectiveness: more robust against counter-measures like masking [MPP16], jitter (misalignment) [CDP17], whether on software or FPGA [Kim+19]

Training a Neural Network

\[ F(x, \theta) \rightarrow \mathcal{L}(y, z) \]

\( z = C(p, k^*) \)

\( \mathcal{L} \): performance metric (accuracy, recall, ...) or loss function (Mean Square Error, NLL, ...)

Parameters \( \theta \)
Open issue with Machine Learning based SCA$^1$

“How to evaluate the quality of a model during training?”

$^1$Picek et al., CHES 2019 [Pic+18]
Open issue with Machine Learning based SCA¹

“How to evaluate the quality of a model during training?”

▶ Accuracy: probability to recover the secret key with one trace

¹Picek et al., CHES 2019 [Pic+18]
Open issue with Machine Learning based SCA\(^1\)

“How to evaluate the quality of a model during training?”
- Accuracy: probability to recover the secret key with one trace

Their observations

“Accuracy does not seem to be the right performance metric in SCA”

\(^1\)Picek et al., CHES 2019 [Pic+18]
Open issue with Machine Learning based SCA

“How to evaluate the quality of a model during training?”

- Accuracy: probability to recover the secret key with one trace

Their observations

“Accuracy does not seem to be the right performance metric in SCA”

- High accuracy $\implies$ successful key recovery

---

$^1$Picek et al., CHES 2019 [Pic+18]
Open issue with Machine Learning based SCA

“How to evaluate the quality of a model during training?”
- Accuracy: probability to recover the secret key with one trace

Their observations

”Accuracy does not seem to be the right performance metric in SCA”
- High accuracy $\implies$ successful key recovery
- Low accuracy $\implies$ nothing

$^1$Picek et al., CHES 2019 [Pic+18]
Open issue with Machine Learning based SCA

“How to evaluate the quality of a model during training?”
- Accuracy: probability to recover the secret key with one trace

Their observations

“Accuracy does not seem to be the right performance metric in SCA”
- **High** accuracy $\implies$ successful key recovery
- **Low** accuracy $\implies$ nothing, problem: often happens (e.g. highly noisy leakages)

---

$^1$Picek et al., CHES 2019 [Pic+18]
Open issue with Machine Learning based SCA

“How to evaluate the quality of a model during training?”
▶ Accuracy: probability to recover the secret key with one trace

Their observations

“Accuracy does not seem to be the right performance metric in SCA”
▶ High accuracy $\implies$ successful key recovery
▶ Low accuracy $\implies$ nothing, problem: often happens (e.g. highly noisy leakages)
▶ Apparently, no other machine learning metric related to SCA metrics

---

$^{1}$Picek et al., CHES 2019 [Pic+18]
Open issue with Machine Learning based SCA

“How to evaluate the quality of a model during training?”
▶ Accuracy: probability to recover the secret key with one trace

Their observations

”Accuracy does not seem to be the right performance metric in SCA”
▶ High accuracy \(\implies\) successful key recovery
▶ Low accuracy \(\implies\) nothing, problem: often happens (e.g. highly noisy leakages)
▶ Apparently, no other machine learning metric related to SCA metrics

Accuracy: find \(\beta\) s.t. \(N^*_a = 1\)

---

\(^1\)Picek et al., CHES 2019 [Pic+18]
Open issue with Machine Learning based SCA

“How to evaluate the quality of a model during training?”
- Accuracy: probability to recover the secret key with one trace

Their observations

”Accuracy does not seem to be the right performance metric in SCA”
- High accuracy \( \implies \) successful key recovery
- Low accuracy \( \implies \) nothing, problem: often happens (e.g. highly noisy leakages)
- Apparently, no other machine learning metric related to SCA metrics

Accuracy: find \( \beta \) s.t. \( N^*_a = 1 \)

\( \neq \) SCA: fix \( \beta \) and find \( N^*_a \) instead

\(^1\)Picek et al., CHES 2019 [Pic+18]
Open issue with Machine Learning based SCA

“How to evaluate the quality of a model during training?”
- Accuracy: probability to recover the secret key with one trace

Their observations

”Accuracy does not seem to be the right performance metric in SCA”
- High accuracy $\Rightarrow$ successful key recovery
- Low accuracy $\Rightarrow$ nothing, problem: often happens (e.g. highly noisy leakages)
- Apparently, no other machine learning metric related to SCA metrics

Accuracy: find $\beta$ s.t. $N_a^* = 1$ $\neq$ SCA: fix $\beta$ and find $N_a^*$ instead

Our claim: we can accurately estimate $N_a^*$ with DL!

---

$^1$Picek et al., CHES 2019 [Pic+18]
Outline

1. Context

2. SCA Optimization Problem versus Deep Learning Based SCA

3. NLL Minimization is PI Maximization

4. Simulation results

5. Experimental results
Bridging the gap between the loss function and the SCA metric
Bridging the gap between the loss function and the SCA metric
Bridging the gap between the loss function and the SCA metric

\[ H(Z) = H(Z | X) \]

\[ \text{MI (Z; X)} \geq f(\beta) \]

\[ \text{PI (Z; X; } \theta) \leq \text{MI (Z; X)} \]

\[ f(\beta) = n - (1 - \beta) \log_2(2n - 1) + \beta \log_2(\beta) + (1 - \beta) \log_2(1 - \beta) \]
Bridging the gap between the loss function and the SCA metric

\[ H(Z) \]

\[ H(Z|X) \]
Bridging the gap between the loss function and the SCA metric

\[ \text{MI}(Z; X) \text{ and } H(Z) \text{ vs. } H(Z|X) \]

\[ \text{Training: minimization of the NLL} \text{ a.k.a. Cross Entropy} \]

\[ L(\theta) = \frac{1}{N_p} \sum_{i=1}^{N_p} -\log F(x_i, \theta)[z_i] = H(Z) - \hat{P}_I(\mathbf{Z}; \mathbf{X}; \theta) \]

\[ f(\beta) = n - (1 - \beta) \log_2(2n - 1) + \beta \log_2(\beta) + (1 - \beta) \log_2(1 - \beta) \]

\[ \text{MI}(\mathbf{Z}; \mathbf{X}) \geq f(\beta) \]

\[ \hat{P}_I(\mathbf{Z}; \mathbf{X}; \theta) \leq \text{MI}(\mathbf{Z}; \mathbf{X}) \]
A Comprehensive Study of Deep Learning for Side-Channel Analysis

Bridging the gap between the loss function and the SCA metric

\[ H(Z) \]

\[ H(Z|X) \]

\[ \text{MI}(Z; X) \geq \frac{f(\beta)}{N^*} \]

Cherisey et al. CHES 19

\[ f(\beta) = n - (1 - \beta) \log_2(2^n - 1) + \beta \log_2(\beta) + (1 - \beta) \log_2(1 - \beta) \]
Bridging the gap between the loss function and the SCA metric

\[ H(Z) \]

\[ H(Z|X) \]

\[ \text{MI } (Z; X) \geq \frac{f(\beta)}{N^*} \]

Cherisey et al. CHES 19

\[ f(\beta) = n - (1 - \beta) \log_2(2^n - 1) + \beta \log_2(\beta) + (1 - \beta) \log_2(1 - \beta) \]
Bridging the gap between the loss function and the SCA metric

\[ H(Z) \]

\[ \text{MI}(Z; X) \geq \frac{f(\beta)}{N^a} \]

Cherisey et al. CHES 19

\[ \text{PI}(Z; X; \theta) \leq \text{MI}(Z; X) \]

Bronchain et al. CRYPTO 19

\[ f(\beta) = n - (1 - \beta) \log_2(2^n - 1) + \beta \log_2(\beta) + (1 - \beta) \log_2(1 - \beta) \]
Bridging the gap between the loss function and the SCA metric

Training: minimization of the NLL a.k.a. Cross Entropy

\[ \mathcal{L}(\theta) = \frac{1}{N_p} \sum_{i=1}^{N_p} - \log_2 F(x_i, \theta)[z_i] = H(Z) - \hat{P}_{|N_p} (Z; X; \theta) \]

\[ f(\beta) = n - (1 - \beta) \log_2(2^n - 1) + \beta \log_2(\beta) + (1 - \beta) \log_2(1 - \beta) \]

This talk \( \implies \mathbb{E}_{X,Z} [\mathcal{L}(\theta)] \)

\[ MI(Z;X) \geq \frac{f(\beta)}{N_a^*} \]

Cherisey et al. CHES 19

\[ PI(Z;X;\theta) \leq MI(Z;X) \]

Bronchain et al. CRYPTO 19
Main Result

Proposition

\( \hat{\theta}_{NP} = \arg\min_{\theta} L(\theta) = \arg\max_{\theta} P(\hat{\theta})_{NP}(Z;X;\theta) \).

Then:

\[ P(\hat{\theta})_{NP}(Z;X;\theta) \xrightarrow{NP} \sup_{\theta} P(\hat{\theta})(Z;X;\theta) \leq \text{MI}(Z;X) \]

- \( L(\theta) \)
- \( H(Z) \)
- \( H(Z|X) \)
- \( \text{MI}(Z;X) \)
- \( \text{Steps} \)
Main Result

Proposition

\[
\mathcal{L}(\theta)
\]

\[
H(Z)
\]

\[
H(Z|X)
\]

\[
\text{MI}(Z; X)
\]

Steps

17/09/2020, ChES | Loïc Masure, Cécile Dumas, Emmanuel Prouff | 11/18
Main Result

Proposition

\[ \hat{\theta}_{N_p} = \arg\min_{\theta} \mathcal{L}(\theta) = \arg\max_{\theta} \hat{P}_{I}(N_p|Z;X;\hat{\theta}_{N_p}) \]

Then:

\[ \lim_{N_p \to \infty} \sup_{\theta} \hat{P}_{I}(Z;X;\theta) \leq \text{MI}(Z;X) \]

\[ \hat{P}_{I}(Z;X;\hat{\theta}_{1,000}) \]

\[ \hat{P}_{I}(Z;X;\hat{\theta}_{2,000}) \]

\[ \hat{P}_{I}(Z;X;\hat{\theta}_{5,000}) \]

\[ \hat{P}_{I}(Z;X;\hat{\theta}_{\infty}) \]
Proposition

Let $\hat{\theta}_{N_p} = \arg\min_\theta \mathcal{L}(\theta) = \arg\max_\theta \text{PI}_{N_p}(Z; X; \theta)$.
Main Result

Proposition

Let $\hat{\theta}_{N_p} = \text{argmin}_\theta \mathcal{L}(\theta) = \text{argmax}_\theta \Pi_{N_p}(Z; X; \theta)$. 

\[ \mathcal{L}(\theta) \]

\[ H(Z) \]

\[ H(Z|X) \]

\[ \Pi(Z; X; \hat{\theta}_{2,000}) \]

\[ \text{MI}(Z; X) \]

Steps
Main Result

Proposition

Let $\hat{\theta}_{N_p} = \arg\min_\theta \mathcal{L}(\theta) = \arg\max_\theta \Pi_{N_p}(Z; X; \theta)$. 

\[ \mathcal{L}(\theta) \]

\[ H(Z) \]

\[ H(Z|X) \]

\[ \Pi(Z; X; \hat{\theta}_{5,000}) \]

\[ \text{MI } (Z; X) \]

Steps

17/09/2020, Ches | Loïc Masure, Cécile Dumas, Emmanuel Prouff | 11/18
Main Result

Proposition

Let $\hat{\theta}_{N_p} = \arg\min_{\theta} \mathcal{L}(\theta) = \arg\max_{\theta} \text{PI}_{N_p}(Z; X; \theta)$.

$\mathcal{L}(\theta)$

$H(Z)$

$H(Z|X)$

$\text{PI}(Z; X; \hat{\theta}_{\infty})$

$\text{MI}(Z; X)$

Steps
Main Result

Proposition

Let $\hat{\theta}_{N_p} = \text{argmin}_\theta \mathcal{L}(\theta) = \text{argmax}_\theta \widehat{\text{PI}}_{N_p}(Z; X; \theta)$. Then:

$$\text{PI}\left(Z; X; \hat{\theta}_{N_p}\right) \xrightarrow{P \text{ as } N_p \to \infty} \sup_\theta \text{PI}(Z; X; \theta) \leq \text{MI}(Z; X)$$
Tightness of the Lower Bound

To what extent the gap $\text{PI}/\text{MI}$ is negligible?

Gap composed of three kinds of errors:

- **Approximation error**: $\sup_{\theta \in \Theta} \text{PI}(Z; X; \theta) - \text{MI}(Z; X) \leq 0$
- **Estimation error**: $N_p < \infty \Rightarrow \sup_{\theta \in \Theta} \text{PI}(Z; X; \theta) \to \hat{\text{PI}}_{N_p}(Z; X; \hat{\theta}_{N_p})$
- **Optimization error**: $\hat{\theta}_{N_p}$ unknown, $\theta_{\text{SGD}}$ instead, by Stochastic Gradient Descent (SGD)

Ideally each error must be discussed through simulations and experiments.
Tightness of the Lower Bound

To what extent the gap $\Pi / M I$ is negligible?

Gap composed of three kinds of errors:

- **Approximation error:** $\sup_{\theta \in \Theta} \Pi (Z; X; \theta) - M I (Z; X) \leq 0$

![Graph](Image)
Tightness of the Lower Bound

To what extent the gap $\text{PI}/\text{MI}$ is negligible?

Gap composed of three kinds of errors:

- **Approximation error**: $\sup_{\theta \in \Theta} \text{PI} (Z; X; \theta) - \text{MI} (Z; X) \leq 0$

- **Estimation error**: $N_p < \infty \implies \sup_{\theta \in \Theta} \text{PI} (Z; X; \theta) \rightarrow \text{PI}_{N_p} (Z; X; \hat{\theta}_{N_p})$

Steps

\[ L(\theta) \]

\[ H(Z) \]

\[ \text{PI} \left( Z; X; \hat{\theta} \right) \]

\[ \text{MI} (Z; X) \]

\[ H(Z|X) \]

Steps
Tightness of the Lower Bound

To what extent the gap $\text{PI}/\text{MI}$ is negligible?

Gap composed of three kinds of errors:

- Approximation error: $\sup_{\theta \in \Theta} \text{PI}(Z; X; \theta) - \text{MI}(Z; X) \leq 0$
- Estimation error: $N_p < \infty \implies \sup_{\theta \in \Theta} \text{PI}(Z; X; \theta) \to \text{PI}_{N_p}(Z; X; \hat{\theta}_{N_p})$
- Optimization error: $\hat{\theta}_{N_p}$ unknown, $\theta_{SGD}$ instead, by SGD

\[
\begin{align*}
\mathcal{L}(\theta) \\
\text{H}(Z) \\
\text{H}(Z|X) \\
\text{MI}(Z; X) \\
\text{PI}(Z; X; \hat{\theta})
\end{align*}
\]
Tightness of the Lower Bound

To what extent the gap $\text{PI}/\text{MI}$ is negligible?

Gap composed of three kinds of errors:

- **Approximation error:** $\sup_{\theta \in \Theta} \text{PI}(Z; X; \theta) - \text{MI}(Z; X) \leq 0$
- **Estimation error:** $N_p < \infty \implies \sup_{\theta \in \Theta} \text{PI}(Z; X; \theta) \to \widehat{\text{PI}}_{N_p}(Z; X; \hat{\theta}_{N_p})$
- **Optimization error:** $\hat{\theta}_{N_p}$ unknown, $\theta_{SGD}$ instead, by SGD

$\implies$ Ideally each error must be discussed through simulations and experiments
Outline

1. Context

2. SCA Optimization Problem versus Deep Learning Based SCA

3. NLL Minimization is PI Maximization

4. Simulation results

5. Experimental results
Settings of the simulations

**Leakage model**

- Hamming weight with additive gaussian noise ($\sigma \in [0.01; 3.2]$)
- Draw an *Exhaustive* dataset: estimation error negligible
## Settings of the simulations

<table>
<thead>
<tr>
<th>Leakage model</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Hamming weight with additive gaussian noise ($\sigma \in [0.01; 3.2]$)</td>
</tr>
<tr>
<td>▶ Draw an <em>Exhaustive</em> dataset: estimation error negligible</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PI/MI computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Computation of $\text{MI}(X; Z)$ with Monte-Carlo simulations</td>
</tr>
</tbody>
</table>
## Settings of the simulations

### Leakage model
- Hamming weight with additive gaussian noise ($\sigma \in [0.01; 3.2]$)
- Draw an *Exhaustive* dataset: estimation error negligible

### PI/MI computation
- Computation of $\text{MI}(X; Z)$ with Monte-Carlo simulations
- Training of a one layer MLP with $1,000$ neurons to maximize
  
  $\text{PI}(Z; X; \theta) = n - \mathcal{L}(\theta)$, where $n = 4$ bits
## Settings of the simulations

### Leakage model
- Hamming weight with additive gaussian noise ($\sigma \in [0.01; 3.2]$)
- Draw an *Exhaustive* dataset: estimation error negligible

### PI/MI computation
- Computation of $\text{MI}(\mathbf{X}; Z)$ with Monte-Carlo simulations
- Training of a one layer MLP with 1,000 neurons to maximize
  \[
  \text{PI}(Z; \mathbf{X}; \theta) = n - \mathcal{L}(\theta), \text{ where } n = 4 \text{ bits}
  \]

### Several case studies
- Higher-order masking: sensitive variable split into $d$ independent parts
- Shuffling: independent operations (*e.g.* 16 SBoxes in SubBytes) randomly shuffled
Simulation results

**Figure:** H-O masking, w.r.t. level of noise

**Figure:** Shuffling, w.r.t. level of noise
Simulation results

**Figure**: H-O masking, w.r.t. level of noise

**Figure**: Shuffling, w.r.t. level of noise

What to interpret

- No matter the masking order, $\text{PI}(Z; X; \theta_{\text{SGD}}) \approx \text{MI}(Z; X)$
A Comprehensive Study of Deep Learning for Side-Channel Analysis

Simulation results

**Figure:** H-O masking, w.r.t. level of noise

**Figure:** Shuffling, w.r.t. level of noise

**What to interpret**

- No matter the masking order, $\text{PI}(Z; X; \theta_{SGD}) \approx \text{MI}(Z; X)$
- For a simple MLP, the approximation error and the optimization error are negligible
Simulation results

**Figure:** H-O masking, w.r.t. level of noise

**Figure:** Shuffling, w.r.t. level of noise

**What to interpret**

- No matter the masking order, $\text{PI}(Z; X; \theta_{\text{SGD}}) \approx \text{MI}(Z; X)$
- For a simple MLP, the approximation error and the optimization error are negligible
- Any more complex model should have a negligible approximation error too
- Empirical verifications: see appendix
Outline

1. Context

2. SCA Optimization Problem versus Deep Learning Based SCA

3. NLL Minimization is PI Maximization

4. Simulation results

5. Experimental results
Application on Public Datasets

- $N_a(\theta)$: number of traces obtained with key recovery.
Application on Public Datasets

- $N_a(\theta)$: number of traces obtained with key recovery.

- So far: $N_a^* \geq \frac{f(\beta)}{\text{MI}(Z;X)}$ and $\text{PI}(Z;X;\theta_{\text{SGD}}) \approx \text{MI}(Z;X)$
Application on Public Datasets

\[ N_a(\theta) \frac{f(\beta)}{\text{PI}(Z;X;\theta)} \approx \frac{f(\beta)}{n-L(\theta)} \] : number of traces obtained with key recovery?

\[ N_a^* \geq \frac{f(\beta)}{\text{MI}(Z;X)} \text{ and } \text{PI}(Z;X;\theta_{\text{SGD}}) \approx \text{MI}(Z;X) \]
Application on Public Datasets

- $N_a(\theta) \frac{f(\beta)}{PI(Z;X;\theta)} \approx \frac{f(\beta)}{n-L(\theta)}$: number of traces obtained with key recovery?

- So far: $N^*_a \geq \frac{f(\beta)}{MI(Z;X)}$ and $PI(Z;X;\theta_{SGD}) \approx MI(Z;X)$

- Tests on public datasets, using architectures proposed in recent papers [MDP19; Kim+19]
Application on Public Datasets

- $N_a(\theta) \frac{f(\beta)}{\mathbb{P}(Z;X;\theta)} \approx \frac{f(\beta)}{n-L(\theta)}$: number of traces obtained with key recovery?

- So far: $N_a^* \geq \frac{f(\beta)}{\text{MI}(Z;X)}$ and $\mathbb{P}(Z;X;\theta_{\text{SGD}}) \approx \text{MI}(Z;X)$

- Tests on public datasets, using architectures proposed in recent papers [MDP19; Kim+19]

- Relative error $\epsilon$ computed at final epoch
Application on Public Datasets

- $N_a(\theta) \frac{f(\beta)}{\text{PI}(Z;X;\theta)} \approx \frac{f(\beta)}{n-L(\theta)}$: number of traces obtained with key recovery?
- So far: $N_a^* \geq \frac{f(\beta)}{\text{MI}(Z;X)}$ and $\text{PI}(Z;X;\theta_{\text{SGD}}) \approx \text{MI}(Z;X)$
- Tests on public datasets, using architectures proposed in recent papers [MDP19; Kim+19]
- Relative error $\epsilon$ computed at final epoch

Micro-controller protected with misalignment

Figure: AES-RD: $\epsilon = 0.16$
Application on Public Datasets

- \( N_a(\theta) \frac{f(\beta)}{\text{PI}(Z;X;\theta)} \approx \frac{f(\beta)}{n-L(\theta)} \): number of traces obtained with key recovery?

- So far: \( N_a^* \geq \frac{f(\beta)}{\text{MI}(Z;X)} \) and \( \text{PI}(Z;X;\theta_{SGD}) \approx \text{MI}(Z;X) \)

- Tests on public datasets, using architectures proposed in recent papers [MDP19; Kim+19]

- Relative error \( \epsilon \) computed at final epoch

**Figure:** AES-RD: \( \epsilon = 0.16 \)

**Figure:** ASCAD: \( \epsilon = 0.16 \)
Application on Public Datasets

- \( N_a(\theta) \frac{f(\beta)}{\text{PI}(Z;X;\theta)} \approx \frac{f(\beta)}{n-L(\theta)} \): number of traces obtained with key recovery?

- So far: \( N_a^* \geq \frac{f(\beta)}{\text{MI}(Z;X)} \) and \( \text{PI}(Z;X;\theta_{SGD}) \approx \text{MI}(Z;X) \)

- Tests on public datasets, using architectures proposed in recent papers [MDP19; Kim+19]

- Relative error \( \epsilon \) computed at final epoch

**Micro-controller protected with misalignment**

**Micro-controller protected with masking**

**Implementation on FPGA (no counter-measure)**

**Figure:** AES-RD: \( \epsilon = 0.16 \)  
**Figure:** ASCAD: \( \epsilon = 0.16 \)  
**Figure:** AES-HD: \( \epsilon = 0.18 \)
Conclusion

Takeaway messages
Conclusion

Takeaway messages

1. Minimizing the NLL loss $\equiv$ maximizing the PI $\implies$ tight lower bound of the MI $\implies$ accurate estimation of $N^*$
Conclusion

Takeaway messages

1. Minimizing the NLL loss $\equiv$ maximizing the PI $\Rightarrow$ tight lower bound of the MI $\Rightarrow$ accurate estimation of $N_a^*$
2. NLL as a loss function is sound from an evaluator point of view
Conclusion

Takeaway messages

1. Minimizing the NLL loss $\equiv$ maximizing the PI $\implies$ tight lower bound of the MI $\implies$ accurate estimation of $N_a^*$

2. NLL as a loss function is sound from an evaluator point of view

3. Enables to quantitatively measure the impact of counter-measures

Thank You! Questions?

Looking for a postdoc candidate in machine-learning-based SCA? Hire me!
A Comprehensive Study of Deep Learning for Side-Channel Analysis

References


References II


References III


A Comprehensive Study of Deep Learning for Side-Channel Analysis

Our home dataset

**Figure**: ChipWhisperer-Lite board

```
Algorithm 1 loadData
1: LD r0, X                ▶ Loads the first byte in r0
2: CLR r0                  ▶ Clears the register
3: ST X, r0                ▶ Stores 0 in the plaintext array
4: LD r0, X                ▶ Do it again to clear the bus
5: CLR r0
6: ST X, r0
7: LD r0, X
8: CLR r0
9: ST X+, r0
```

Loads sequentially an array of 16 bytes into one register and clears it \(\Longrightarrow\) no joint leakage at order \(d \geq 2\).

500,000 traces acquired.

We only work on \(n = 4\) bits,

\(|\mathcal{Z}| = 2^n = 16\).
Experiment on ChipWhisperer-Lite: masking

- Emulation of order $d$ leakages:
  \[ Z = \bigoplus_{i \in [0,d]} plain[i] \text{ for } d \in \{0, 1, 2\} \]
- Extraction of PoIs according to SNR.
- Learning curve: \( \text{PI} (Z; X; \theta_{\text{SGD}}) \)
  and \( \hat{\text{PI}}_{N_p} (Z; X; \theta_{\text{SGD}}) \) w.r.t. \( N_p \)
  \( \Rightarrow \) measures the estimation error.
Experiment on ChipWhisperer-Lite: masking

- Emulation of order $d$ leakages:
  \[ Z = \bigoplus_{i \in [0,d]} \text{plain}[i] \quad \text{for} \quad d \in \{0, 1, 2\} \]

- Extraction of PoIs according to SNR.

- Learning curve: $\Pi(Z; X; \theta_{SGD})$ and $\hat{\Pi}_{N_p}(Z; X; \theta_{SGD})$ w.r.t. $N_p$
  \[ \implies \text{measures the estimation error.} \]
Experiment on ChipWhisperer-Lite: masking

- Emulation of order $d$ leakages: 
  \[ Z = \bigoplus_{i \in [0,d]} plain[i] \] for 
  \[ d \in \{0, 1, 2\} \]
- Extraction of PoIs according to SNR.
- Learning curve: 
  \[ \hat{PI}(Z; X; \theta_{SGD}) \]
  and 
  \[ \hat{PI}_{N_p}(Z; X; \theta_{SGD}) \] w.r.t. $N_p$ 
  \( \Rightarrow \) measures the estimation error.

What to interpret

- \( \approx \) one decade lost for each new masking order \( \Rightarrow \) masking remains sound
Experiment on ChipWhisperer-Lite: masking

- Emulation of order $d$ leakages:
  $Z = \bigoplus_{i \in [0,d]} plain[i]$ for $d \in \{0, 1, 2\}$
- Extraction of PoIs according to SNR.
- Learning curve: $PI(Z; X; \theta_{SGD})$ and $\hat{PI}_{Np}(Z; X; \theta_{SGD})$ w.r.t. $N_p$ measures the estimation error.

**What to interpret**

- $\approx$ one decade lost for each new masking order $\implies$ masking remains sound
- Masking has an effect on the estimation error
Experiment on ChipWhisperer-Lite: masking

- Emulation of order $d$ leakages:
  \[ Z = \bigoplus_{i \in [0,d]} \text{plain}[i] \] for $d \in \{0, 1, 2\}$
- Extraction of PoIs according to SNR.
- Learning curve: $\text{PI}(Z; X; \theta_{SGD})$ and $\hat{\text{PI}}_{N_p}(Z; X; \theta_{SGD})$ w.r.t. $N_p$.
  \[ N_p \Rightarrow \text{measures the estimation error.} \]

What to interpret

- \[ \approx \text{one decade lost for each new masking order} \Rightarrow \text{masking remains sound} \]
- Masking has an effect on the estimation error
- For $d = 3$, $N_p < 100,000$, no information!
Experiment 5: shuffling

- Emulation of order $c$ shuffling:  
  \[ Z = \text{plain}[i] \text{ where } i \text{ is randomly drawn from a subset of } c \text{ indices} \]
- Complete trace:  \( D = 250 \)
Experiment 5: shuffling

- Emulation of order $c$ shuffling: $Z = plain[i]$ where $i$ is randomly drawn from a subset of $c$ indices
- Complete trace: $D = 250$

**Figure**: Exp. 5, shuffling
Experiment 5: shuffling

- Emulation of order $c$ shuffling: $Z = plain[i]$ where $i$ is randomly drawn from a subset of $c$ indices
- Complete trace: $D = 250$

![Figure: Exp. 5, shuffling](image)

**What to interpret**

- Linear decrease of PI, as expected [Vey+12]
Experiment 5: shuffling

- Emulation of order $c$ shuffling: $Z = plain[i]$ where $i$ is randomly drawn from a subset of $c$ indices
- Complete trace: $D = 250$

![Graph showing PI for different values of $c$ with epochs on the x-axis and PI values on the y-axis.]

**Figure:** Exp. 5, shuffling

**What to interpret**

- Linear decrease of PI, as expected [Vey+12]
- Clearly over-fitting: the estimation error non-negligible