FROM RESEARCH TO INDUSTRY

Classifying Side-Channel Desynchronized Signals with Convolutional Neural Networks

Eleonora Cagli

16/04/2019, WRAC'H 2019

LETI ITSEF - Information Technology Security Evaluation Facility - CEA Grenoble

Contents

1. Context and State of the Art

2. Deep Learning against Misalignment

2.1 Neural Network Classifiers

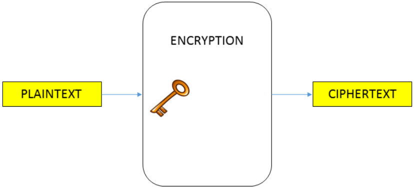
2.2 Data Augmentation

2.3 Experimental Results

3. Gradient Visualization

4. Conclusions

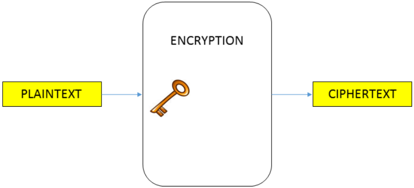
Side-Channel Vulnerability of Embedded Cryptography



Attack $\Longrightarrow$ a secret

Classical Attacks	Side-Channel Attacks
Mathematical vulnerability	
Black Box	

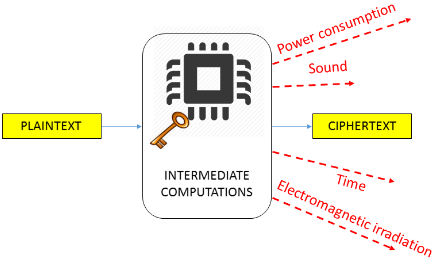
Side-Channel Vulnerability of Embedded Cryptography



Attack $\Longrightarrow$ 

	Classical Attacks	Side-Channel Attacks
	Mathematical vulnerability	
	Black Box	

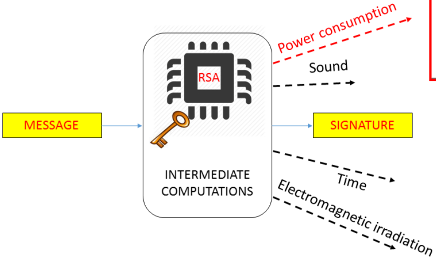
Side-Channel Vulnerability of Embedded Cryptography

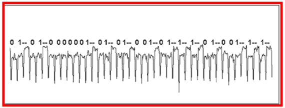


Attack $\Longrightarrow$ 

	Classical Attacks	Side-Channel Attacks
	Mathematical vulnerability	Physical vulnerability
	Black Box	Grey Box / Divide-and-conquer

Side-Channel Vulnerability of Embedded Cryptography





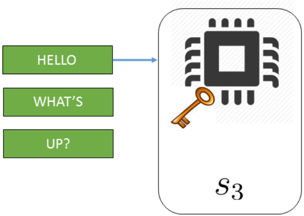
Source - Paul Kocher et al. "Introduction to differential power analysis".

Simple attack:
a single input

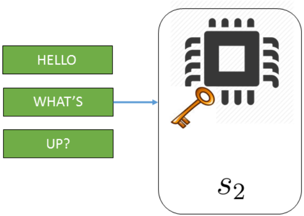
Attack $\Longrightarrow$ 

Classical Attacks	Side-Channel Attacks
Mathematical vulnerability	Physical vulnerability
Black Box	Grey Box / Divide-and-conquer

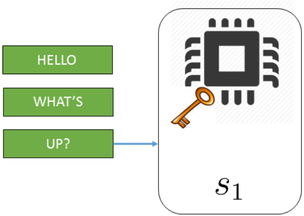
Advanced Side-Channel Attacks



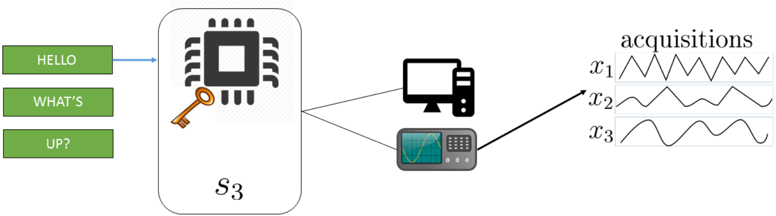
Advanced Side-Channel Attacks



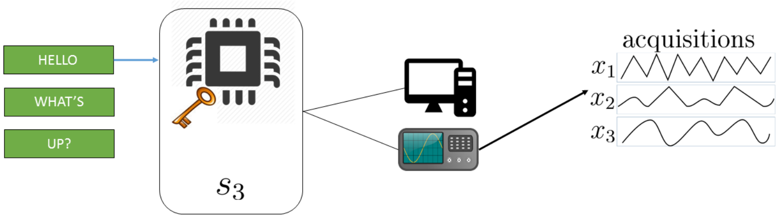
Advanced Side-Channel Attacks

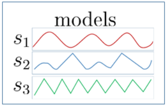


Advanced Side-Channel Attacks

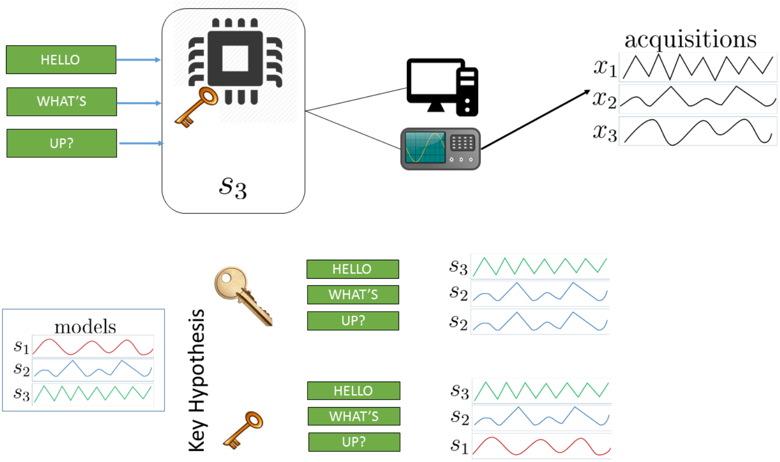


Advanced Side-Channel Attacks

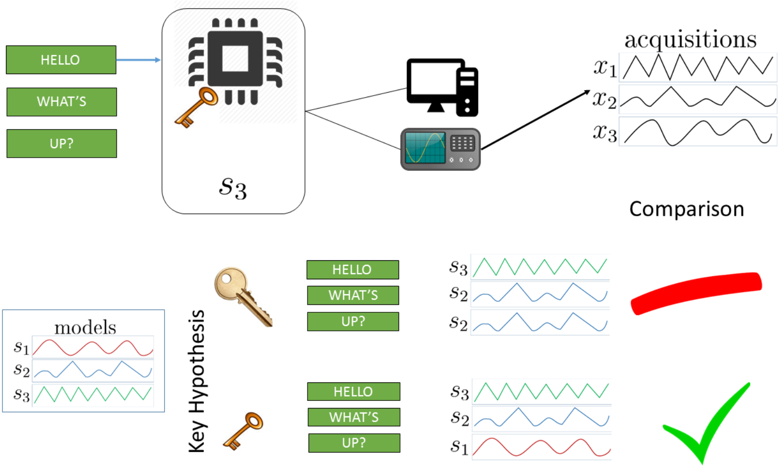




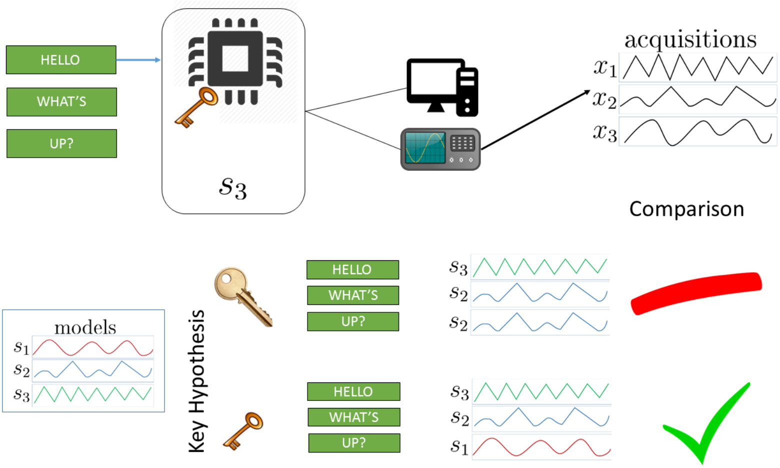
Advanced Side-Channel Attacks



Advanced Side-Channel Attacks



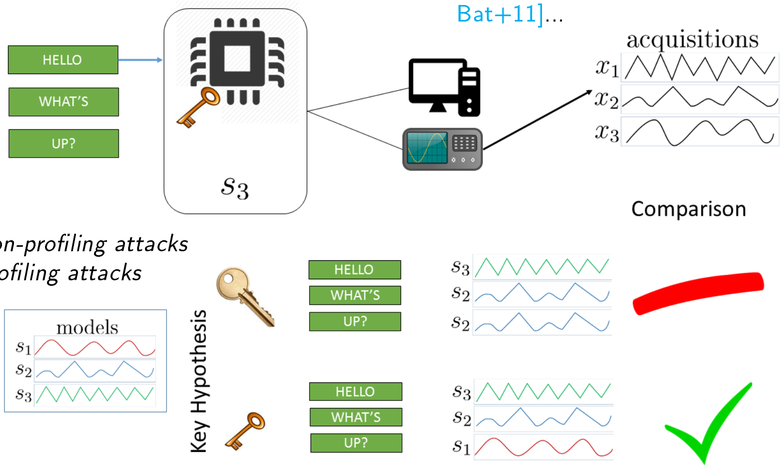
Advanced Side-Channel Attacks



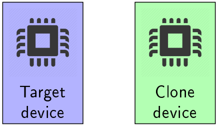
Advanced Side-Channel Attacks

Differential Power Analysis [KJJ99]
Correlation Power Analysis [BCO04]
Mutual Information Analysis [Gie+08; Bat+11]...

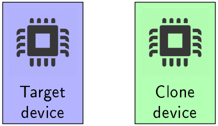
Non-profiling attacks
Profiling attacks



Profiling Attacks...Supervised Learning



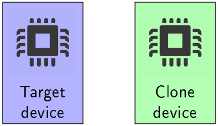
Profiling Attacks...Supervised Learning



Machine Learning

Supervised Learning

Profiling Attacks...Supervised Learning

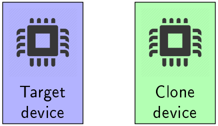


Machine Learning

Learn from data via statistic models
Task - Performance - Experience [TM97]

Supervised Learning

Profiling Attacks...Supervised Learning



Machine Learning

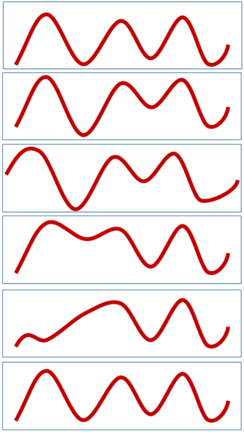
Learn from data via statistic models

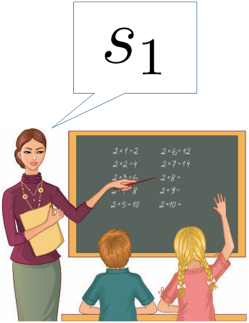
Task - Performance - Experience [TM97]

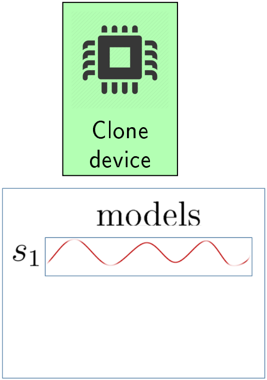
Supervised Learning

The *supervised* learning algorithms access to a dataset of examples, each associated in general to a *target* or *label*.

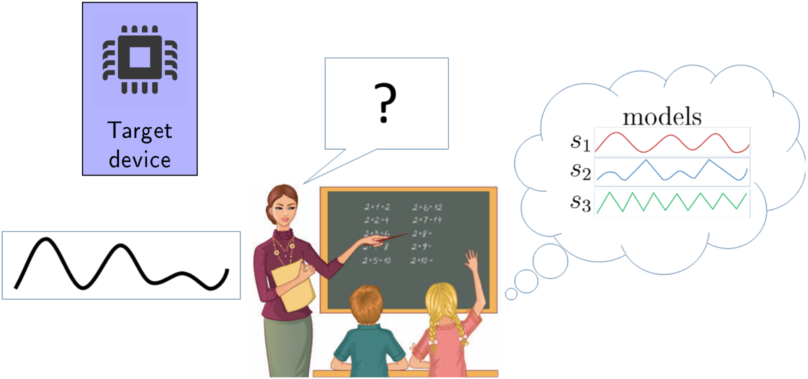
Classroom Side-Channel Attacks







Classroom Side-Channel Attacks



Classification

Classification problem

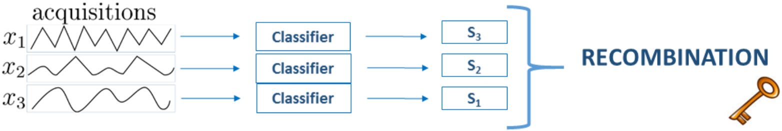
Assign to a datum $\vec{X}$ a label Z among a set of possible labels $\mathcal{Z} = \{s_1, s_2, s_3\}$, or probabilities.

Classification

Classification problem

Assign to a datum $\vec{X}$ a label Z among a set of possible labels $\mathcal{Z} = \{s_1, s_2, s_3\}$, or probabilities.

Advanced Attack as Multiple Classification Problems



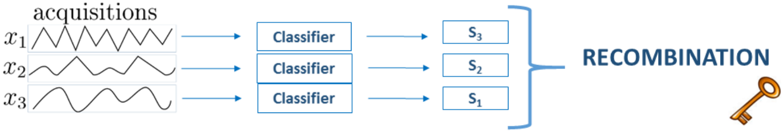
Classification

Machine Learning classifiers in Side-Channel literature: SVM ([Hos+11; HZ12]), RF ([LBM14; LBM15])

Classification problem

Assign to a datum $\vec{X}$ a label Z among a set of possible labels $\mathcal{Z} = \{s_1, s_2, s_3\}$, or probabilities.

Advanced Attack as Multiple Classification Problems



Notations

Notations and generalities

- Side-channel traces: realizations of a random vector $\vec{X} \in \mathbb{R}^D$
- D is the number of time samples (or features)
- Target: a *sensitive* variable $Z = f(\mathrm{e}, \mathrm{k})$ in $\mathcal{Z} = \{s_1, \ldots, s_{|\mathcal{Z}|}\}$

Profiling attack scenario

- labelled traces $\mathcal{D}_{\text{train}} = (\vec{x}_i, e_i, k_i)_{i=1}^{N}$, acquired under known secrets
- attack traces $\mathcal{D}_{\text{attack}} = (\vec{x}_i, e_i)_{i=1}^{N_a}$ acquired under unknown secrets

Profiling Attack

Profiling phase

- estimate
  - $p_{\vec{X} \mid Z=z}$

Attack phase

- Likelihood score for each key hypothesis k

$$d_k = p_{\vec{X} \mid Z}\left((\vec{x}_i)_{i=1,\dots,N_a}, (f(e_i, k))_{i=1,\dots,N_a}\right)$$

Profiling Attack

Profiling phase

- estimate
  - $p_{\vec{X} \mid Z=z}$ $p_{\vec{X}}$ p_Z (generative model)
  - $p_{Z \mid \vec{X}=\vec{x}}$ (discriminative model)

Attack phase

- Likelihood score for each key hypothesis k

$$d_k = p_{\vec{X} \mid Z}\left((\vec{x}_i)_{i=1,\dots,N_a}, (f(e_i,k))_{i=1,\dots,N_a}\right)$$

- A-posteriori probability score for each key hypothesis k

$$d_k = p_{Z \mid \vec{X}}\left(f\,(e_i,k)_{i=1,\dots,N_a}, (\vec{x}_i)_{i=1,\dots,N_a}\right) \ ,$$

Profiling Attack

Profiling phase

- estimate
  - $p_{\vec{X} \mid Z=z}$ $p_{\vec{X}}$ p_Z (generative model)
  - $p_{Z \mid \vec{X}=\vec{x}}$ (discriminative model)

Attack phase

- Likelihood score for each key hypothesis k

$$d_k = p_{\vec{X} \mid Z}\left((\vec{x}_i)_{i=1,\dots,N_a}, (f(e_i,k))_{i=1,\dots,N_a}\right)$$

- A-posteriori probability score for each key hypothesis k

$$d_k = p_{Z \mid \vec{X}}\left(f\left(e_i,k\right)_{i=1,\dots,N_a}, (\vec{x}_i)_{i=1,\dots,N_a}\right) \; ,$$

Profiling Attack

$\vec{X} \in \mathbb{R}^D$

Curse of dimensionality!

Profiling phase

- estimate
  - $p_{\vec{X} \mid Z=z}$ $p_{\vec{X}}$ p_Z (generative model)
  - $p_{Z \mid \vec{X}=\vec{x}}$ (discriminative model)

Attack phase

- Likelihood score for each key hypothesis k

$$d_k = p_{\vec{X} \mid Z}\left((\vec{x}_i)_{i=1,\dots,N_a}, (f(e_i,k))_{i=1,\dots,N_a}\right)$$

- A-posteriori probability score for each key hypothesis k

$$d_k = p_{Z \mid \vec{X}}\left(f\,(e_i,k)_{i=1,\dots,N_a}, (\vec{x}_i)_{i=1,\dots,N_a}\right) \ ,$$

Profiling Attack

$\vec{X} \in \mathbb{R}^D$

Curse of dimensionality!

Profiling phase

- estimate
  - $p_{\vec{X} \mid Z=z}$ $p_{\vec{X}}$ p_Z (generative model)
    - Gaussian hypothesis (**Template Attack**) [CRR03]
  - $p_{Z \mid \vec{X}=\vec{x}}$ (discriminative model)

Attack phase

- Likelihood score for each key hypothesis k

$$d_k = p_{\vec{X} \mid Z}\left((\vec{x}_i)_{i=1,\dots,N_a}, (f(e_i,k))_{i=1,\dots,N_a}\right)$$

- A-posteriori probability score for each key hypothesis k

$$d_k = p_{Z \mid \vec{X}}\left(f(e_i,k)_{i=1,\dots,N_a}, (\vec{x}_i)_{i=1,\dots,N_a}\right) ,$$

Profiling Attack

$\vec{X} \in \mathbb{R}^D$
Curse of dimensionality!

Profiling phase

- mandatory dimensionality reduction $[\mathcal{D}_{\text{train}} \longrightarrow \epsilon \colon \mathbb{R}^D \to \mathbb{R}^C]$
- estimate
  - $p_{\epsilon(\vec{X}) \mid Z=z}$ $p_{\epsilon(\vec{X})}$ p_Z (generative model)
    - Gaussian hypothesis (**Template Attack**) [CRR03]
  - $p_{Z \mid \epsilon(\vec{X})=\epsilon(\vec{x})}$ (discriminative model)

Attack phase

- Likelihood score for each key hypothesis k

$$d_k = p_{\epsilon(\vec{X}) \mid Z}\left(\left(\epsilon(\vec{x}_i)\right)_{i=1,\dots,N_a}, \left(f(e_i,k)\right)_{i=1,\dots,N_a}\right)$$

- A-posteriori probability score for each key hypothesis k

$$d_k = p_{Z \mid \epsilon(\vec{X})}\left(f\left(e_i,k\right)_{i=1,\dots,N_a}, \left(\epsilon(\vec{x}_i)\right)_{i=1,\dots,N_a}\right) \quad ,$$

Profiling Attack

$\vec{X} \in \mathbb{R}^D$

Curse of dimensionality!

Profiling phase

- manage desynchronization problem $[\mathcal{D}_{\mathbf{train}} \longrightarrow \rho \colon \mathbb{R}^D \to \mathbb{R}^D]$
- mandatory dimensionality reduction $[\mathcal{D}_{\mathbf{train}} \longrightarrow \epsilon \colon \mathbb{R}^D \to \mathbb{R}^C]$
- estimate
  - $p_{\epsilon(\rho(\vec{X}))\,|\,Z=z}\ p_{\epsilon(\rho(\vec{X}))}\ p_Z$ (generative model)
    - Gaussian hypothesis (**Template Attack**) [CRR03]
  - $p_{Z\,|\,\rho(\epsilon(\vec{X}))=\epsilon(\rho(\vec{x}))}$ (discriminative model)

Attack phase

- Likelihood score for each key hypothesis k

$$d_k = p_{\epsilon(\rho(\vec{X}))\,|\,Z}\left((\epsilon(\rho(\vec{x}_i)))_{i=1,\dots,N_a}, (f(e_i,k))_{i=1,\dots,N_a}\right)$$

- A-posteriori probability score for each key hypothesis k

$$d_k = p_{Z\,|\,\epsilon(\rho(\vec{X}))}\left(f\,(e_i,k)_{i=1,\dots,N_a}, (\epsilon(\rho(\vec{x}_i)))_{i=1,\dots,N_a}\right) ,$$

Mandatory Dimensionality Reduction

A vast domain

Features (Points of Interests - PoI) selection

- SOD [CRR03]
- SOST [BDP10]
- SNR [MOP08]/ NICV [Bha+14]
- t-test, F-test,... [GLRP06; CK14]

Feature extraction

- Principal Component Analysis (PCA) [Arc+06; BHW12]
- Linear Discriminant Analysis (LDA) [SA08; Bru+15]
- Projection Pursuits (PP) [Dur+15]

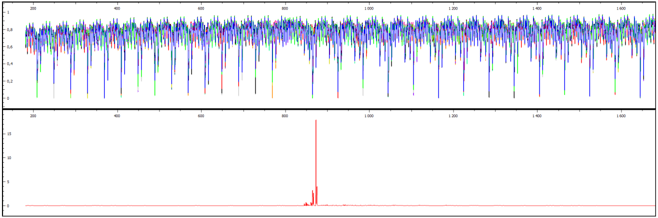


Figure: SNR computed on synchronized traces.

Manage desynchronization problem

Misaligning Countermeasures

- Random Delays, Clock Jittering, ...
- In theory: insufficient to provide security, since information still leak (somewhere)
- In practice: one of the main issues for evaluators

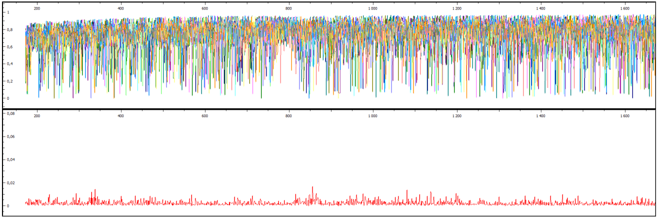


Figure: SNR computed on desynchronized traces.

Manage desynchronization problem

Misaligning Countermeasures

- Random Delays, Clock Jittering, ...
- In theory: insufficient to provide security, since information still leak (somewhere)
- In practice: one of the main issues for evaluators

Realignment

Mandatory realignment preprocessing

- not a wide literature
- in practice: evaluation labs home-made realignment techniques
- signal deformations or pattern extraction based on prior unverified assumptions
- Risks:
  - deformations $\rightarrow$ information degradation
  - pattern extraction $\rightarrow$ information loss

This talk

Profiling phase

- manage de-synchronization problem $[\mathcal{D}_{\text{train}} \longrightarrow \rho\colon \mathbb{R}^D \to \mathbb{R}^D]$
- mandatory dimensionality reduction $[\mathcal{D}_{\text{train}} \longrightarrow \epsilon\colon \mathbb{R}^D \to \mathbb{R}^C]$
- estimate
  - $p_{\epsilon(\rho(\vec{X}))\mid Z=z}$, $p_{\epsilon(\rho(\vec{X}))}$, p_Z (generative model)
    - Gaussian hypothesis (**Template Attack**)[CRR03]
- $p_{Z\mid\epsilon(\rho(\vec{x}))}$ (discriminative model)

This talk

Convolutional Neural Network: integrated approach (deal desynchronization + extraction feature + approximate a discriminative model)

This talk

Profiling phase

- manage de-synchronization problem $[\mathcal{D}_{\text{train}} \longrightarrow \rho\colon \mathbb{R}^D \to \mathbb{R}^D]$ DEEP LEARNING
- mandatory dimensionality reduction $[\mathcal{D}_{\text{train}} \longrightarrow \epsilon\colon \mathbb{R}^D \to \mathbb{R}^C]$
- estimate
  - $p_{\epsilon(\rho(\vec{X}))\mid Z=z}$, $p_{\epsilon(\rho(\vec{X}))}$, p_Z (generative model)
    - Gaussian hypothesis (**Template Attack**)[CRR03]
- $p_{Z\mid\vec{X}}$ (discriminative model)
  by means of a neural network $\hat{p}(\vec{x}, W) \approx p_{Z\mid\vec{X}=\vec{x}}$

This talk

Convolutional Neural Network: integrated approach (deal desynchronization + extraction feature + approximate a discriminative model)

Contents

1. Context and State of the Art

2. Deep Learning against Misalignment

2.1 Neural Network Classifiers

2.2 Data Augmentation

2.3 Experimental Results

3. Gradient Visualization

4. Conclusions

Multi-Layer Perceptron

In SCA litterature [MHM13; MZ13; MMT15; MDM16]

Multi-Layer Perceptron (MLP)

$$\hat{p}(\vec{x}, W) = s \circ \lambda_n \circ \sigma_{n-1} \circ \lambda_{n-1} \circ \cdots \circ \lambda_1(\vec{x}) = \vec{y} \approx p_{Z \mid \vec{X}=\vec{x}}$$

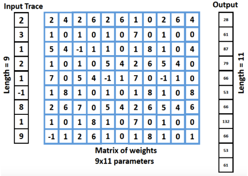
Multi-Layer Perceptron

In SCA litterature [MHM13; MZ13; MMT15; MDM16]

Multi-Layer Perceptron (MLP)

$$\hat{p}(\vec{x}, W) = s \circ \lambda_n \circ \sigma_{n-1} \circ \lambda_{n-1} \circ \cdots \circ \lambda_1(\vec{x}) = \vec{y} \approx p_{Z \mid \vec{X}=\vec{x}}$$

λ_i linear functions (linear combinations of time samples) depending on some **trainable weights** W



Multi-Layer Perceptron

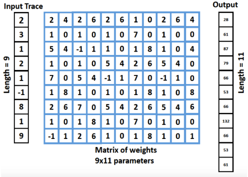
In SCA litterature [MHM13; MZ13; MMT15; MDM16]

Multi-Layer Perceptron (MLP)

$$\hat{p}(\vec{x}, W) = s \circ \lambda_n \circ \sigma_{n-1} \circ \lambda_{n-1} \circ \cdots \circ \lambda_1(\vec{x}) = \vec{y} \approx p_{Z \mid \vec{X}=\vec{x}}$$

λ_i linear functions (linear combinations of time samples) depending on some **trainable weights** W

σ_i non-linear *activation* functions



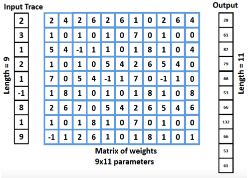
Multi-Layer Perceptron

In SCA litterature [MHM13; MZ13; MMT15; MDM16]

Multi-Layer Perceptron (MLP)

$$\hat{p}(\vec{x}, W) = s \circ \lambda_n \circ \sigma_{n-1} \circ \lambda_{n-1} \circ \cdots \circ \lambda_1(\vec{x}) = \vec{y} \approx p_{Z \mid \vec{X}=\vec{x}}$$

- λ_i linear functions (linear combinations of time samples) depending on some **trainable weights** W
- σ_i non-linear *activation* functions
- s normalizing *softmax* function



Multi-Layer Perceptron

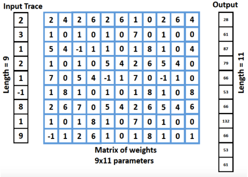
In SCA litterature [MHM13; MZ13; MMT15; MDM16]

Multi-Layer Perceptron (MLP)

$$\hat{p}(\vec{x}, W) = s \circ \lambda_n \circ \sigma_{n-1} \circ \lambda_{n-1} \circ \cdots \circ \lambda_1(\vec{x}) = \vec{y} \approx p_{Z \mid \vec{X}=\vec{x}}$$

- λ_i linear functions (linear combinations of time samples) depending on some **trainable weights** W
- σ_i non-linear *activation* functions
- s normalizing *softmax* function

Architecture hyper-parameters



Multi-Layer Perceptron

In SCA litterature [MHM13; MZ13; MMT15; MDM16]

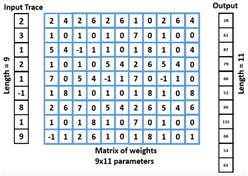
Multi-Layer Perceptron (MLP)

$$\hat{p}(\vec{x}, W) = s \circ \lambda_n \circ \sigma_{n-1} \circ \lambda_{n-1} \circ \cdots \circ \lambda_1(\vec{x}) = \vec{y} \approx p_{Z \mid \vec{X}=\vec{x}}$$

- λ_i linear functions (linear combinations of time samples) depending on some **trainable weights** W
- σ_i non-linear *activation* functions
- s normalizing *softmax* function

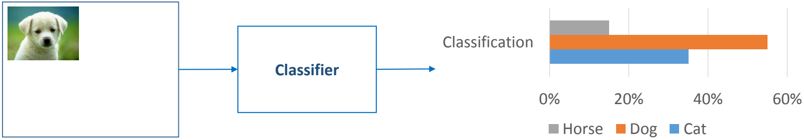
Architecture hyper-parameters

Universal approximation theorem



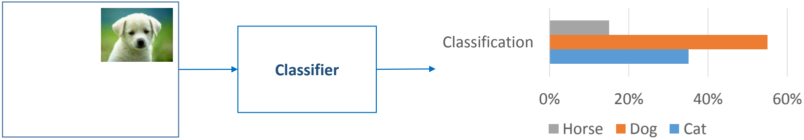
Convolutional Neural Networks

Translation-Invariance



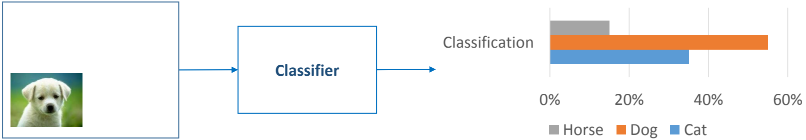
Convolutional Neural Networks

Translation-Invariance



Convolutional Neural Networks

Translation-Invariance



Convolutional Neural Networks

Translation-Invariance

Convolutional Neural Networks

Translation-Invariance

Convolutional Neural Networks

Translation-Invariance

Convolutional Layers

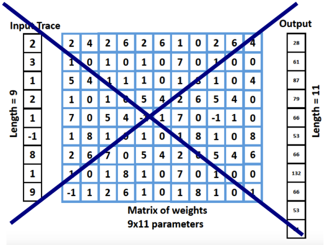


Figure: Linear layer in an MLP.

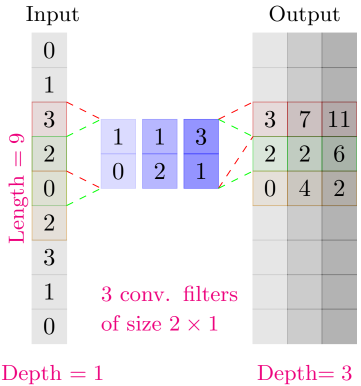


Figure: Convolutional layer in a CNN.

Pooling Layers

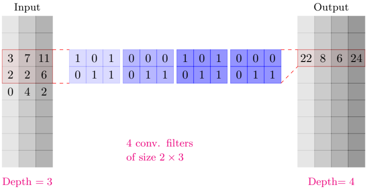


Figure: Convolutional layer in a CNN.

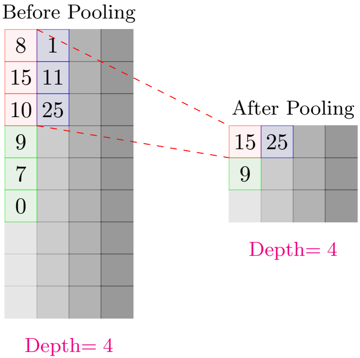
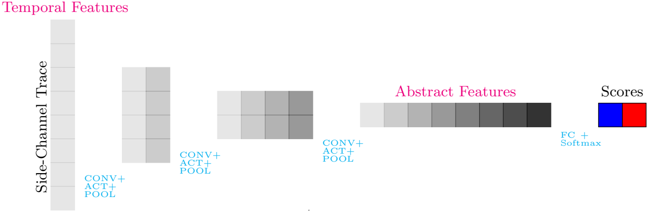


Figure: Pooling layer in a CNN.

A kind of CNN architecture



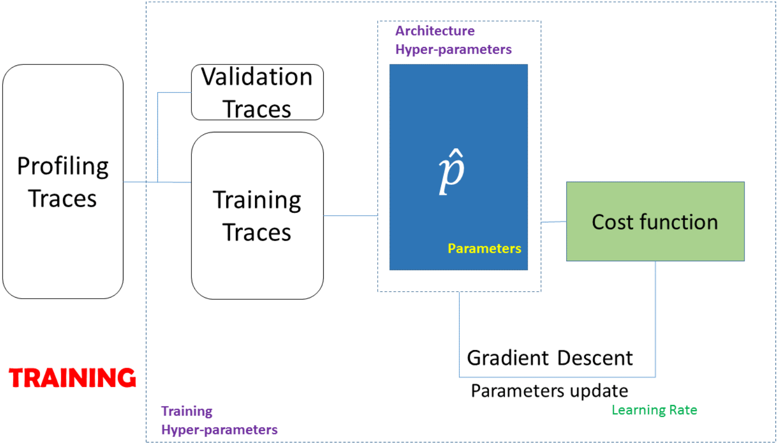
Architecture inspired by AlexNet [KSH12], VGG [SZ14], ResNet [He+16] design rules:

- Reduce temporal features to only one
- Maintain time complexity of each layer (one-half pooling when number of feature maps is doubled)

CHES 2017 - *Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures - Profiling Attacks Without Pre-processing*. E. Cagli - C. Dumas - E. Prouff

- 4 Conv + Pool layers
- tanh activations
- batch normalisation [IS15]
- 1 *fully connected layer* + softmax

Training and Validation (1)



Cost function - Cross-entropy

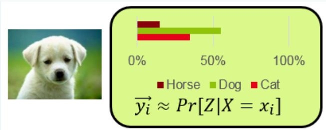
- batch of training data $(\vec{x}_i, z_i)_{i\in I}$, outputs of the current model $(\vec{y}_i)_{i\in I}$
- labels $z_i = s_j$ are *one-hot encoded*: $\vec{z}_i = \vec{s}_j = (0, \ldots, 0, \underbrace{1}_{j}, 0, \ldots, 0)$

Loss function

$$\mathcal{L} = -\frac{1}{|I|}\sum_{i\in I}\sum_{t=1}^{|\mathcal{Z}|}\vec{z}_i[t]\log\vec{y}_i[t] \qquad (1)$$

Maximum-*a-posteriori* or Cross-entropy

- $\vec{y}_i \approx \Pr[Z \mid \vec{X} = \vec{x}_i]$



Cost function - Cross-entropy

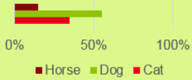
- batch of training data $(\vec{x}_i, z_i)_{i \in I}$, outputs of the current model $(\vec{y}_i)_{i \in I}$
- labels $z_i = s_j$ are *one-hot encoded*: $\vec{z}_i = \vec{s}_j = (0, \dots, 0, \underbrace{1}_{j}, 0, \dots, 0)$

Loss function

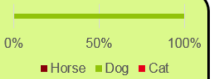
$$\mathcal{L} = -\frac{1}{|I|} \sum_{i \in I} \sum_{t=1}^{|\mathcal{Z}|} \vec{z}_i[t] \log \vec{y}_i[t] \qquad (1)$$

Maximum-*a-posteriori* or Cross-entropy

- $\vec{y}_i \approx \Pr[Z \mid \vec{X} = \vec{x}_i]$
- $\vec{z}_i \approx \Pr[Z \mid Z = \vec{s}_j]$
- $\mathbb{H}(\vec{z}_i, \vec{y}_i) = \mathbb{H}(\vec{z}_i) + D_{KL}(\vec{z}_i || \vec{y}_i) = \mathbb{E}_{\vec{z}_i}[-\log \vec{y}_i] = -\sum_{t=1}^{|\mathcal{Z}|} \vec{z}_i[t] \log \vec{y}_i[t]$

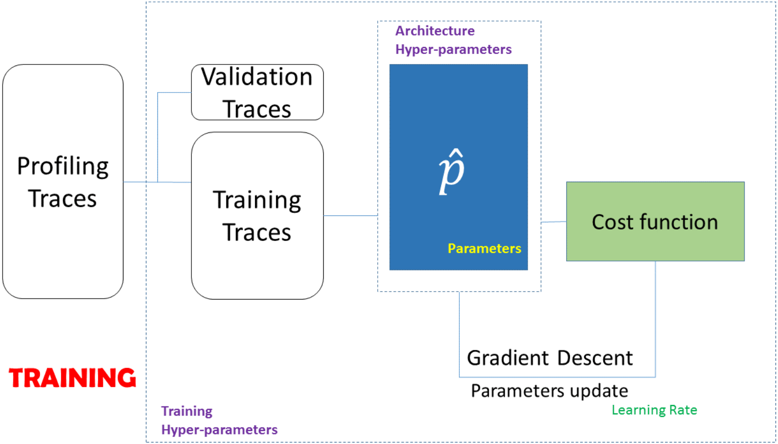


$\vec{y}_i \approx Pr[Z|X = x_i]$

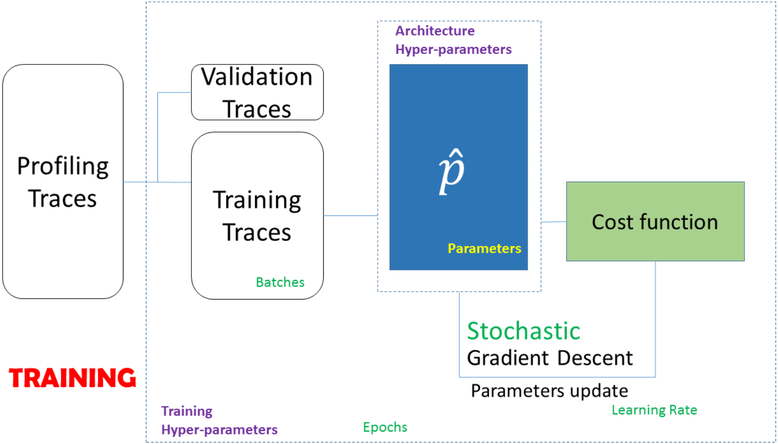


$\vec{z}_i = Pr[Z|Z = s_j]$

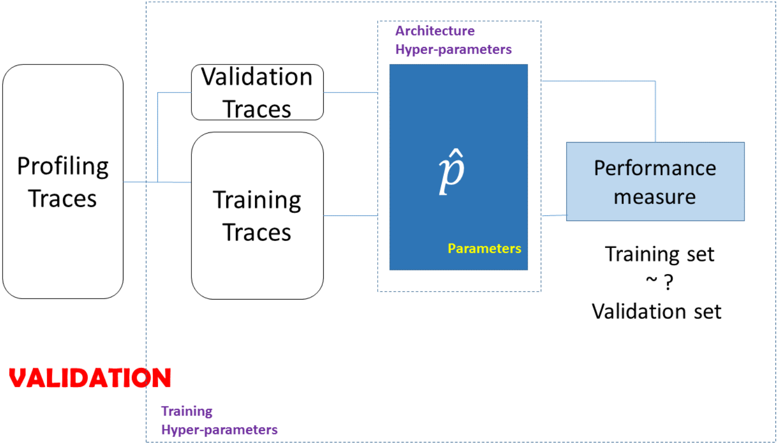
Training and Validation (2)



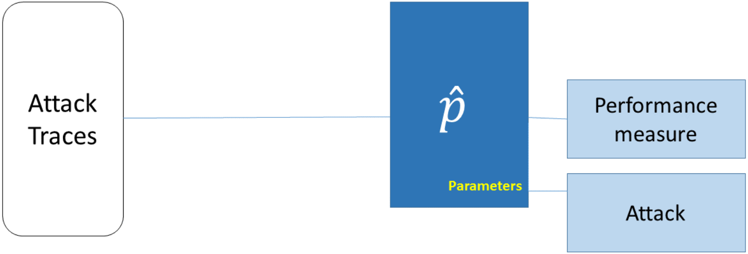
Training and Validation (2)



Training and Validation (2)



Training and Validation (2)



TEST

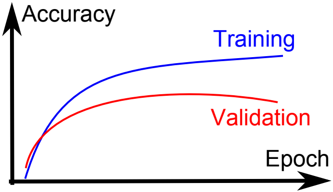
Overfitting

Accuracy

$$\frac{\text{Correct predictions}}{\text{Total predictions}}$$

Evaluate and compare training and validation accuracy

Learn by heart (**OVERFITTING**)



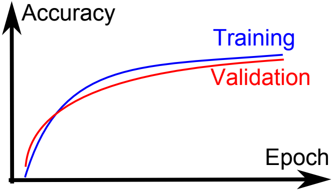
Overfitting

Accuracy

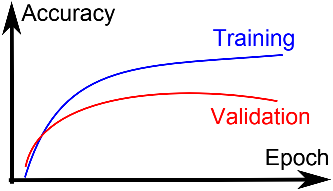
$$\frac{\text{Correct predictions}}{\text{Total predictions}}$$

Evaluate and compare training and validation accuracy

Understand significant features



Learn by heart (**OVERFITTING**)



Overfitting

Accuracy

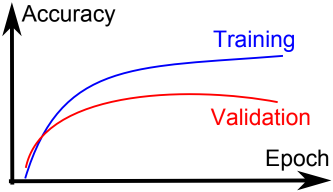
$$\frac{\text{Correct predictions}}{\text{Total predictions}}$$

Evaluate and compare training and validation accuracy

Why?
- Too complex model
- Not enough training data

Solution?
- Reduce model capacity
- Regularization
- Dropout
- Early-Stopping
- Data augmentation

Learn by heart (**OVERFITTING**)



Overfitting

Accuracy

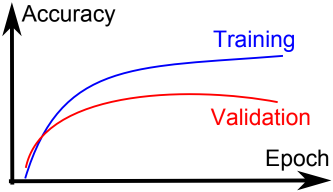
$$\frac{\text{Correct predictions}}{\text{Total predictions}}$$

Evaluate and compare training and validation accuracy

Why?
- Too complex model
- Not enough training data

Solution?
- Reduce model capacity
- Regularization
- Dropout
- Early-Stopping
- Data augmentation

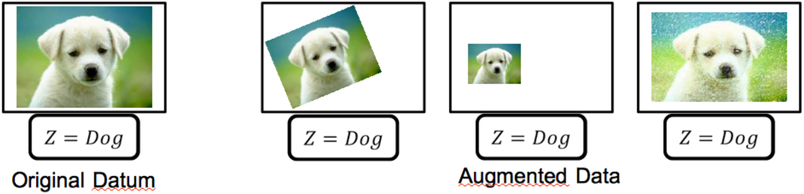
Learn by heart (**OVERFITTING**)



Data Augmentation

Data Augmentation

Artificially generate new training data by deforming those previously acquired,
Applying transformations that preserve the label Z



Side-Channel Data Augmentation

Countermeasure Emulation Idea

Emulate the effects of misaligning countermeasures to generate new traces

SHIFTING

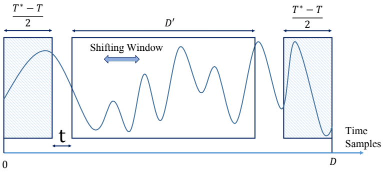


Figure: SH_T

ADD-REMOVE

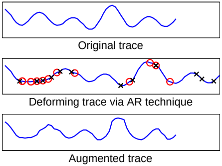


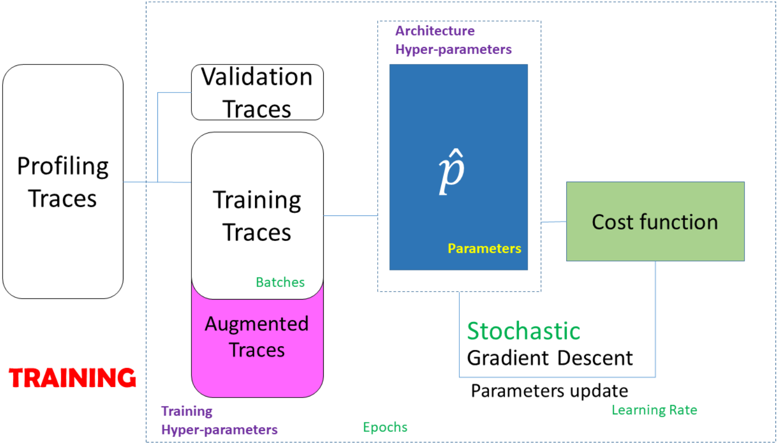
Figure: AR_R

Parameter T: $\sharp$ of possible positions

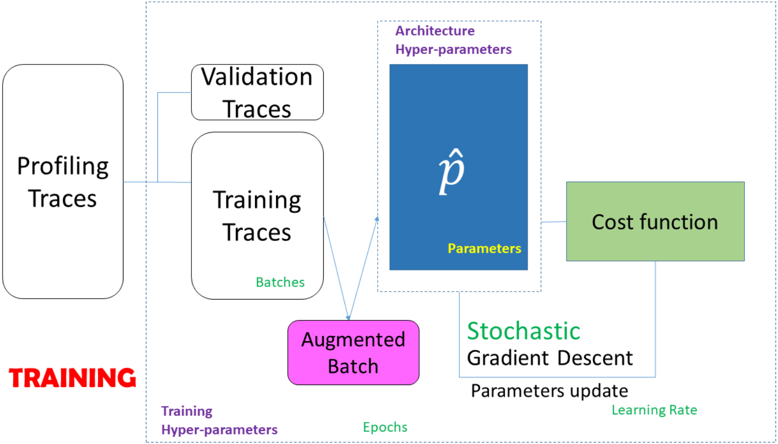
Parameter R: $\sharp$ of added and removed points

Data Augmentation techniques are applied online during training phase.

Training with Data Augmentation



Training with Data Augmentation



Experimental Results

- Random delays (software countermeasure)
- Artificial Jitter (simulated hardware countermeasure)
- Real Jitter (hardware countermeasure)

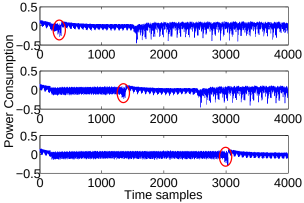
Keras 1.2.1 library with Tensorflow backend [Cho+15] (open source, today 2.2.4)

Experimental Results

- Random delays (software countermeasure)
- Artificial Jitter (simulated hardware countermeasure)
- Real Jitter (hardware countermeasure)

Keras 1.2.1 library with Tensorflow backend [Cho+15] (open source, today 2.2.4)

Random delays



(a) One leaking operation

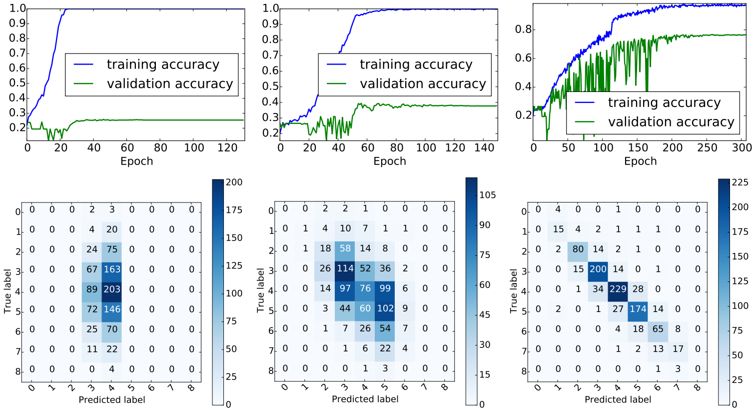
Setup

- Target Chip: Atmega328P
- Target Variable: $Z = \mathrm{HW}(\mathrm{Sbox}(P \oplus K))$
- Acquisition: through ChipWhisperer[OC14] platform, $\approx 4,000$ time samples
- Countermeasure: Random Delays - insertion of r *nop* operations, $r \in [0,127]$ uniform random
- $1,000$ training traces

Random delays

Data augmentation vs overfitting

Training

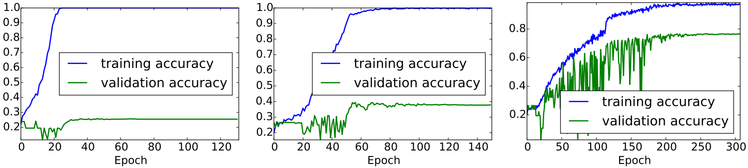


SH_0 SH_{100} SH_{500}

Random delays

Data augmentation vs overfitting

Training



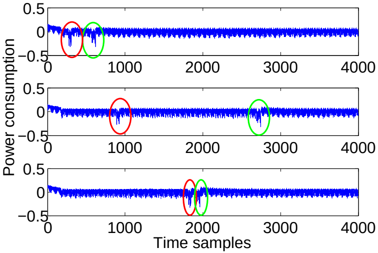
SH_0 SH_{100} SH_{500}

Attack

		SH_0		SH_{100}		SH_{500}	
Accuracy	$N^\star$	27.0%	$> 1,000$	31.8%	101	**78%**	**7**

Table: $N^\star$ = number of attack traces to have GE = 1.

Random Delays - Two Leaking Operations



Two leaking operations

First operation - Test acc: 76.8%, $N^{\star} = 7$
Second operation - Test acc: 82.5%, $N^{\star} = 6$

Conclusions about CNN

- CNNs provide an integrated approach to construct a discriminative model from misaligned data

Conclusions about CNN

- CNNs provide an integrated approach to construct a discriminative model from misaligned data
- CNN models may have high capacity and require plenty of data to be trained

Conclusions about CNN

- CNNs provide an integrated approach to construct a discriminative model from misaligned data
- CNN models may have high capacity and require plenty of data to be trained
- Side-Channel-adapted Data Augmentation techniques

Conclusions about CNN

- CNNs provide an integrated approach to construct a discriminative model from misaligned data
- CNN models may have high capacity and require plenty of data to be trained
- Side-Channel-adapted Data Augmentation techniques
- Effectiveness/efficiency of the CNN+Data Augmentation approach experimentally verified

Conclusions about CNN

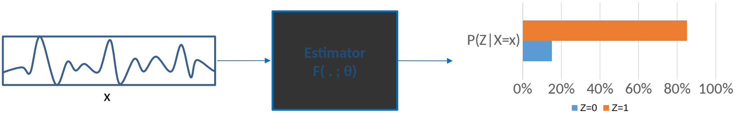
- CNNs provide an integrated approach to construct a discriminative model from misaligned data
- CNN models may have high capacity and require plenty of data to be trained
- Side-Channel-adapted Data Augmentation techniques
- Effectiveness/efficiency of the CNN+Data Augmentation approach experimentally verified
- Today Deep Learning attacks systematically performed in Side-Channel tests for embedded cryptography evaluation

Conclusions about CNN

- CNNs provide an integrated approach to construct a discriminative model from misaligned data
- CNN models may have high capacity and require plenty of data to be trained
- Side-Channel-adapted Data Augmentation techniques
- Effectiveness/efficiency of the CNN+Data Augmentation approach experimentally verified
- Today Deep Learning attacks systematically performed in Side-Channel tests for embedded cryptography evaluation

Among new problematics...

Deep Learning provides black-box models:

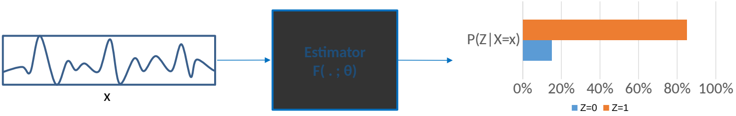


Conclusions about CNN

- CNNs provide an integrated approach to construct a discriminative model from misaligned data
- CNN models may have high capacity and require plenty of data to be trained
- Side-Channel-adapted Data Augmentation techniques
- Effectiveness/efficiency of the CNN+Data Augmentation approach experimentally verified
- Today Deep Learning attacks systematically performed in Side-Channel tests for embedded cryptography evaluation

Among new problematics...

Deep Learning provides black-box models:



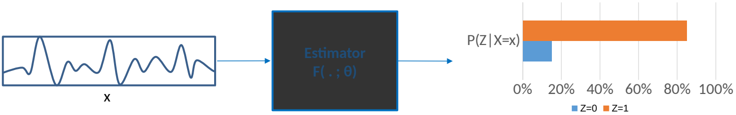
Lack of posterior knowledge: how did the model learn?

Conclusions about CNN

- CNNs provide an integrated approach to construct a discriminative model from misaligned data
- CNN models may have high capacity and require plenty of data to be trained
- Side-Channel-adapted Data Augmentation techniques
- Effectiveness/efficiency of the CNN+Data Augmentation approach experimentally verified
- Today Deep Learning attacks systematically performed in Side-Channel tests for embedded cryptography evaluation

Among new problematics...

Deep Learning provides black-box models:



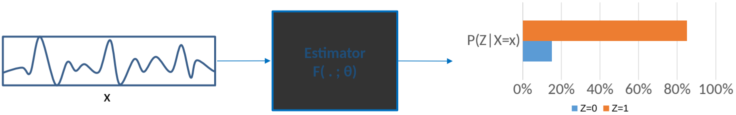
Lack of posterior knowledge: how did the model learn?
Lack of trust: where did the model get the information?

Conclusions about CNN

- CNNs provide an integrated approach to construct a discriminative model from misaligned data
- CNN models may have high capacity and require plenty of data to be trained
- Side-Channel-adapted Data Augmentation techniques
- Effectiveness/efficiency of the CNN+Data Augmentation approach experimentally verified
- Today Deep Learning attacks systematically performed in Side-Channel tests for embedded cryptography evaluation

Among new problematics...

Deep Learning provides black-box models:



Lack of posterior knowledge: how did the model learn?
Lack of trust: where did the model get the information?
No hints to correct vulnerability

Contents

1. Context and State of the Art

2. Deep Learning against Misalignment

2.1 Neural Network Classifiers

2.2 Data Augmentation

2.3 Experimental Results

3. Gradient Visualization

4. Conclusions

Gradient Visualization

L.Masure *et al.*, *Gradient Visualization for General Characterization in Profiling Attacks*, COSADE 2019 (Darmstadt, 5th April 2019)

- proposes a characterization technique based on a trained CNN

Gradient Visualization

L.Masure *et al.*, *Gradient Visualization for General Characterization in Profiling Attacks*, COSADE 2019 (Darmstadt, 5th April 2019)

- proposes a characterization technique based on a trained CNN
- able to detect Points of Interest (PoIs) as long as the model has learned something

Gradient Visualization

L.Masure *et al.*, *Gradient Visualization for General Characterization in Profiling Attacks*, COSADE 2019 (Darmstadt, 5th April 2019)

- proposes a characterization technique based on a trained CNN
- able to detect Points of Interest (PoIs) as long as the model has learned something
- already proposed in Image Recognition [SVZ13; Spr+14]

Gradient Visualization

L.Masure *et al.*, *Gradient Visualization for General Characterization in Profiling Attacks*, COSADE 2019 (Darmstadt, 5th April 2019)

- proposes a characterization technique based on a trained CNN
- able to detect Points of Interest (PoIs) as long as the model has learned something
- already proposed in Image Recognition [SVZ13; Spr+14]
- starts to be used in SCA [Tim19; HGG19]

An ideal case

Ideal case: we know $F^* = \Pr[Z|\mathbf{X}]$ (*i.e.* $F^* : \mathbb{R}^D \rightarrow \mathcal{P}(\mathcal{Z}) \subset [0,1]^{|\mathcal{Z}|}$)

An example

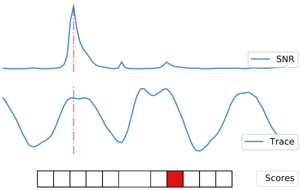
An explanation

- Assume the informative leakage is very localized (few PoIs)

An ideal case

Ideal case: we know $F^* = \Pr[Z|\mathbf{X}]$ (*i.e.* $F^* : \mathbb{R}^D \to \mathcal{P}(\mathcal{Z}) \subset [0,1]^{|\mathcal{Z}|}$)

An example



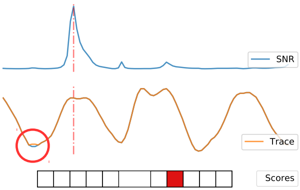
An explanation

- Assume the informative leakage is very localized (few PoIs)
- Consider a new trace and its label $\vec{x}, z$

An ideal case

Ideal case: we know $F^* = \Pr[Z|\mathbf{X}]$ (*i.e.* $F^* : \mathbb{R}^D \to \mathcal{P}(\mathcal{Z}) \subset [0,1]^{|\mathcal{Z}|}$)

An example



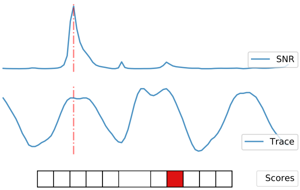
An explanation

- Assume the informative leakage is very localized (few PoIs)
- t_0 non informative: $\vec{x}[t_0] \mapsto \vec{x}[t_0] + \epsilon$ not sensitive
- In other words, t_0 non informative $\to \frac{\partial}{\partial \vec{x}[t_0]} F^*(\vec{x})[z] \approx 0$

An ideal case

Ideal case: we know $F^* = \Pr[Z|\mathbf{X}]$ ($i.e.$ $F^* : \mathbb{R}^D \to \mathcal{P}(\mathcal{Z}) \subset [0,1]^{|\mathcal{Z}|}$)

An example



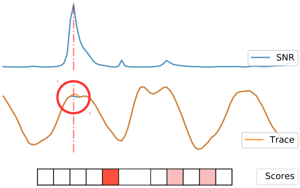
An explanation

- Assume the informative leakage is very localized (few PoIs)
- t_0 non informative: $\vec{x}[t_0] \mapsto \vec{x}[t_0] + \epsilon$ not sensitive
- In other words, t_0 non informative $\rightarrow \frac{\partial}{\partial \vec{x}[t_0]} F^*(\vec{x})[z] \approx 0$

An ideal case

Ideal case: we know $F^* = \Pr[Z|\mathbf{X}]$ (*i.e.* $F^* : \mathbb{R}^D \rightarrow \mathcal{P}(\mathcal{Z}) \subset [0,1]^{|\mathcal{Z}|}$)

An example



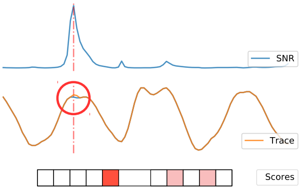
An explanation

- Assume the informative leakage is very localized (few PoIs)
- t_1 informative: $\vec{x}[t_1] \mapsto \vec{x}[t_1] + \epsilon$ is likely to affect the optimal model's decision
- t_1 informative $\rightarrow \left| \frac{\partial}{\partial \vec{x}[t_1]} F^*(\vec{x})[z] \right| > 0$

An ideal case

Ideal case: we know $F^* = \Pr[Z|\mathbf{X}]$ (*i.e.* $F^* : \mathbb{R}^D \to \mathcal{P}(\mathcal{Z}) \subset [0,1]^{|\mathcal{Z}|}$)

An example



An explanation

- Assume the informative leakage is very localized (few PoIs)
- t_1 informative: $\vec{x}[t_1] \mapsto \vec{x}[t_1] + \epsilon$ is likely to affect the optimal model's decision
- t_1 informative $\to \left| \frac{\partial}{\partial \vec{x}[t_1]} F^*(\vec{x})[z] \right| > 0$

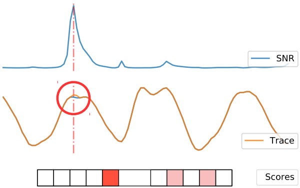
Consequences

If t is a PoI, then it should be seen in the gradients $\nabla_{\vec{x}} F^*(\vec{x})[z]$

An ideal case

Ideal case: we know $F^* = \Pr[Z|\mathbf{X}]$ (*i.e.* $F^* : \mathbb{R}^D \rightarrow \mathcal{P}(\mathcal{Z}) \subset [0,1]^{|\mathcal{Z}|}$)

An example



An explanation

- Assume the informative leakage is very localized (few PoIs)
- t_1 informative: $\vec{x}[t_1] \mapsto \vec{x}[t_1] + \epsilon$ is likely to affect the optimal model's decision
- t_1 informative $\rightarrow \left| \frac{\partial}{\partial \vec{x}[t_1]} F^*(\vec{x})[z] \right| > 0$

Consequences

If t is a PoI, then it should be seen in the gradients $\nabla_{\vec{x}} F^*(\vec{x})[z]$

Application on experimental data

Description

ASCAD dataset [Pro+18]: https://github.com/ANSSI-FR/ASCAD
50,000 traces, each of 700 points
Source codes of secure implementations of AES128 for public 8-bit architectures (https://github.com/ANSSI-FR/secAES-ATmega8515)
Corresponds to the first AES round
Three cases studied:

1. **No countermeasure**: synchronized traces, no masking
2. **Artificial random shift**
3. **Synchronized traces, boolean masking (unknown masks)**

Trained model

CNN with a VGG-like architecture
Grid search of hyperparameters
Best model: minimal trace number when the guessing entropy reaches 2

First experiment: no countermeasure

Average number of traces to recover the secret key: 3

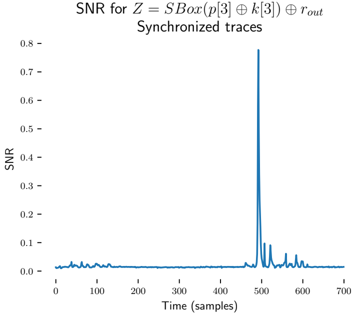


Figure: SNR

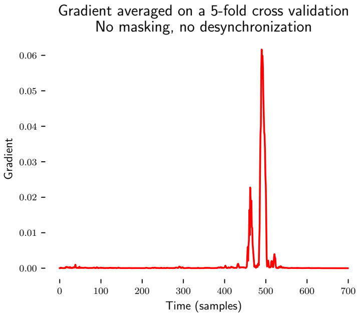


Figure: Gradient Visualization

Second experiment: with desynchronization

Average number of traces to recover the secret key: 3.6

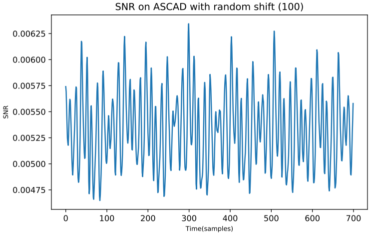


Figure: No PoI emphasized ☹

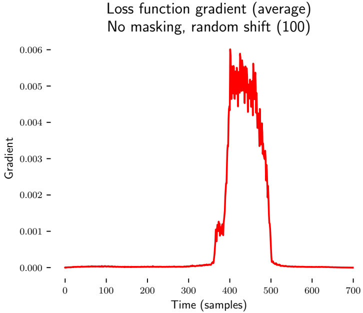


Figure: Band of peaks ☺

Second experiment: with desynchronization

Average number of traces to recover the secret key: 3.6

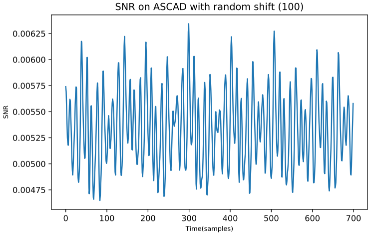


Figure: No PoI emphasized ☹

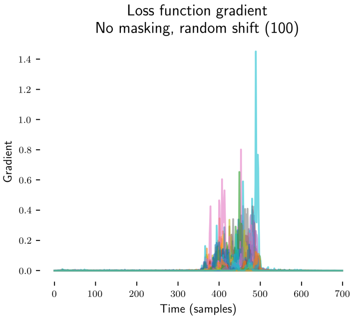


Figure: Characterization for each trace 🙂

Third experiment: with masking

Average number of traces to recover the secret key: ≈ 100

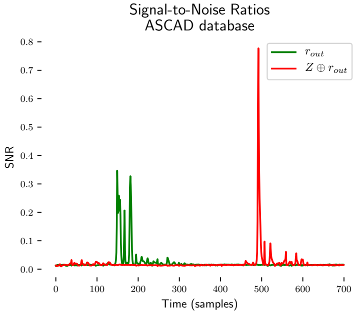


Figure: Requires knowledge of the masks ☹

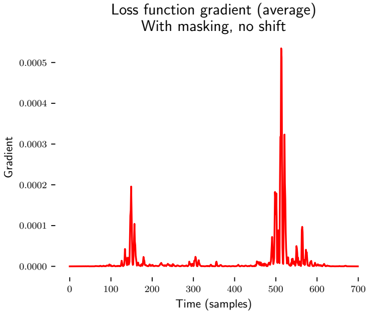


Figure: No knowledge required ☺

Be careful not to overfit !

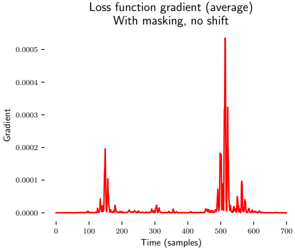


Figure: GV without overfitting 🙂

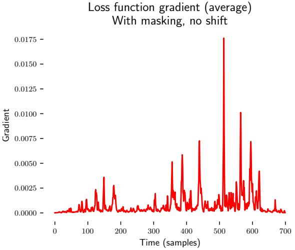


Figure: GV with overfitting ☹

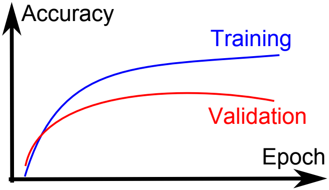


Figure: Solution: early-stopping

Conclusions on Gradient Visualization

- Reinforces trust into Deep Learning tools: in absence of overfitting information comes from well-identifiable regions of interest
- May be used to guide early-stopping and prevent overfitting
- Provides characterization of leakages, allows developpers to correct the vulnerability

Contents

1. Context and State of the Art

2. Deep Learning against Misalignment

2.1 Neural Network Classifiers

2.2 Data Augmentation

2.3 Experimental Results

3. Gradient Visualization

4. Conclusions

Conclusions

- Curse of dimensionality affects the potential optimality of profiling attacks
- Machine Learning : profiling attacks $\approx$ classification task
- Generative model approach: Template Attacks
- Discriminative model approach:
  - Neural Networks, big data scalability
  - CNN to integrate resynchronization in a unique model construction process

Conclusions

- Curse of dimensionality affects the potential optimality of profiling attacks
- Machine Learning : profiling attacks $\approx$ classification task
- Generative model approach: Template Attacks
- Discriminative model approach:
  - Neural Networks, big data scalability
  - CNN to integrate resynchronization in a unique model construction process

Today and in the future

- Going towards a community "ML for Embedded Security Analysis"

Conclusions

- Curse of dimensionality affects the potential optimality of profiling attacks
- Machine Learning : profiling attacks $\approx$ classification task
- Generative model approach: Template Attacks
- Discriminative model approach:
  - Neural Networks, big data scalability
  - CNN to integrate resynchronization in a unique model construction process

Today and in the future

- Going towards a community "ML for Embedded Security Analysis" (A whole session at WRAC'H today!)

Conclusions

- Curse of dimensionality affects the potential optimality of profiling attacks
- Machine Learning : profiling attacks $\approx$ classification task
- Generative model approach: Template Attacks
- Discriminative model approach:
  - Neural Networks, big data scalability
  - CNN to integrate resynchronization in a unique model construction process

Today and in the future

- Going towards a community "ML for Embedded Security Analysis"
  (A whole session at WRAC'H today!)
  (since ASCAD publication $\sim$ 15 published papers in which ASCAD is used as benchmark)

Conclusions

- Curse of dimensionality affects the potential optimality of profiling attacks
- Machine Learning : profiling attacks $\approx$ classification task
- Generative model approach: Template Attacks
- Discriminative model approach:
  - Neural Networks, big data scalability
  - CNN to integrate resynchronization in a unique model construction process

Today and in the future

- Going towards a community "ML for Embedded Security Analysis"
  (A whole session at WRAC'H today!)
  (since ASCAD publication $\sim$ 15 published papers in which ASCAD is used as benchmark)
- Beyond Classification

Conclusions

- Curse of dimensionality affects the potential optimality of profiling attacks
- Machine Learning : profiling attacks $\approx$ classification task
- Generative model approach: Template Attacks
- Discriminative model approach:
  - Neural Networks, big data scalability
  - CNN to integrate resynchronization in a unique model construction process

Today and in the future

- Going towards a community "ML for Embedded Security Analysis"
  (A whole session at WRAC'H today!)
  (since ASCAD publication $\sim$ 15 published papers in which ASCAD is used as benchmark)
- Beyond Classification
  - Collision attacks $\approx$ verification task (siamese network)

Conclusions

- Curse of dimensionality affects the potential optimality of profiling attacks
- Machine Learning : profiling attacks $\approx$ classification task
- Generative model approach: Template Attacks
- Discriminative model approach:
  - Neural Networks, big data scalability
  - CNN to integrate resynchronization in a unique model construction process

Today and in the future

- Going towards a community "ML for Embedded Security Analysis"
  (A whole session at WRAC'H today!)
  (since ASCAD publication $\sim$ 15 published papers in which ASCAD is used as benchmark)
- Beyond Classification
  - Collision attacks $\approx$ verification task (siamese network)
  - Does "accuracy" matter? Need for specifying a proper "Advanced-attack-oriented machine learning task" (SCA-specific loss functions and metrics)

Thank You!

- Eleonora Cagli, Cécile Dumas, Emmanuel Prouff: *Convolutional Neural Networks with Data Augmentation against Jitter-Based Countermeasures - Profiling Attacks without Pre-Processing -.* IACR Cryptology ePrint Archive 2017: 740 (2017) - CHES 2017:45-68
- Emmanuel Prouff, Remi Strullu, Ryad Benadjila, Eleonora Cagli, Cécile Dumas: *Study of Deep Learning Techniques for Side-Channel Analysis and Introduction to ASCAD Database.* IACR Cryptology ePrint Archive 2018: 53 (2018) `https://github.com/ANSSI-FR/ASCAD`
- Loïc Masure, Cécile Dumas, Emmanuel Prouff: *Gradient Visualization for General Characterization in Profiling Attacks* (COSADE 2019)

References I

[Arc+06] C. Archambeau et al. “Template Attacks in Principal Subspaces”. English. In: *Cryptographic Hardware and Embedded Systems - CHES 2006*. Ed. by Louis Goubin and Mitsuru Matsui. Vol. 4249. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 1–14. ISBN: 978-3-540-46559-1. DOI: 10.1007/11894063_1. URL: http://dx.doi.org/10.1007/11894063_1.

[BDP10] Martin Bär, Hermann Drexler, and Jürgen Pulkus. “Improved template attacks”. In: *COSADE2010* (2010).

[BHW12] Lejla Batina, Jip Hogenboom, and Jasper G.J. van Woudenberg. “Getting More from PCA: First Results of Using Principal Component Analysis for Extensive Power Analysis”. English. In: *Topics in Cryptology CT-RSA 2012*. Ed. by Orr Dunkelman. Vol. 7178. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 383–397. ISBN: 978-3-642-27953-9. DOI: 10.1007/978-3-642-27954-6_24. URL: http://dx.doi.org/10.1007/978-3-642-27954-6_24.

References II

[Bat+11] Lejla Batina et al. "Mutual information analysis: a comprehensive study". In: *Journal of Cryptology* 24.2 (2011), pp. 269–291.

[Bha+14] Shivam Bhasin et al. "Side-channel leakage and trace compression using normalized inter-class variance". In: *Proceedings of the Third Workshop on Hardware and Architectural Support for Security and Privacy*. ACM. 2014, p. 7.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. "Correlation power analysis with a leakage model". In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2004, pp. 16–29.

[Bru+15] Nicolas Bruneau et al. "Less is more". In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2015, pp. 22–41.

References III

[CRR03] Suresh Chari, JosyulaR. Rao, and Pankaj Rohatgi. "Template Attacks". English. In: *Cryptographic Hardware and Embedded Systems - CHES 2002*. Ed. by Burton S. Kaliski, Cetin K. Koc, and Christof Paar. Vol. 2523. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, pp. 13–28. ISBN: 978-3-540-00409-7. DOI: 10.1007/3-540-36400-5_3. URL: http://dx.doi.org/10.1007/3-540-36400-5_3.

[Cho+15] François Chollet et al. *Keras*. https://github.com/fchollet/keras. 2015.

[CK14] Omar Choudary and Markus G Kuhn. "Efficient template attacks". In: *Smart Card Research and Advanced Applications*. Springer, 2014, pp. 253–270.

[Dur+15] François Durvaux et al. "Efficient selection of time samples for higher-order DPA with projection pursuits". In: *Constructive Side-Channel Analysis and Secure Design*. Springer, 2015, pp. 34–50.

References IV

[GLRP06] Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. "Templates vs. stochastic methods". In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2006, pp. 15–29.

[Gie+08] Benedikt Gierlichs et al. "Mutual information analysis". In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2008, pp. 426–442.

[He+16] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[HGG19] Benjamin Hettwer, Stefan Gehrer, and Tim Gneysu. *Deep Neural Network Attribution Methods for Leakage Analysis and Symmetric Key Recovery*. 143. 2019. URL: `https://eprint.iacr.org/2019/143` (visited on 02/21/2019).

References V

[HZ12] Annelie Heuser and Michael Zohner. “Intelligent Machine Homicide”. English. In: *Constructive Side-Channel Analysis and Secure Design*. Ed. by Werner Schindler and SorinA. Huss. Vol. 7275. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 249–264. ISBN: 978-3-642-29911-7. DOI: 10.1007/978-3-642-29912-4_18. URL: http://dx.doi.org/10.1007/978-3-642-29912-4_18.

[Hos+11] Gabriel Hospodar et al. “Machine learning in side-channel analysis: a first study”. English. In: *Journal of Cryptographic Engineering* 1.4 (2011), pp. 293–302. ISSN: 2190-8508. DOI: 10.1007/s13389-011-0023-x. URL: http://dx.doi.org/10.1007/s13389-011-0023-x.

[IS15] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *CoRR* abs/1502.03167 (2015). URL: http://arxiv.org/abs/1502.03167.

References VI

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. "Differential power analysis". In: *Annual International Cryptology Conference*. Springer. 1999, pp. 388–397.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. 2012, pp. 1106–1114.

[LBM15] Liran Lerman, Gianluca Bontempi, and Olivier Markowitch. "A machine learning approach against a masked AES". In: *Journal of Cryptographic Engineering* 5.2 (2015), pp. 123–139.

[LBM14] Liran Lerman, Gianluca Bontempi, and Olivier Markowitch. "Power analysis attack: an approach based on machine learning". In: *International Journal of Applied Cryptography* 3.2 (2014), pp. 97–115.

References VII

[MOP08] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks: Revealing the secrets of smart cards*. Vol. 31. Springer Science & Business Media, 2008.

[MDM16] Zdenek Martinasek, Petr Dzurenda, and Lukas Malina. "Profiling power analysis attack based on MLP in DPA contest V4. 2". In: *Telecommunications and Signal Processing (TSP), 2016 39th International Conference on*. IEEE. 2016, pp. 223–226.

[MHM13] Zdenek Martinasek, Jan Hajny, and Lukas Malina. "Optimization of power analysis using neural network". In: *International Conference on Smart Card Research and Advanced Applications*. Springer. 2013, pp. 94–107.

[MMT15] Zdenek Martinasek, Lukas Malina, and Krisztina Trasy. "Profiling power analysis attack based on multi-layer perceptron network". In: *Computational Problems in Science and Engineering*. Springer, 2015, pp. 317–339.

References VIII

[MZ13] Zdenek Martinasek and Vaclav Zeman. “Innovative method of the power analysis”. In: *Radioengineering* 22.2 (2013), pp. 586–594.

[OC14] Colin O’Flynn and Zhizhang David Chen. “ChipWhisperer: An open-source platform for hardware embedded security research”. In: *Constructive Side-Channel Analysis and Secure Design*. Springer, 2014, pp. 243–260.

[Pro+18] Emmanuel Prouff et al. *Study of Deep Learning Techniques for Side-Channel Analysis and Introduction to ASCAD Database*. Cryptology ePrint Archive, Report 2018/053. https://eprint.iacr.org/2018/053. 2018.

[SVZ13] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”. In: *arXiv:1312.6034 [cs]* (Dec. 20, 2013). arXiv: 1312.6034. URL: http://arxiv.org/abs/1312.6034 (visited on 09/07/2018).

References IX

[SZ14] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[Spr+14] Jost Tobias Springenberg et al. "Striving for Simplicity: The All Convolutional Net". In: *arXiv:1412.6806 [cs]* (Dec. 21, 2014). arXiv: 1412.6806. URL: http://arxiv.org/abs/1412.6806 (visited on 09/07/2018).

[SA08] François-Xavier Standaert and Cedric Archambeau. "Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages". English. In: *Cryptographic Hardware and Embedded Systems CHES 2008*. Ed. by Elisabeth Oswald and Pankaj Rohatgi. Vol. 5154. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 411–425. ISBN: 978-3-540-85052-6. DOI: 10.1007/978-3-540-85053-3_26. URL: http://dx.doi.org/10.1007/978-3-540-85053-3_26.

[TM97] Mitchell T. M. *Machine Learning*. McGraw-Hill, New York, 1997.

References X

[Tim19] Benjamin Timon. "Non-Profiled Deep Learning-based Side-Channel attacks with Sensitivity Analysis". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (Feb. 28, 2019), pp. 107–131. ISSN: 2569-2925. DOI: 10.13154/tches.v2019.i2.107-131. URL: https://tches.iacr.org/index.php/TCHES/article/view/7387 (visited on 03/25/2019).