Make Some Noise
Unleashing the Power of Convolutional Neural Networks for Profiled Side-channel Analysis

Jaehun Kim[1] Stjepan Picek[1] Annelie Heuser[2]
Shivam Bhasin[3] Alan Hanjalic[1]

Delft University of Technology, Delft, The Netherlands

Univ Rennes, Inria, CNRS, IRISA, France

Physical Analysis and Cryptographic Engineering,
Temasek Laboratories at Nanyang Technological University, Singapore

CHES 2019, Atlanta, August 28, 2019

Outline

Outline

What and Why

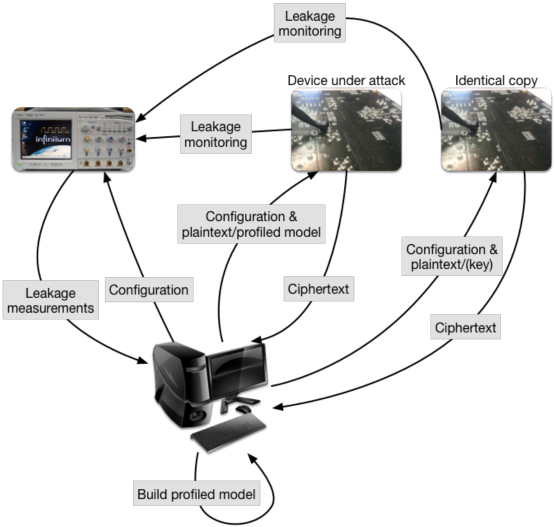
- Investigate the limits of CNNs' performance when considering side-channel analysis.
- Propose new CNN instance (and justify the choice for that instance).
- Investigate how additional, non-task-specific noise can significantly improve the performance of CNNs in side-channel analysis.
- Show how small changes in setup result in big differences in performance.

Outline

Profiled Attacks

- Side-channel attacks (SCAs) are passive, non-invasive implementation attacks.
- Profiled attacks have a prominent place as the most powerful among side channel attacks.
- Within profiling phase the adversary estimates leakage models for targeted intermediate computations, which are then exploited to extract secret information in the actual attack phase.

Profiled Attacks



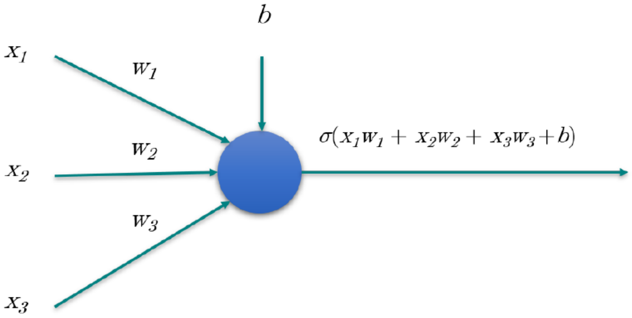
Profiled Attacks

- Template Attack (TA) is the most powerful attack from the information theoretic point of view.
- Some machine learning (ML) techniques also belong to the profiled attacks.
- Deep learning has been shown to be able to reach top performance even if the device is protected with countermeasures.

Outline

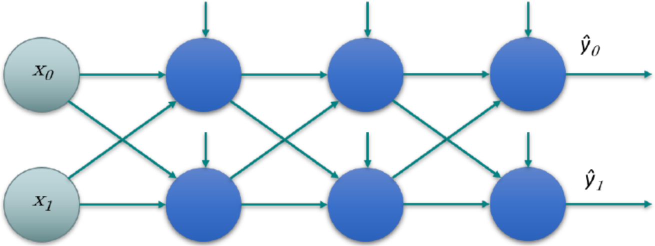
Deep Learning

- Let us build a neural network.

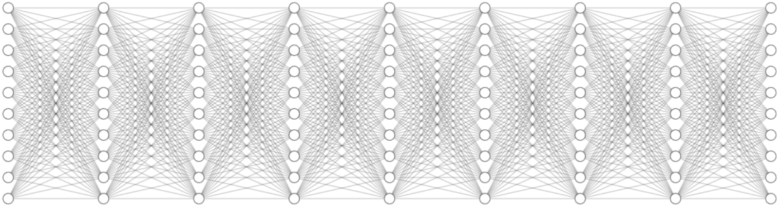


Deep Learning

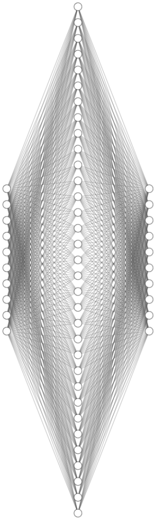
- Let us continue adding neurons.



Multilayer Perceptron - “Many” Hidden Layers



Multilayer Perceptron - One Hidden Layer



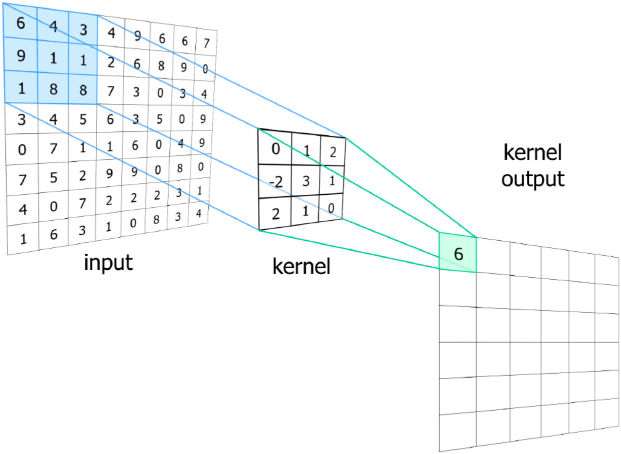
Universal Approximation Theorem

- A feed-forward network with a single hidden layer containing a finite number of neurons can approximate continuous functions on compact subsets of $\mathbb{R}^n$.
- Given enough hidden units and enough data, multilayer perceptrons can approximate virtually any function to any desired accuracy.
- Valid results if and only if there is a sufficiently large number of training data in the series.

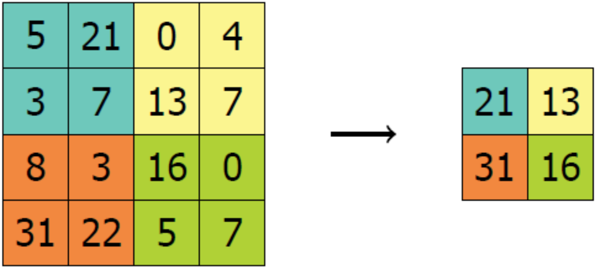
Convolutional Neural Networks

- CNNs represent a type of neural networks which were first designed for 2-dimensional convolutions.
- They are primarily used for image classification but lately, they have proven to be powerful classifiers in other domains.
- From the operational perspective, CNNs are similar to ordinary neural networks: they consist of a number of layers where each layer is made up of neurons.
- CNNs use three main types of layers: convolutional layers, pooling layers, and fully-connected layers.

Convolutional Neural Networks - Convolution Layer



Convolutional Neural Networks - Pooling



Design Principle - VGG Like CNN

- Small kernel size: 3×3 for every layer.
- Max pooling with 2 x 2 windows, with stride 2.
- Increasing number of filters per layer: doubled after every max pooling layer.
- Convolutional blocks are added until the spatial dimension is reduced to 1.
- After the fully connected layers is the output layey.
- The convolutional and fully connected layers use ReLu activations, the output layer uses Softmax to normalize the predictions.

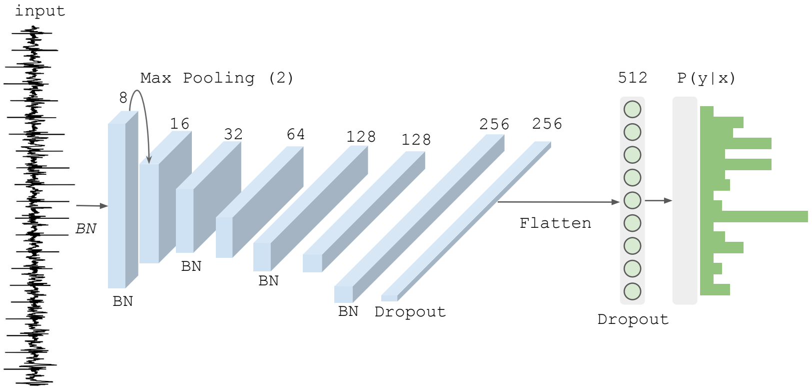
Design Principle - VGG Like CNN

$$\mathtt{net} = \mathtt{fc}_{\theta,\mathtt{softmax}} \circ \prod_{p=1}^{P} \mathtt{fc}_{\theta^p,\mathtt{ReLU}} \circ \prod_{q=1}^{Q} \left(\mathtt{pool}_{\mathtt{Max}} \circ \prod_{r=1}^{R_q} \mathtt{conv}_{\phi^r,\mathtt{ReLU}}\right), \tag{1}$$

$$\mathtt{conv}_{\phi,\sigma}(X) = \sigma(\phi * X), \tag{2}$$

$$\mathtt{fc}_{\theta,\sigma}(x) = \sigma(\theta^\intercal x). \tag{3}$$

Convolutional Neural Networks - Final



Outline

Why Adding Noise

- To reduce the overfitting of the model, we introduce noise to the training phase.
- Since in our case, the input normalization is also learned during the training process via the BN layer, we added the noise tensor after the first BN.

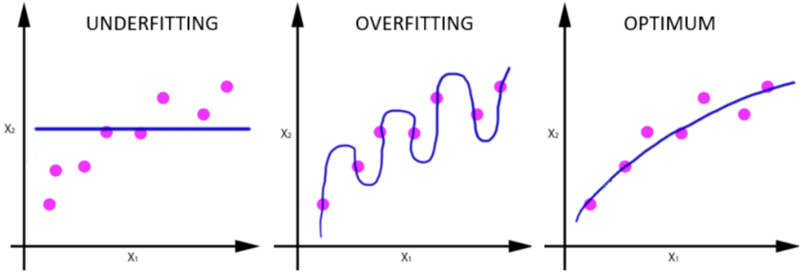
$$X^* = BN_0(X) + \Psi, \quad \Psi \sim \mathcal{N}(0, \alpha). \tag{4}$$

- The noise tensor follows the normal distribution.

Data Augmentation

- Is this data augmentation?
- Data augmentation typically applies domain knowledge to deform the original signal into more “plausible” way and uses both the original and transformed measurements in the training phase.
- Our technique is a regularization technique and can be seen as 1) a noisy training and 2) data augmentation.
- We are closer to the noisy training as we neither: 1) add transformed measurements and 2) use domain-specific knowledge.

Underfitting and Overfitting

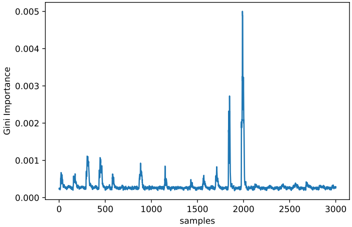


Outline

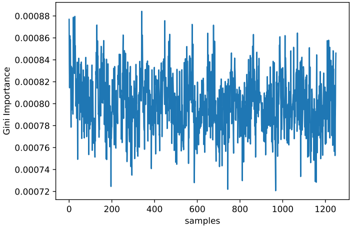
Results

- We consider only the intermediate value model (256 classes).
- We experiment with CNNs, template attack, and pooled template attack.
- 4 publicly available datasets.

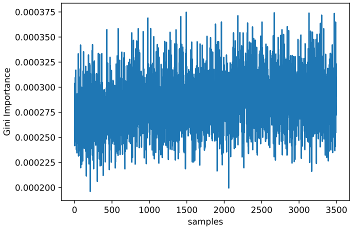
Datasets

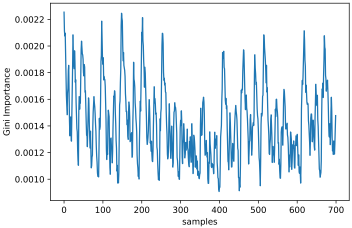


(a) DPAcontest v4 dataset.

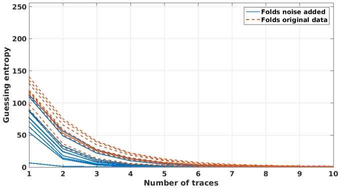


(b) AES_HD dataset.

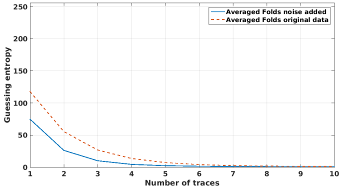




Results DPAv4

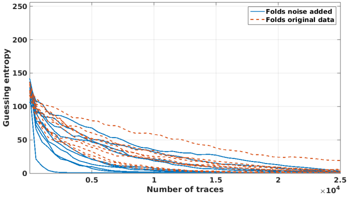


(a) RD network per fold.

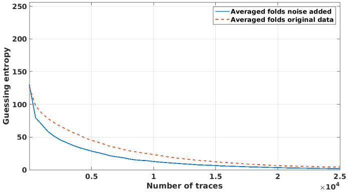


(b) RD network averaged.

Results AES_HD

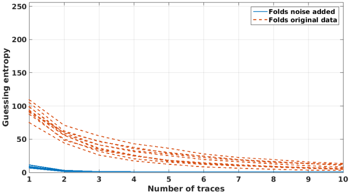


(a) ASCAD network per fold.

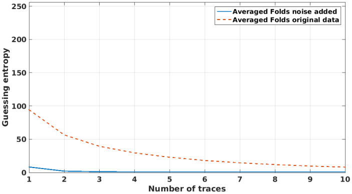


(b) ASCAD network averaged.

Results AES_RD

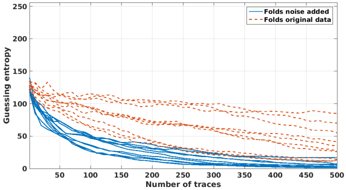


(a) RD network per fold.

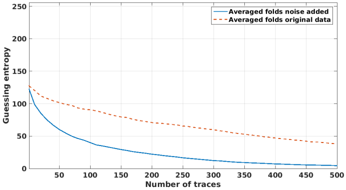


(b) RD network averaged.

Results ASCAD



(a) ASCAD network per fold.

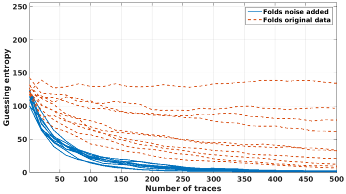


(b) ASCAD network averaged.

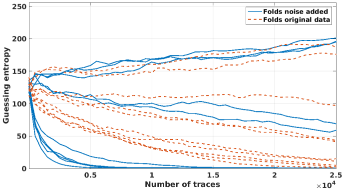
What Else Do We Demonstrate

- We show noise addition to be quite stable over different levels of noise, number of epochs, and profiling set sizes.
- Our results indicate that it is not possible to have a single best CNN architecture even if considering "only" SCA.
- Attacking a dataset without countermeasure could be more difficult than attacking one that has countermeasures.
- What is really a new CNN architecture and what is simply a new instance in accordance to the input data?
- The less traces we have in the profiling phase, the more noise we need.

Beware of the Choice of the Profiling Set



(a) ASCAD network per fold.



(b) RD network per fold.

Outline

Conclusions

- VGG-like CNNs seem to work very good for SCA.
- There are other domains that use machine learning/deep learning and we can learn a lot from them.
- Here, by using such good practices, we are able to reach top performance in SCA.
- We propose to add noise addition as a standard technique in the SCA evaluation for deep learning techniques.

Questions?

Thanks for your attention! Q?