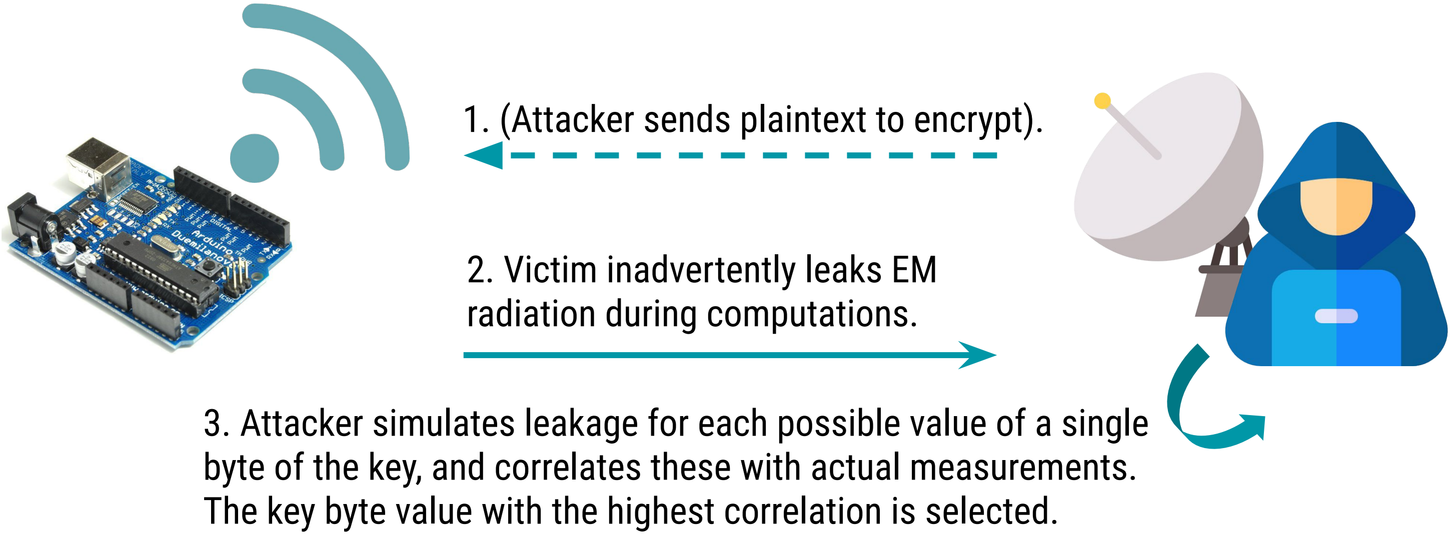
Improving CEMA using Correlation Optimization

Pieter Robyns Peter Quax Wim Lamotte

Introduction and motivation

Introduction

- Electromagnetic (EM) side-channel attacks
  - Possible when EM leakage differs between key-dependent operations
  - In this presentation: CEMA attack on AES
  - Uses Pearson correlation as metric to compare leakage vs. hypothesis key



Introduction: CEMA attack

- For n_m encryption measurements x_t of key byte s:

$$H_{sj}^{(1,2,\ldots n_m)} = HW(sbox(p_s \oplus j))$$

Hamming Weight (HW) leakage model

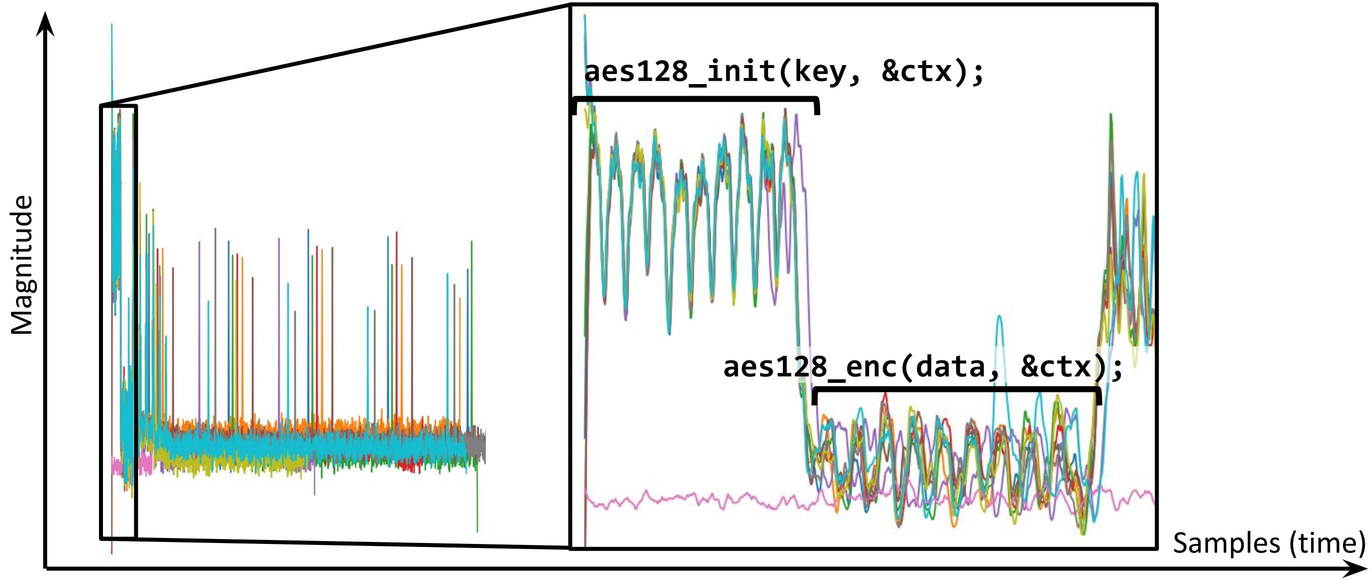
$$H_{s0}^{(1,2,\ldots n_m)} = HW(sbox(p_s \oplus \texttt{0x00}))$$
$$H_{s1}^{(1,2,\ldots n_m)} = HW(sbox(p_s \oplus \texttt{0x01}))$$
$$\vdots$$
$$H_{s255}^{(1,2,\ldots n_m)} = HW(sbox(p_s \oplus \texttt{0xff}))$$

Simulate leakage for each possible key byte value j

$$\rho_{x_t, H_{sj}} = \frac{cov(x_t, H_{sj})}{\sigma_{x_t} \sigma_{H_{sj}}}$$

Motivation

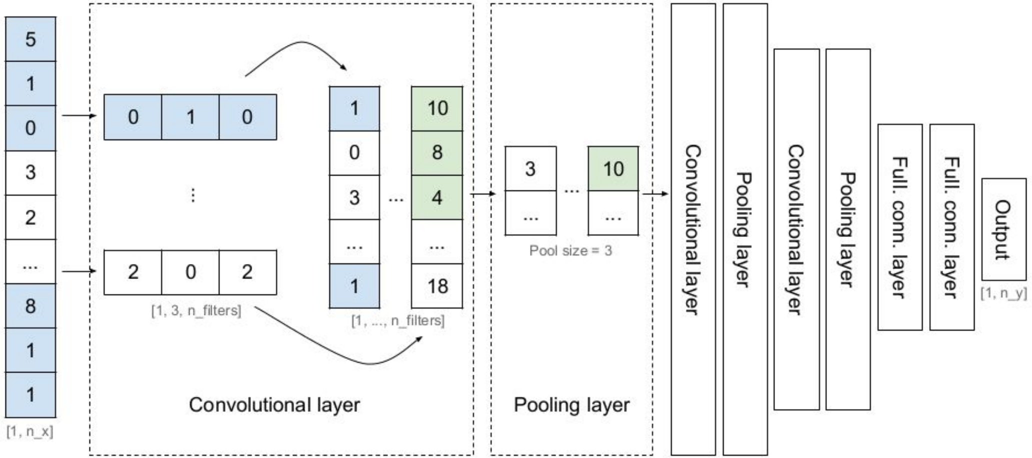
- Recent advances in machine learning and deep learning
  - Outperform classical methods for pattern recognition in other domains [1]
    - → Can we apply this to SCA to improve leakage detection in noisy, high-dimensional signals?
    - → Already some promising results in recent related works [2,3,4]



[1] https://www.nature.com/articles/nature14539 [2] https://eprint.iacr.org/2018/053 [3] https://eprint.iacr.org/2017/740.pdf
[4] https://i.blackhat.com/us-18/Thu-August-9/us-18-perin-ege-vanwoudenberg-Lowering-the-bar-Deep-learning-for-side-channel-analysis-wp.pdf

Motivation

- Previous works: CNN classification of fixed set of classes
  - Output of CNN is probability distribution for the (inter.) value of a key byte
    - → Optimized using average cross entropy loss to match true probability distribution
    - → Typically: attack 1 key byte and predict probability of (intermediate) value (256 classes)
      - Alternatively: predict probability of key byte Hamming weight (9 classes)
    - → Then, to attack entire key: train multiple networks



Contributions in our work

- "Correlation Optimization" approach
  - Inspired by recent works related to face recognition [5]
  - Idea is to not use classification, but **learn representation / encoding** of the signal that is correlated with the true leakage value
    - → Optimized using "correlation loss function" (a.k.a. cosine proximity)
  - This encoding consists of only one value per key byte
    - → Number of outputs reduced by factor 9 (HW classification) or 256 (byte classification)
    - → Trivial to learn model for entire key instead of just 1 byte
    - → However, we do need to perform a standard CEMA attack on the outputs
      - Fortunately, this is **fast** since we only need to attack 16 points for a 16-byte key

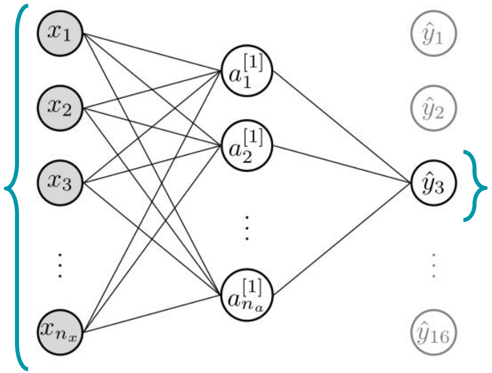
- Methodology to remove alignment requirement
  - By applying correlation optimization in the frequency domain

[5] https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Schroff_FaceNet_A_Unified_2015_CVPR_paper.pdf

Correlation Optimization

- Example for one byte of the key and 5 traces
  - Suppose the true HW values of $sbox(p_s \oplus k_s)$ are: [5. 6. 7. 5. 1.]

5 input traces



$$\mathcal{L}(\hat{y}_k, y_k) = 1 - \frac{\hat{y}_k \cdot y_k}{||\hat{y}_k|| \cdot ||y_k|| + \epsilon}$$

5 output encodings after training:

$$\underbrace{\begin{matrix}[\,0.2059 & 0.3877 & 0.5690 & 0.2057 & -0.4889] & \textit{or scaled e.g.} \\ [\,20.59 & 38.77 & 56.90 & 20.57 & -48.89] & \end{matrix}}$$

➔ Both have correlation 0.9999 with the true Hamming Weights
➔ "Useless" points of the input traces are discarded

Removing the trace alignment requirement

- Simple networks such as MLPs are sensitive to feature translations
  - ⇒ Use magnitude / power spectrum of Fourier transform as features
  - Similar idea applied in DEMA context by Tiu et al. [6]

- Why does this work?
  - Demo: https://research.edm.uhasselt.be/probyns/fft_phase.html

[6] http://cacr.uwaterloo.ca/techreports/2005/cacr2005-13.pdf

Results

Results

- Two experiments
  - Comparison to SCAnet-based model on ASCAD dataset (protected AES)
  - Attack noisy, unaligned Arduino traces recorded with SDR (unprotected AES)
    - → Measured at our research lab
    - → Also released to public domain

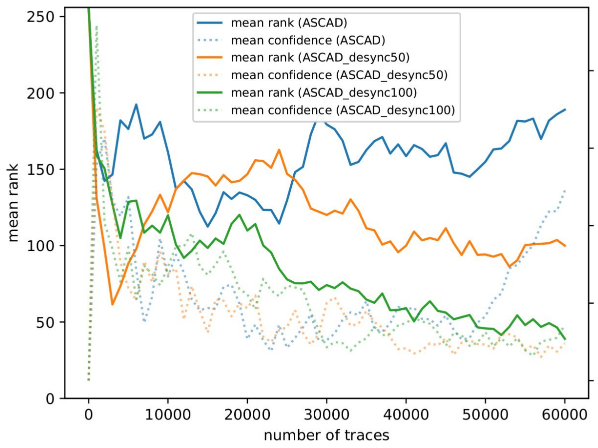
- Outperforms previous deep learning models (8-layer CNN) using only a very simple architecture (2-layer MLP)

ASCAD dataset

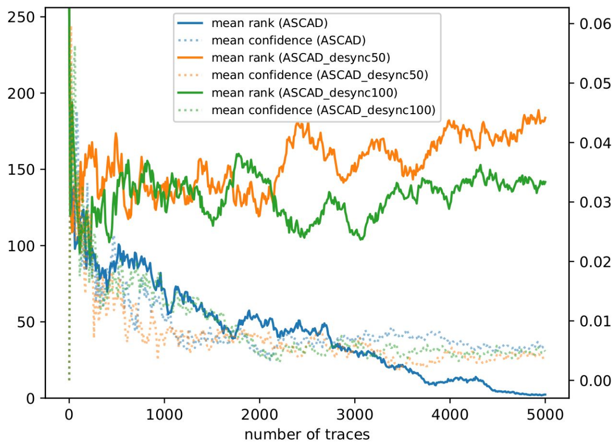
- Introduced by Prouff et al. in [2]
- AES protected against first-order side-channel attacks
- 50,000 training / 10,000 test traces of 700 samples, collected at 2 GS/s from ATMega8515
  - ASCAD: time-aligned traces in preprocessing step
  - ASCAD_desync50: desynced traces with maximum jitter of 50 samples
  - ASCAD_desync100: desynced traces with maximum jitter of 100 samples

ASCAD experiment (time domain)

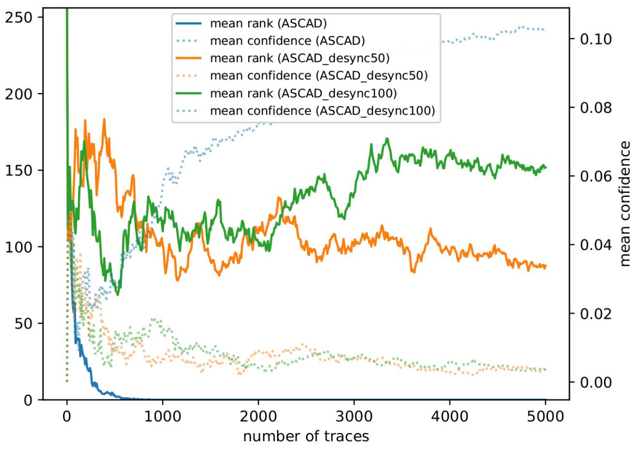
Regular CEMA



1-layer MLP



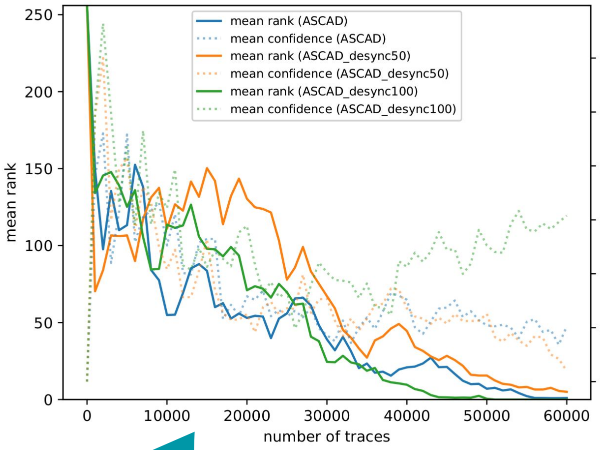
2-layer MLP



For the aligned traces (blue line), there is a clear improvement over regular CEMA. However, MLPs are very sensitive to misaligned traces (orange and green lines).

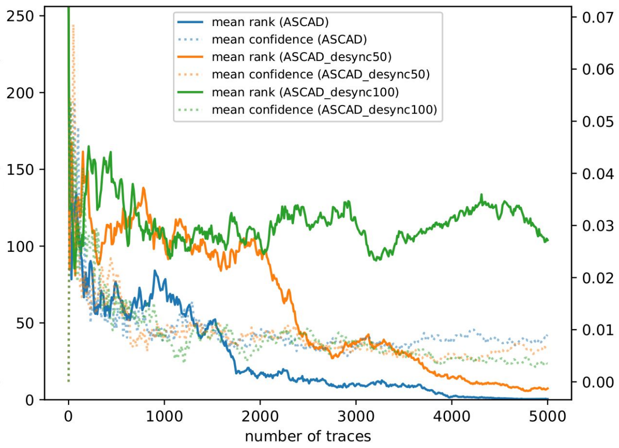
ASCAD experiment (frequency domain)

Regular CEMA

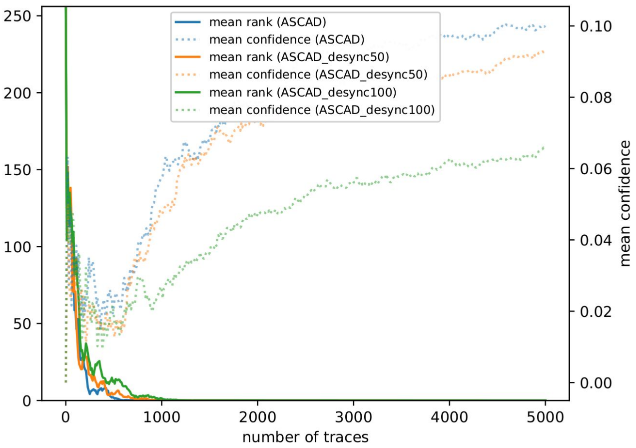


Surprising result

1-layer MLP



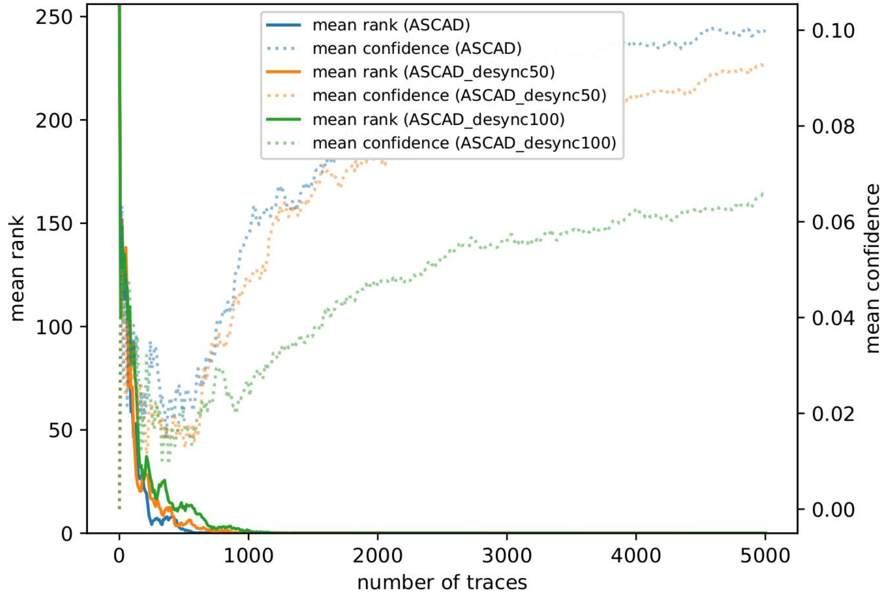
2-layer MLP



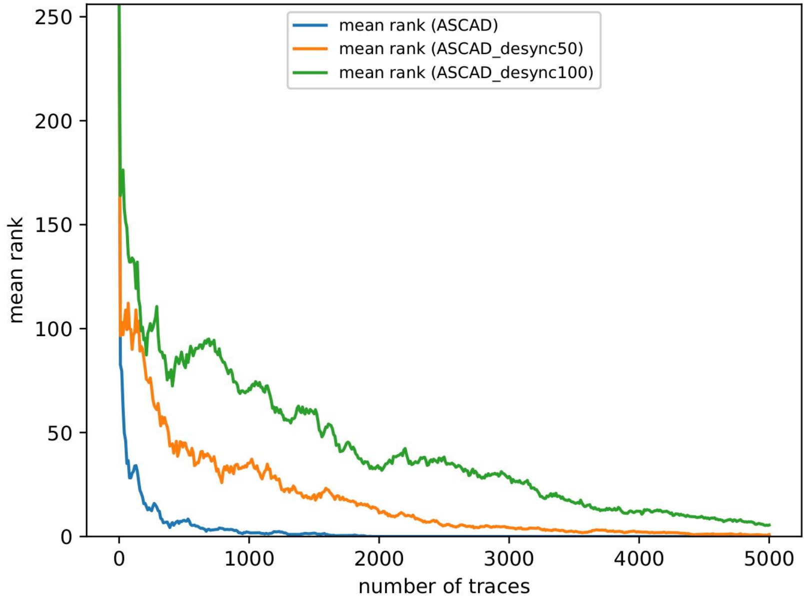
Using frequency-domain features, the 2-layer MLP finds the correct key in ~1,000 traces for each of the ASCAD datasets

ASCAD experiment (comparison to previous work)

2-layer MLP

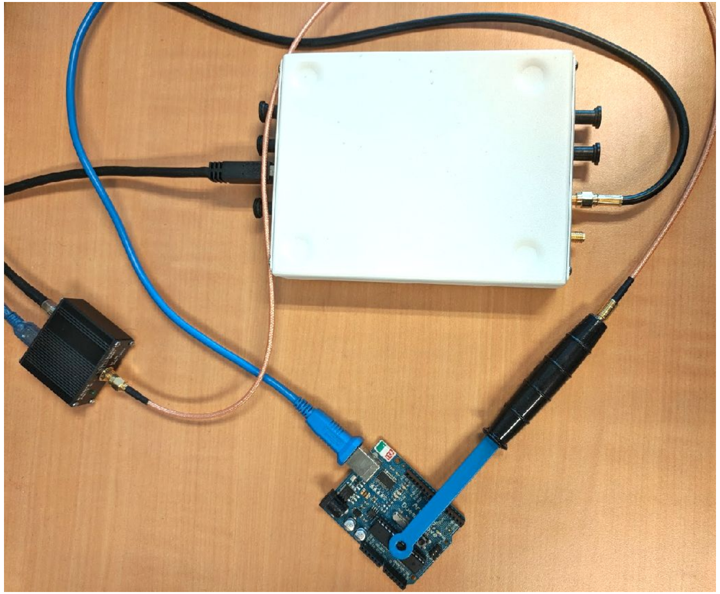


best_cnn model from previous work [2]

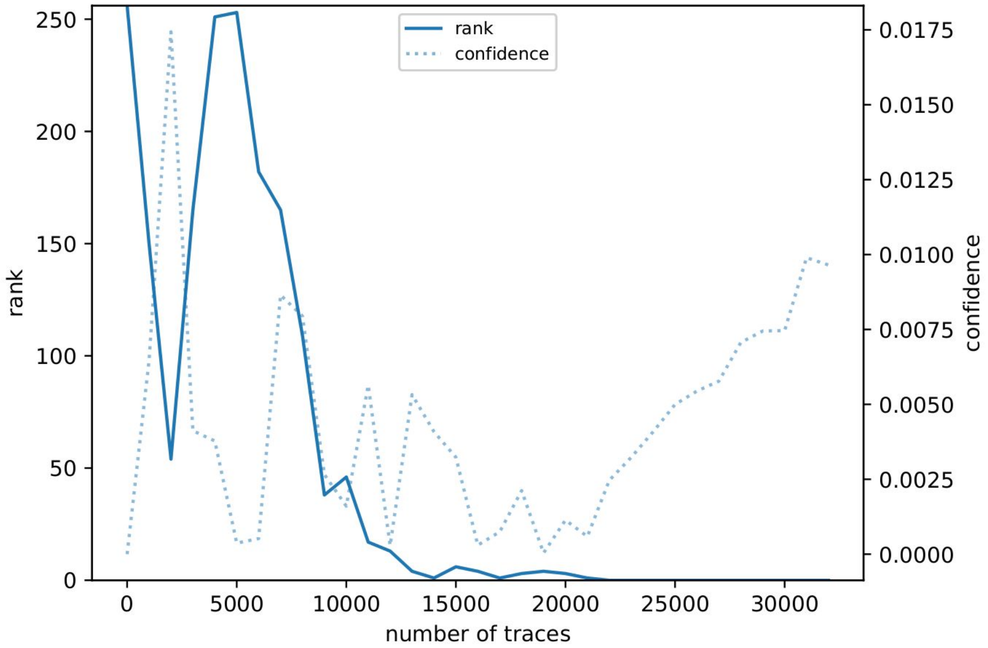


Arduino Duemilanove + SDR experiment

- USRP B210 and TBPS01 + TBWA2 to capture EM traces
  - Training set: 51,200 traces of uniform random key encryptions
  - Validation set: 32,768 traces of fixed-key encryptions
  - Sample rate of 8 MS/s
  - No preprocessing / alignment



Attack against Arduino Duemilanove (unprotected AES)



Note: no 10-fold cross-validation applied as in previous figures

Correct key found in ~22,000 traces using frequency-domain 2-layer MLP model.

Conclusions and future work

Conclusions

- We've demonstrated the usage of ML as a means for feature extraction (encodings) rather than classification
- Features are extracted by optimizing the correlation loss
- On the ASCAD dataset, we achieve better performance despite using only a shallow MLP architecture
- Alignment issues can be resolved by operating in the frequency domain
- All code and data is open source: https://github.com/rpp0/correlation-optimization-paper

Future work

- Siamese networks → triplet loss (see [5])
- Applications to other crypto algorithms
- Improvements to existing benchmark datasets
  - ASCAD uses fixed key (fortunately variable masking values)
- Implement state-of-the-art architectures from CV domain
  - For example: ResNets

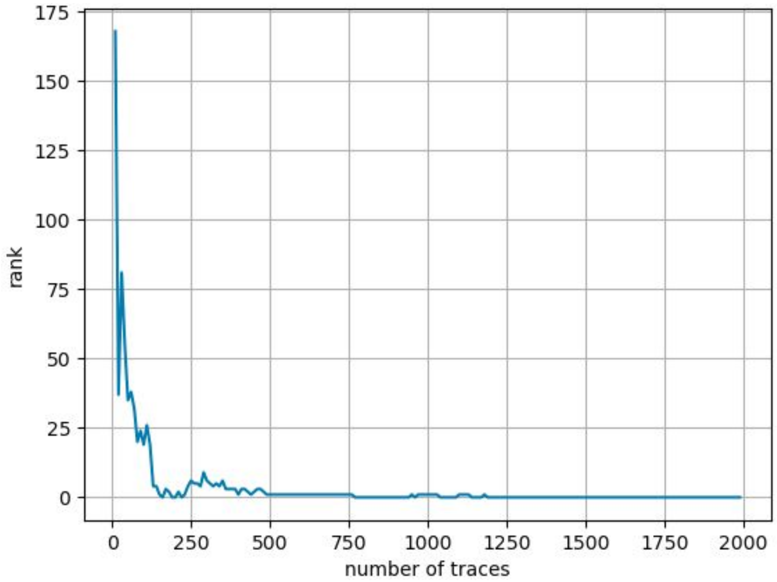
Questions?

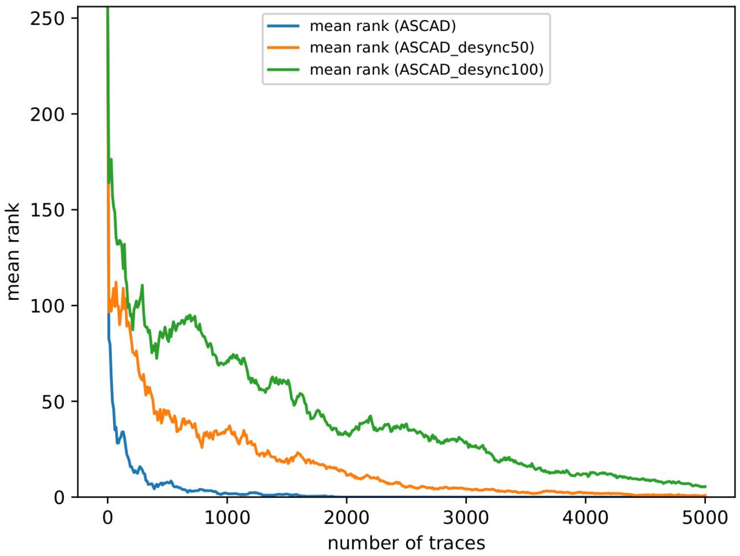
pieter.robyns@uhasselt.be

Extra slides

Reproducing best_cnn results

- Complete retrain of best_cnn model





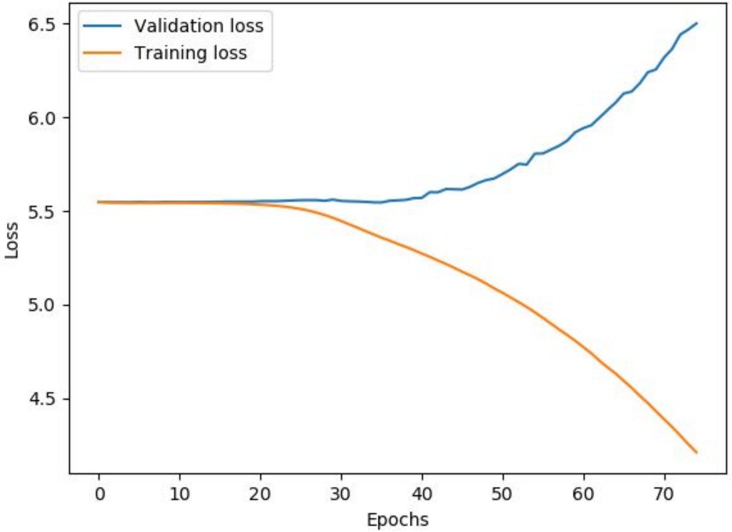
- For desync50 and desync100 results are identical. Small difference (~500-1,000 traces) for desync0 → could be due to lesser number of training examples used (45,000)*?

* Their paper states that 45,000 training examples were used (page 9), whereas their implementation actually uses 50,000 training examples. We decided to use 45,000 traces for all experiments in our paper.

Reproducing best_cnn results

- ASCAD paper code (Github): no validation set used
  - When added: validation loss actually increases over time → it overfits!
    - → However, rank still decreases in both cases below
  - Possible reason: multiple labels should actually be 1 since only HW leaks?

cross-entropy loss used in ASCAD paper



correlation loss used in our work