Give Me 5 Minutes

Attacking ASCAD with a Single Side-Channel Trace

Olivier Bronchain, Gaëtan Cassiers and François-Xavier Standaert

ICTEAM Institute, UCLouvain, Louvain-la-Neuve, Belgium.
{olivier.bronchain,gaetan.cassiers,fstandae}@uclouvain.be

Abstract. In this note, we describe an attack against the ANSSI Side-Channel Analysis Database (ASCAD), which recovers the full key using the leakage of a single masked block cipher execution. The attack uses a new open-source Side-Channel Analysis Library (SCALib), which allows running the leakage profiling and attacking in less than 5 minutes. It exploits well-known techniques, yet improves significantly over the best known attacks against ASCAD. We conclude by questioning the impact of these experimental findings for side-channel security evaluations.

Keywords: Profiled Side-Channel Analysis · Soft Analytical Side-Channel Attacks

1 Introduction

Since 2018, ASCAD has become one of the most used datasets for side-channel analysis benchmarking [BPS+20]. It contains power measurements of a two-share masked AES implementation running on an 8-bit micro-controller. As intended by the authors, most of the published attacks against ASCAD use Deep Learning (DL) – we refer to the publications using this dataset at CHES 2020 for illustration [ZBHV20, MDP20, WAGP20, WP20, PCP20, HHO20, ZZN+20, ZDF20]. Most of these works focus on recovering a single key byte by looking at a small (possibly desynchronized) part of the leakage traces, and do not exploit the knowledge of the masking randomness during their profiling stage. An exception is the recent work of Xiangjun Lu et al. which we discuss in conclusion [LZC+21].

In this note, we take another approach and use the recently introduced SCALib to analyze ASCAD. The proposed attack efficiently takes advantage of all the time samples in the leakage traces (next called raw traces) and leverages the knowledge of the masking randomness during the profiling phase. It mixes Signal to Noise Ratio (SNR) computations, Linear Discriminant Analysis (LDA) [SA08] and Soft Analytical Side-Channel Attacks (SASCA) [VGS14]. As a result, we are able to efficiently recover a full encryption key by exploiting the leakage of a single masked block cipher execution.[1]

2 SCALib

The Side-Channel Analysis Library (SCALib) is an open-source library that provides all the tools needed to run our attack. SCALib is a python package available on PyPI that provides optimized implementations of a series of algorithms for side-channel analysis. The focus of this library is on performance (both single-core and multi-threaded) and usability. SCALib is hosted at https://github.com/simple-crypto/SCALib and the documentation lives at https://scalib.readthedocs.io/en/stable/.

[1] The full key is 14 bytes since the 2 first bytes of the AES key are not masked in ASCAD.

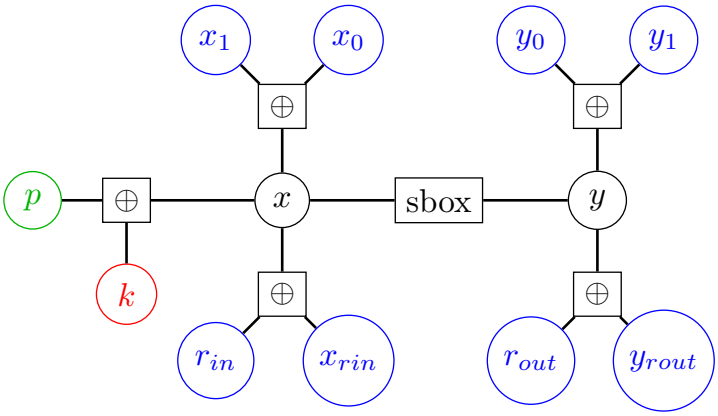

Figure 1: Single byte graph.

3 Attack methodology

The methodology used to mount the attack is similar to the one proposed by Bronchain and Standaert in [BS20, BS21]. Namely, the profiling goes in two steps:

1. The **SNR** is computed for all the shares within the implementation [Man04]. The samples with largest SNR are taken as Points-Of-Interest (POIs) for each share. We kept 500 POIs per share out of the 250,000 traces' samples.

2. **Models** are then built for each share thanks to LDA, with limitation to a small subspace. LDA can be viewed as a (Gaussian) template attack embedding a dimensionality reduction as pre-processing. We kept 10 dimensions per share. The model of each share is computed only from the corresponding POIs.

The attack phase uses SASCA. For each key byte, it goes in three sequential steps:

1. **Graph description** consists in representing the implementation with a factor graph including operations and variables. The graph used in this note is represented in Figure 1 and follows the guidelines of [BS21]. More precisely, each shared intermediate variables is unmasked. As an example, the Sbox output y is derived from its shares y_i such that $y = y_0 \oplus y_1$. Then the circuit is represented with operations on the unshared variables. In Figure 1, the shares are in blue, the plaintext is in green and the key byte to recover is in red. A rectangle stands for an operation and a circle for a variable. If multiple traces are used for the attack, the graph is replicated once for each trace, with the k node being common across all replicas.

2. **Probabilities extraction** consists in using the models to obtain the distribution $(\Pr[x_i = \alpha|\mathbf{l}])_{\alpha \in \mathbb{F}_{256}}$ of the shares x_i given the leakage $\mathbf{l}$. Then, this probability is inserted in the factor graph as the initial distribution of these shares.

3. **Belief propagation** is a message passing rule between the nodes of the factor graph [VGS14]. It allows us to estimate the probability of a key byte (i.e., k in Figure 1) based on the initial distribution of the shares.

Evaluation. To evaluate the attack, we compute the rank of the correct key based on the probabilities retrieved by the attack for each key byte. SCALib provides an implementation of the histogram-based rank estimation from CHES 2016 [PSG16].

Implementation. The implementation of the attack and its evaluation are both available at https://github.com/cassiersg/ASCAD-5minutes.

4 Results

We first ran our attack on the full traces (i.e., 250,000 time samples). We used 5000 traces for profiling (i.e., computing the SNRs and building models for) the 86 shares. The attack was then run 100 independent times using $n = 1$, then $n = 2$ attack traces.

We computed the resulting rank of the 14 byte masked key. The results are reported in Figure 2, and show that even with a single trace, the attack always succeeds. The total execution time (for both profiling and attacking) is less than 4 minutes on a 4-core laptop (16 GB RAM, Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz).

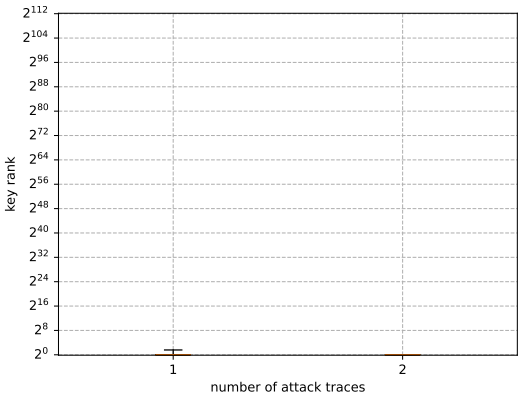

Figure 2: Attack against full ASCAD raw traces for the full (14 byte) key: boxplot of the rank of the correct key. Attacks with two traces always result in finding the correct key at rank 1. With only one attack trace, the correct key is always at very low (i.e., trivially enumerable) rank, and more than 0.75 % of the times at rank 1.

For the sake of completeness, we also report in Figure 3 an attack on the third byte exploiting only a few samples (1400) of the traces, as usually considered in previous works. As before, we run the attack 100 times for each number of attack traces. This attack uses 100,000 traces for the profiling and the total execution time is 12 seconds. We can see that if all bytes were giving similar results, 8 attack traces would be enough to perform a full key recovery attack. As expected, due to the reduction in the number of leakage points, this attack is less powerful than the one exploiting the full trace.

5 Conclusion: so what?

The full key recovery attack described in this note significantly improves the state-of-the-art, both in terms of profiling cost and in terms of online attack complexity. At high level, efficient profiling is obtained thanks to a good exploitation of the masking randomness while the improved online attack is primarily due to an analytical strategy that takes advantage of the complete (raw) traces. These results naturally raise the question of what is their impact for security evaluations. That is, should one consider that the security level of the ASCAD implementation corresponds to worst-case attacks with strong capabilities or to more relaxed attacks (e.g., without masking randomness during profiling)?

Our main conclusions in this respect are twofold. On the one hand, worst-case evaluations are significantly faster to perform and are also a useful tool to anticipate the risk of improved attacks due to continuous research developments. So following the backwards evaluation approach outlined in [ABB+20], we believe they are anyway an

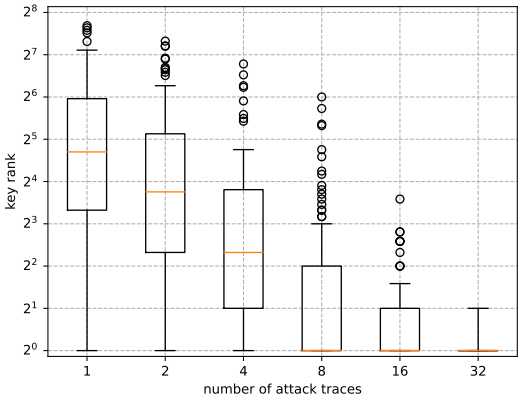

Figure 3: Attack against partial ASCAD traces for one key byte: boxplot of the rank of the correct key. Attacks with 32 traces almost always result in finding the correct key at rank 1. With only 8 traces, the correct key is at rank 1 more than half of the times.

interesting asset before launching more expensive attacks in less permissive contexts.[2] On the other hand, the gap between worst-case attacks and relaxed ones remains to be thoroughly investigated. Understanding whether it affects the profiling complexity, the online attack complexity or both (and to what extent) would help to better clarify the conservativeness/practicality of these two evaluation settings. Without such a clearly established and quantified gap, considering worst-case attacks as unpractical appears as an undesirable risk of security overstatement, especially in the long term.

We finally note that the recent work in [LZC+21] provides an interesting counterpart to our results and does a first step in quantifying the gap between worst-case and relaxed adversaries. It shows attacks against a single byte of the ASCAD implementation that succeed with 3 to 10 raw traces. Their profiling does not leverage the known masking randomness of the ASCAD traces and is therefore significantly more expansive, confirming our worst-case approach as a useful shortcut for evaluators in this case.

Acknowledgments. Gaëtan Cassiers and François-Xavier Standaert are respectively Research Fellow and Senior Research Associate of the Belgian Fund for Scientific Research (FNRS-F.R.S). This work has been funded by the ERC project SWORD (724725).

References

[ABB+20] Melissa Azouaoui, Davide Bellizia, Ileana Buhan, Nicolas Debande, Sébastien Duval, Christophe Giraud, Éliane Jaulmes, François Koeune, Elisabeth Oswald, François-Xavier Standaert, and Carolyn Whitnall. A systematic appraisal of side channel evaluation strategies. In *SSR*, volume 12529 of *Lecture Notes in Computer Science*, pages 46–66. Springer, 2020.

[BPS+20] Ryad Benadjila, Emmanuel Prouff, Rémi Strullu, Eleonora Cagli, and Cécile Dumas. Deep learning for side-channel analysis and introduction to ASCAD database. *J. Cryptogr. Eng.*, 10(2):163–188, 2020.

[2] Note that this conclusion is not specific to the choice of statistical tools we made in our experiments. For example, deep learning can be used in in a worst-case context too and could possibly lead to even more powerful attacks at the cost of a more expensive profiling. The winning (divide-and-conquer) attack of the CHES 2020 CTF is an example in this direction: https://ctf.spook.dev/submissions/. Its combination with an analytical strategy is an interesting scope for further investigations.

[BS20] Olivier Bronchain and François-Xavier Standaert. Side-channel countermeasures' dissection and the limits of closed source security evaluations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(2):1–25, 2020.

[BS21] Olivier Bronchain and François-Xavier Standaert. Breaking masked implementations with many shares on 32-bit software platforms or when the security order does not matter. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):XXX–XXX, 2021.

[HHO20] Anh-Tuan Hoang, Neil Hanley, and Máire O'Neill. Plaintext: A missing feature for enhancing the power of deep learning in side-channel analysis? breaking multiple layers of side-channel countermeasures. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):49–85, 2020.

[LZC+21] Xiangjun Lu, Chi Zhang, Pei Cao, Dawu Gu, and Haining Lu. Pay attention to the raw traces: A deep learning architecture for end-to-end profiling attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):XXX–XXX, 2021.

[Man04] Stefan Mangard. Hardware countermeasures against DPA ? A statistical analysis of their effectiveness. In *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004.

[MDP20] Loïc Masure, Cécile Dumas, and Emmanuel Prouff. A comprehensive study of deep learning for side-channel analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(1):348–375, 2020.

[PCP20] Guilherme Perin, Lukasz Chmielewski, and Stjepan Picek. Strength in numbers: Improving generalization with ensembles in machine learning-based profiled side-channel analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):337–364, 2020.

[PSG16] Romain Poussier, François-Xavier Standaert, and Vincent Grosso. Simple key enumeration (and rank estimation) using histograms: An integrated approach. In *CHES*, volume 9813 of *Lecture Notes in Computer Science*, pages 61–81. Springer, 2016.

[SA08] François-Xavier Standaert and Cédric Archambeau. Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer, 2008.

[VGS14] Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Soft analytical side-channel attacks. In *ASIACRYPT (1)*, volume 8873 of *Lecture Notes in Computer Science*, pages 282–296. Springer, 2014.

[WAGP20] Lennert Wouters, Victor Arribas, Benedikt Gierlichs, and Bart Preneel. Revisiting a methodology for efficient CNN architectures in profiling attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(3):147–168, 2020.

[WP20] Lichao Wu and Stjepan Picek. Remove some noise: On pre-processing of side-channel measurements with autoencoders. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):389–415, 2020.

[ZBHV20] Gabriel Zaid, Lilian Bossuet, Amaury Habrard, and Alexandre Venelli. Methodology for efficient CNN architectures in profiling attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(1):1–36, 2020.

[ZDF20] Ziyue Zhang, A. Adam Ding, and Yunsi Fei. A fast and accurate guessing entropy estimation algorithm for full-key recovery. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(2):26–48, 2020.

[ZZN+20] Jiajia Zhang, Mengce Zheng, Jiehui Nan, Honggang Hu, and Nenghai Yu. A novel evaluation metric for deep learning-based side channel analysis and its extended application to imbalanced data. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(3):73–96, 2020.