# SCALib: A Side-Channel Analysis Library

Gaëtan Cassiers ⬥ [2,1*¶] and Olivier Bronchain ⬥ [1*]

1 UCLouvain, Belgium **2** TU Graz, Austria ¶ Corresponding author * These authors contributed equally.

Summary

Side-channel attacks exploit unintended leakage from an electronic device in order to retrieve secret data. In particular, attacks exploiting physical side-channels such as power consumption or electromagnetic radiations to recover cryptographic keys are an important threat to embedded devices. Countermeasures against these attacks have been extensively researched for more than two decades and are often deployed in security-critical devices.

A side-channel attack is made of three steps. First, the leakage is measured. Then, a statistical processing is applied to this leakage in order to infer the internal behavior of the device (typically, an intermediate state of the cryptographic algorithm). Finally, the cryptographic key is recovered from the known behavior ([Standaert et al., 2009](#)).

For the statistical processing, we distinguish between two classes of attacks, based on the use of a profiling dataset. Such a dataset consists of leakage measurements on a device running the cryptographic algorithm with the known key. Profiled attacks use this data to fit a statistical model (or train a machine-learning model) of the device, while non-profiled attacks have to rely on *a priori* models and are therefore less powerful ([Chari et al., 2002](#)).

There are two main approaches for evaluating the security of devices against side-channel attacks. First, attack-based evaluations try to attack the device and report their success or failure. In case of success, the main figure of merit is the number of traces (i.e., number of executions of a cryptographic algorithm for which the leakage is measured). Second, detection-based evaluations try to detect the presence of key-dependent leakage and sometimes quantify it. These two types of methods can be complementary in the evaluation of a device.

Side-channel evaluations are used in various research contexts, such as analyzing the effectiveness of a newly proposed countermeasure or analyzing a widely deployed device. In SCALib, we implement algorithms for commonly used metrics and methods in side-channel security evaluations, attack-based and evaluation-based. We focus on the requirements of evaluations and do not implement complete attacks when they are not needed to evaluate the security of a device.

SCALib is distributed as a Python package and uses 16-bit integer NumPy ([Harris et al., 2020](#)) arrays for leakage traces. For the sake of efficiency, most algorithms are implemented in Rust, allowing fine control of the memory accesses and enabling efficient parallelization.

Statement of need

Many of the algorithms used in side-channel security evaluations are well-known statistical techniques. For instance, the widely used TVLA methodology is based on the Welch t-test for the difference of means ([Schneider & Moradi, 2015](#)). Also, when modeling the leakage, techniques such as Linear Discriminant Analysis (LDA) ([Standaert & Archambeau, 2008](#)) can be used. While implementations of these algorithms are fairly easy to find, our use-case has a few particularities that motivate dedicated implementations. For example, the number

of traces used in an evaluation can be very large, amounting to terabytes of data, hence incremental single-pass algorithms (that avoid the need to store and/or load multiple times the dataset) are highly desirable. Moreover, while the leakage samples are acquired at a fairly low-resolution (8-bit to 16-bit integers), detection of very small effect sizes is needed, as they can potentially be exploited to mount an attack. Besides this requirement, leakage traces contain many points (typically thousands), and many metrics have to be computed for each of these points, providing parallelization opportunities. As a result of these characteristics, dedicated implementations can achieve much better accuracy and performance than generic or naive (e.g., pure NumPy) ones.

On the other hand, security-specific algorithms are also used, such as key rank estimation (which allows us to know the computational cost of the last part of a side-channel attack without actually running it) (Poussier et al., 2016).

While multiple open-source side-channel attack and evaluation libraries exist, most of them offer a very limited feature set and are unmaintained. The most comprehensive libraries are lascar (Charles Guillemet & Servant, 2023) and SCAred (Guillaume Bethouart, 2023), which offer implementations of some evaluation metrics and non-profiled attacks.

SCALib complements and improves over these libraries by providing better implementations for the computation of two common evaluation metrics, by providing algorithms for profiled side-channel attacks and including a key rank enumeration algorithm as a final evaluation step. More precisely, for leakage metrics, we implement the Welch t-test and the computation of the signal-to-noise ratio, and our implementations are significantly faster than the ones of lascar and SCAred (Cassiers, 2023). Moreover, our t-test implementation includes so-called higher-order and multivariate evaluations (Schneider & Moradi, 2015). Regarding profiled attacks, SCALib includes an implementation of LDA with a dimensionality reduction step (this provides a regularization and improves classification performance) (Standaert & Archambeau, 2008). We also implement the soft analytical side-channel attack (SASCA), which is a variant of the belief propagation algorithm (Veyrat-Charvillon et al., n.d.). Finally, our key-rank estimation implementation relies on an efficient histogram-based algorithm (Poussier et al., 2016).

SCALib has been used in many recent papers as a tool to validate new protected designs (Nagpal et al., 2022), to publish new attacks on public implementations (Bronchain et al., 2021), and also as a basis to develop new attack and evaluation methodologies (Bronchain & Standaert, 2021).

Acknowledgments

References

Bronchain, O., Cassiers, G., & Standaert, F.-X. (2021). Give me 5 minutes: Attacking ASCAD with a single side-channel trace. *IACR Cryptol. ePrint Arch.*, 817.

Bronchain, O., & Standaert, F.-X. (2021). 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), 202–234. https://doi.org/10.46586/tches.v2021.i3.202-234

Cassiers, G. (2023). SCABench: A benchmark suite for side-channel analysis libraries. In *GitHub repository*. GitHub. https://github.com/cassiersg/SCABench

Chari, S., Rao, J. R., & Rohatgi, P. (2002). Template attacks. In B. S. K. Jr., Çetin Kaya Koç, & C. Paar (Eds.), *4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002), Revised Papers* (Vol. 2523, pp. 13–28). Springer. https://doi.org/10.1007/3-540-36400-5_3

Charles Guillemet, M. S. P., & Servant, V. (2023). LASCAR: Ledger's advanced side channel analysis repository. In *GitHub repository*. GitHub. https://github.com/Ledger-Donjon/lascar

Guillaume Bethouart, R. M., Rémi Huguet. (2023). SCAred. In *GitLab repository*. GitLab. https://gitlab.com/eshard/scared

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Nagpal, R., Gigerl, B., Primas, R., & Mangard, S. (2022). Riding the waves towards generic single-cycle masking in hardware. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, *2022*(4), 693–717. https://doi.org/10.46586/tches.v2022.i4.693-717

Poussier, R., Standaert, F.-X., & Grosso, V. (2016). Simple key enumeration (and rank estimation) using histograms: An integrated approach. In B. Gierlichs & A. Y. Poschmann (Eds.), *Proceedings of 18th International Conference on Cryptographic Hardware and Embedded Systems - (CHES 2016)* (Vol. 9813, pp. 61–81). Springer. https://doi.org/10.1007/978-3-662-53140-2_4

Schneider, T., & Moradi, A. (2015). Leakage assessment methodology - A clear roadmap for side-channel evaluations. In T. Güneysu & H. Handschuh (Eds.), *Proceedings of 17th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2015)* (Vol. 9293, pp. 495–513). Springer. https://doi.org/10.1007/978-3-662-48324-4_25

Standaert, F.-X., & Archambeau, C. (2008). Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In E. Oswald & P. Rohatgi (Eds.), *Proceedings of 10th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2008)* (Vol. 5154, pp. 411–425). Springer. https://doi.org/10.1007/978-3-540-85053-3_26

Standaert, F.-X., Malkin, T., & Yung, M. (2009). A unified framework for the analysis of side-channel key recovery attacks. In A. Joux (Ed.), *Advances in Cryptology - Proceedings of 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 200:9)* (Vol. 5479, pp. 443–461). Springer. https://doi.org/10.1007/978-3-642-01001-9_26

Veyrat-Charvillon, N., Gérard, B., & Standaert, F.-X. (n.d.). Soft analytical side-channel attacks. In P. Sarkar & T. Iwata (Eds.), *Advances in Cryptology - Proceedings of 20th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2014), part i*.