IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2025, No. 3, pp. 717–744. DOI:10.46586/tches.v2025.i3.717-744

Primitive-Level vs. Implementation-Level DPA Security: a Certified Case Study

(Pleading for Standardized Leakage-Resilient Cryptography)

Charles Momin[1], François-Xavier Standaert[1] and Corentin Verhamme[1]

Crypto Group, ICTEAM Institute, UCLouvain, Louvain-la-Neuve, Belgium

Abstract. Implementation-level countermeasures like masking can be applied to any cryptographic algorithm in order to mitigate Differential Power Analysis (DPA). Leveraging re-keying with a Leakage-Resilient PRF (LR-PRF) is an alternative countermeasure that requires a change of primitive. Both options rely on different security mechanisms: signal-to-noise ratio amplification for masking, signal reduction for LR-PRFs. This makes their general comparison difficult and suggests the investigation of relevant case studies to identify when to use one or the other as an interesting research direction. In this paper, we provide such a case study and compare the security that can be obtained by using an unprotected hardware coprocessor, to be integrated into a leakage-resilient PRF, and a certified one, protected with implementation-level countermeasures. Both are available on "commercial off-the-shelf" devices and could be used for lightweight IoT applications. We first perform an in-depth analysis of these targets. It allows us to put forward the different evaluation challenges that they raise, and the similar to slightly better cost vs. security tradeoff that the leakage-resilient PRF offers in our experiments. We then discuss the advantages and limitations of both types of countermeasures. While there are contexts where the higher flexibility of masking is needed, we conclude that there are also applications that would strongly benefit from the simplicity of the LR-PRF's design and evaluation. Positing that the lack of standards is the main impediment to their more widespread deployment, we therefore hope that our results can motivate such standardization efforts.

Keywords: Side-Channel Analysis · Countermeasures · Masking · Re-keying

1 Introduction

Differential Power Analysis (DPA) is a very powerful type of side-channel attack [KJJ99]. As for example reflected by the linear bound in [dCGRP19], it allows reducing the security of a block cipher key exponentially in the number of plaintexts for which the adversary can combine her physical measurements. There exists two main approaches in order to mitigate such differential attacks. One is to reduce the leakage at the implementation level. The most popular solution for this purpose is masking (aka secret sharing) [CJRR99, GP99]. Under some (noise and independence) assumptions, it allows reducing the information leaked by an implementation exponentially in the number of shares used by the countermeasure, which in turn leads to an exponential security amplification [ISW03, PR13, DFS15, IUH22, BCG+23]. Another approach is to design primitives that limit the exponential security decrease that DPA causes, by limiting the number of plaintexts for which the adversary can combine her measurements, thanks to re-keying. One popular solution for this purpose is to use a leakage-resilient PRF [DP10, SPY+10, FPS12, YS13, BSH+14].

It is already known that masking and leakage-resilient PRFs may not blend gracefully [BGS15, UBS21]. The reason for this poor combination is intuitive and informative.

Received: 2025-01-15 Accepted: 2025-03-15 Published: 2025-06-05

On the one hand, Boolean masking, which is prominently used in order to protect ciphers operating in binary fields like the AES [DR20] or Ascon [DEMS21], essentially increases security against DPA via noise amplification. That is, it aims to force the adversary to estimate higher-order statistical moments of the leakage distribution, a task of which the complexity grows exponentially in the number of shares if the measurements are sufficiently noisy. On the other hand, leakage-resilient PRFs limit the number of plaintexts for which the adversary can observe the leakage, but they do not limit the number of times each plaintext can be observed. That is, they aim to limit the amount of side-channel signal that remains (for example) after averaging the measurement noise, thanks to repetition. Since Boolean masking amplifies the noise and leakage-resilient PRFs allow repetition to get rid of the noise, it implies that the complexity to attack their combination adds up, rather than multiplies, as expected for a graceful combination of countermeasures [RPD09].

Given that combining Boolean masking and leakage-resilient PRFs is not desirable, the important next question becomes: *which one to privilege in which context?* Quite naturally, the fact that primitive-level and implementation-level countermeasures leverage very different security mechanisms makes their general comparison difficult and, to the best of our knowledge, there are no works in this direction so far. In this paper, we therefore relax this problem by comparing these two types of countermeasures in a specific yet practically-relevant context. Namely, we consider the security that can be obtained by using commercial MCUs as could be exploited in lightweight (e.g., IoT) applications. We focus in particular on the security that can be obtained by leveraging the AES hardware implementations which are popular coprocessors in the embedded security market.

Unsurprisingly, AES unprotected coprocessors remain the most widespread and the landscape of MCUs protected with implementation-level countermeasures is quite scarce, especially when focusing on devices that are accessible for academic research. One of the few (if not the only) option for this purpose is the STM32U5 family of chips, which is based on the ARM Cortex M33 architecture. Of particular interest for our investigations, these devices embed an AES coprocessor protected against physical attacks which is PSA-certified Level-3.[1] As a natural competitor, we integrate the unprotected AES coprocessor of the (ARM Cortex M4 based) STM32F4 family of chips into a leakage-resilient PRF.

Due to the closed-source nature of these implementations, our study is performed without an accurate description of the targets' hardware architecture nor, for the STM32U5 device, of the countermeasures it implements. As a result, we start our investigations with some side-channel reverse engineering. For the unprotected AES implementation of the STM32F4, Signal-to-Noise Ratio (SNR) analyzes confirm that it corresponds to a 128-bit loop architecture (matching the specification that the AES is performed in 11 cycles). For the protected AES implementation of the STM32U5, a similar analysis rather suggests a 32-bit architecture, with noise/jitter engines (mentioned in the specifications), first-order masking for the rounds and no masking for the key scheduling algorithm.[2]

Based on these preliminary efforts, we then try to compare the STM32U5 AES protected with implementation-level countermeasures, which we from now assume to leverage Boolean masking for simplicity, and the STM32F4 AES integrated in a leakage-resilient PRF, on sound empirical bases. For this purpose, we perform an in-depth investigation of state-of-the-art profiled attacks, trying to optimize their main parameters. Due to the very different nature of the challenge to evaluate the STM32U5 and STM32F4 implementations, we borrow different strategies for both targets. Namely, we follow a (more) certification-style/qualitative approach for the protected AES, where we use one month of measurements in order to exhibit a first baseline attack, and we follow a (more) worst-case/quantitative approach, trying to bound the best attacks, for the unprotected AES integrated in a

[1] https://newsroom.st.com/media-center/press-item.html/n4377.html.
[2] We also observed redundancy, presumably for fault detection/correction purposes.

leakage-resilient PRF [Sta24]. In both cases, we discuss the risk of security overstatements. For the STM32F4, we additionally discuss the possible gap between our close-to-worst-case evaluations and more practical ones, as advertized by the backwards security evaluations put forward in [ABB+20]. By combining these "best-effort" estimations, we observe that the cycle count vs. security tradeoff of the leakage-resilient PRF using an unprotected coprocessor is similar to slightly better than the one of the protected AES core of the STM32U5 chip. Given the more worst-case analysis of the leakage-resilient PRF, we deem this conclusion more likely to be amplified than weakened with more evaluation efforts.

We conclude the paper by discussing the generalization and the impact of our findings. From the generalization viewpoint, we highlight that our comparisons are highly technology-dependent and therefore hard to capture with theoretical models. From the impact viewpoint, we nevertheless argue that considering the standardization of leakage-resilient PRFs able to leverage the (e.g., AES) coprocessors that are readily available on a vast range of products could be highly desirable: not because of a better cost vs. security tradeoff but because of the easier risk assessment (i.e., security evaluation) that they enable, their more direct link between functional correctness and physical security and, mostly, their currently wider availability than protected implementations. This makes them a solution of choice for integration in low-cost applications where secure coprocessors and/or the expertise required to implement masking securely may be too expensive.

Responsible disclosure and open designs. Our analyzes put forward a first-order flaw in the (presumably masked) AES coprocessor of the STM32U5 chip. This observation has been communicated to the STMicroelectronics Product Security Incident Response Team before submission, who will provide guidance to end users so that this flaw cannot be exploited in practical applications. We further note that this flaw is only detectable with a large number of measurements (in the tenths of millions range) and may not contradict the PSA certificate claims. It could be tempting, based on these results, to conclude that preventing cryptographic implementations to be scrutinized by the academic community could be beneficial to security. We believe the exact opposite conclusion should prevail. Namely, these results mostly question (once again) the limited quantitative value of current certification schemes, and the importance of a continuous and open approach to gain confidence in implementation security.[3,4,5] Positively, enabling the public investigation of cryptographic coprocessors can only help identifying tracks for improving them in the long-term. Such public investigations would become even more effective if they could leverage more open designs (avoiding the loss of time devoted to reverse engineering). Negatively, a time-limited and closed-source evaluation process inherently comes with an increased risk of overstated security claims, potentially leaving (sometimes important) attack vectors hidden to end users, despite exploitable by determined adversaries.

Related works. Our focus is on protecting the execution of basic primitives (e.g., block ciphers or PRFs) against leakage, which is orthogonal to the challenge of minimizing the amount of executions which require DPA protection in a mode of operation, thanks to leveled implementations [BBC+20]. Primitive-level countermeasures could also be designed based on permutations, as for example exhibited with ISAP [DEM+20]. The reason of our focus on the AES is the better availability of its coprocessors in lightweight embedded devices. Abdalla et al. show how to use re-keying in order to design a leakage-resilient encryption scheme [ABF13]. Ueno et al. show how to further limit the leakage of temporal keys by using multiple re-keying [UHIM24], which was not needed in our work given that we already reach security with a single re-keying. Finally, (fresh) re-keying can also rely on key-homomorphic primitives, which both improve the simplicity to implement and the

[3] https://ninjalab.io/ledger-challenge-2018/
[4] https://ninjalab.io/a-side-journey-to-titan/
[5] https://ninjalab.io/eucleak/

performances of masking [MSGR10, DKM+15, DFH+16, DMMS21, HMM+23]. It can therefore be viewed as an intermediate between leakage-resilient PRFs and the masked implementation of symmetric primitives like block ciphers or permutations.

Cautionary note. As mentioned above, a general comparison of masking and leakage-resilient PRFs is difficult because they exploit different security mechanisms. It implies that there is no single technology that would be most suitable to implement both types of countermeasures. We illustrate this claim by analyzing the LR-PRF based on the unprotected implementation that is also available on the STM32U5 chip. As reported in Appendix A, such an unprotected implementation has slightly more physical noise in its leakages, presumably due to a more recent technology, but it relies on a 32-bit architecture. Such a higher physical noise level is typically beneficial for masking (which aims at noise amplification). By contrast, it is less beneficial when combined with the LR-PRF for which the noise can be mitigated thanks to repeated observations and the main security parameter is the size of the target architecture. This can be explained by the fact that the security of LR-PRFs is usually established in the bounded leakage model [DP10, SPY+10, FPS12, YS13]. Such a model requires limiting the side-channel signal, which itself decreases with the size of the target implementations, as for example shown in [CDSU23] and confirmed by our experiments in appendix. So overall, masking benefits from noise engines much more than LR-PRFs which only benefit from signal reduction techniques. Whether this implies a better security vs. performance tradeoff therefore depends on the efficiency of such noise engines, which is case-specific and hard to capture theoretically. As a result, the main viewpoint of the paper is the one of an end-user willing to implement symmetric authentication or encryption (or authenticated encryption) with side-channel security guarantees, based on a best-effort selection of COTS devices. Given this restricted goal, we argue that our selection of COTS devices is a reasonable starting point since alternatives to the STM32F5 (protected coprocessor) are scarce and the 128-bit architecture of the STM32F4 (unprotected coprocessor) is the most parallel we could find. Quite naturally, our results can also provide COTS devices' manufacturers with new incentives to design implementations suitable for leakage-resilience.

Terminology. The leakage-resilient PRF is usually denoted as a mode-level countermeasure in the literature. We denote it as a primitive-level countermeasure since such a PRF is only a building block that still has to be integrated in a mode of operation for authentication, encryption or authenticated encryption. Most of our observations are nevertheless valid for re-keying, whether used in a primitive or a mode of operation.

2 Background

2.1 Leakage-resilient PRF

The tree-based PRF we consider in this paper was initially proposed by Goldreich, Goldwasser and Micali [GGM84]. It has then been shown in a sequence of works that it has good properties for improving security against leakage [DP10, SPY+10, FPS12, YS13, BSH+14]. As illustrated in Figure 1, the function processes a 128-bit input x with a 128-bit key k in order to create a 128-bit output y, by sequentially applying $128/n_x$ AES-based stages.

In each of those stages, n_x bits of the input x are used to select an AES plaintext among $N_x^s = 2^{n_x}$ fixed (public) ones. At the i-th stage, the selected plaintext is AES-encrypted by the stage key k^i (with $k^0 = k$), which corresponds to the output of the previous stage. Therefore, a complete LR-PRF execution requires $128/n_x$ AES executions.

Such a PRF is interesting from the leakage viewpoint since it bounds the amount of AES plaintexts that an adversary can observe in order to perform a DPA against each stage key to N_x^s. By contrast, it does not bound the number of times each of these N_x^s plaintexts can be observed. We will next use the notation N_r (where r stands for repetition) to denote

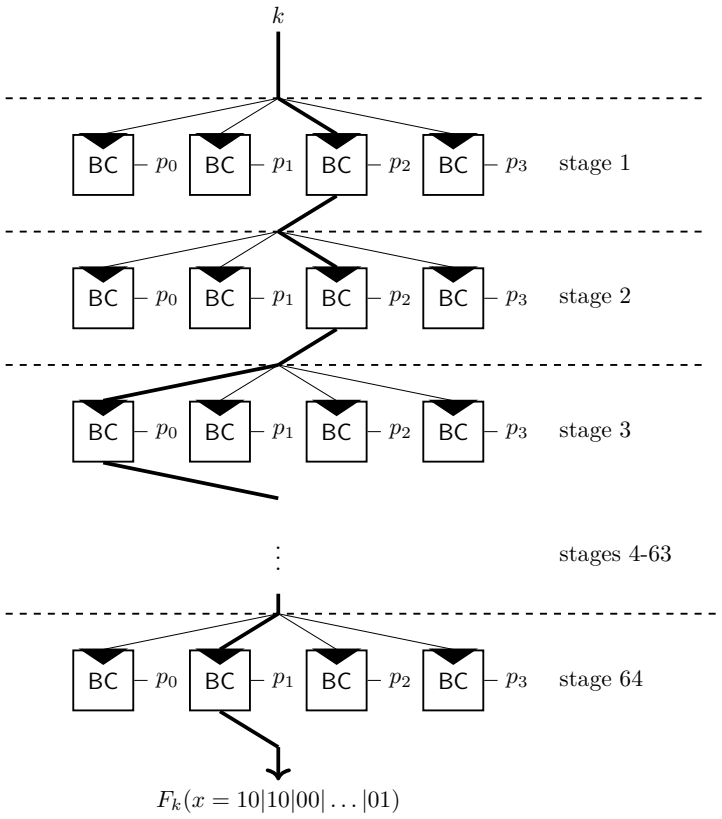


Figure 1: Leakage-resilient PRF, $N_x^s = 4$ ($n_x = 2$).

the number of times each AES plaintext is observed by the adversary. The expectation is that such a LR-PRF can be secure against DPA without expensive implementation-level countermeasures, for example by leveraging parallel hardware implementations which are assumed to be difficult to break with a few plaintexts, even after many repetitions.

Remark. Variants of the LR-PRF exist. Medwed, Standaert and Joux proposed to choose the $N_x^s \leq 256$ plaintexts so that all bytes take the same values, in order to generate "key-dependent algorithmic noise" [MSJ12]. Medwed et al. also considered the blinding of the LR-PRF plaintexts to improve security [MSNF16]. While these variants can indeed deliver higher security, these improvements come at the cost of stronger and harder to evaluate leakage assumptions, which we therefore did not consider for simplicity (which we believe to be an important asset in the context of our standardization plea).

2.2 Evaluation tools and metrics

Additive noise assumption. A common setting for side-channel security evaluations consists in considering that an adversary is able to measure the physical behavior (e.g., power consumption or electromagnetic emanation) of a chip during its executions. In this context, the measurements acquired during each execution (i.e., the traces) take the form of a vector of so-called time samples, representing the instantaneous current consumption of the device. These vectors are typically acquired using a digital oscilloscope sampling several samples per clock cycle of the device being measured. The measurements are generally subject to some variations known as the noise, and a standard assumption is to consider each time sample as a combination of a deterministic part depending on the

current internal state of the device and a (random) additive noise. Under this assumption, the leakage sample l_t of an intermediate state z can be expressed as:

$$l_t(z) = \delta(z) + r_t,$$

where δ is the deterministic part of the leakage and r is the noise. The dependency between the leakage of the device and the internal states it manipulates can then be exploited in order to retrieve information about the intermediate variables being manipulated.

Attack framework. A standard strategy to extract information from a leaking device is to rely on profiled attacks, initially introduced under the name "template attacks" [CRR02]. They work in two phases: first, an offline profiling phase during which the evaluator builds a probabilistic model of the target device's leakages. Second, an online attack phase where information is extracted by inferring the internal state manipulated by the target device, thanks to this probabilistic model. Profiled attacks therefore rely on two main quantities: the profiling complexity, which corresponds to the amount of data (and time) used during the profiling phase, and the attack complexity, defined as the amount of traces (and time) required to successfully mount an attack during the online attack phase.

Signal-to-Noise Ratio. Building multidimensional statistical models can be expensive (in terms of profiling complexity). As a result, a classical preliminary step when performing profiled attacks consists in identifying relevant time samples in the traces, usually denoted as the Point-Of-Interest (POIs). A common tool for that purpose is the Signal-to-Noise Ratio (SNR) [Man04]. It defines the (univariate) side-channel signal as the variance of the mean leakages (computed for the different values of the intermediate state z), leading to the following definition of SNR:

$$\mathsf{SNR}_t(Z) = \frac{\mathsf{Var}_{z \in \mathcal{Z}}\left(\mathsf{E}\left[\mathbf{l}_t^z\right]\right)}{\mathsf{E}_{z \in \mathcal{Z}}\left[\mathsf{Var}\left(\mathbf{l}_t^z\right)\right]},$$

where $\mathsf{Var}(\cdot)$ and $\mathsf{E}[\cdot]$ respectively denote the variance and the expected value, and $\mathbf{l}_t^z$ is the vector containing all the leakage values taken by the t-th time sample under the hypothesis that $Z = z$, with $\mathcal{Z}$ being the domain of z. Under the additive noise assumption, the SNR can be rewritten as:

$$\mathsf{SNR}_t(Z) = \frac{\mathsf{Var}\left(\delta_t(Z)\right)}{\mathsf{Var}\left(R_t\right)}.$$

Linear subspace template attack. In their original form, template attacks model the true (unknown) PDF of the leakage with a multivariate Gaussian distribution [CRR02]. The goal of the profiling phase is thus to evaluate $\hat{\mathsf{f}}(\mathbf{l}|z)$, the joint distribution of N_s POIs conditioned on the targeted state value z. When the number of informative POIs in the leakage traces is large, it is convenient to additionally leverage dimensionality reduction techniques, for example by estimating the templates in a linear subspace as follows:

$$\hat{\mathsf{f}}(\mathbf{l}|z) = \frac{1}{\sqrt{(2\pi)^{N_d}\left|\hat{\mathbf{\Sigma}}_z\right|}} \exp\left(-\frac{1}{2}(\mathbf{W}\mathbf{l} - \hat{\boldsymbol{\mu}}_z)^\top \hat{\mathbf{\Sigma}}_z^{-1}(\mathbf{W}\mathbf{l} - \hat{\boldsymbol{\mu}}_z)\right), \tag{1}$$

where $\mathbf{l}$ is a leakage vector composed of N_s POIs, $\mathbf{W}$ is the linear projection matrix, $\hat{\boldsymbol{\mu}}_z$ is the estimated mean vector and $\hat{\mathbf{\Sigma}}$ the estimated covariance matrix. In the following, we will use Linear Discriminant Analysis (LDA) in order to estimate the $\mathbf{W}$ matrix, which optimizes the SNR of the (N_d) dimensions kept after projection [SA08].

Note that we use LDA as an interpretable starting point that enables efficient profiling. More advanced statistical methods could be considered to further improve our evaluations [PPM+23], which we leave as an interesting scope for further research.

IT metrics and model quality assessment. Information theoretic metrics like the Mutual Information (MI) have been introduced in [SMY09] as a natural generalization of the (univariate) SNR to multivariate leakages. The MI theoretically quantifies the amount of information related to an intermediate variable present in any leakage function. However, evaluating it directly requires the knowledge of the true (unknown) distribution of the leakage function and estimating it for a large number of POIs can be computationally hard [CRBO24]. An efficiently computable alternative is to evaluate the Perceived Information (PI) [RSV+11], which is a lower bound of the MI quantifying the amount of information that can be extracted from the leakage by a previously trained statistical model [BHM+19]. Considering a profiling traces set $\mathcal{L}_p$ used to estimate a model and an independent test traces set $\mathcal{L}_t$, the PI can be easily estimated as:

$$\hat{\mathsf{PI}}(Z;\mathbf{L}) \approx \mathsf{H}(Z) + \sum_{z\in\mathcal{Z}} \Pr(z) \sum_{\mathbf{l}\in\mathcal{L}_t(z)} \frac{1}{|\mathcal{L}_t(z)|} \log_2 \hat{\Pr}(z|\mathbf{l}),$$

where $\hat{\Pr}(z|\mathbf{l})$ is obtained by applying Bayes using the profiled model, and the value $\mathcal{L}_t(z)$ denotes the subset of $\mathcal{L}_t$ containing the traces under the condition $Z = z$. Note that this estimation can be used for both univariate and multivariate leakage PDFs.

As a complement to the PI, the Training Information (TI) has been shown to be a upper bound of the PI [MCHS23]. The TI is estimated in the same way as the PI, but using the profiling set instead of the test set (i.e., by evaluating the information in an overfitted manner). Since the PI increases with the profiling complexity and the TI decreases with the profiling complexity, they are convenient metrics to determine the extent to which model improvements remain possible by increasing the profiling complexity.

Security evaluation and key ranking. The most straightforward way to evaluate the security of an implementation is to assess the feasibility of (ideally worst-case) attacks by performing them in practice. A standard metric for the attack success is the logarithm of the key rank, which can be computed using rank estimation algorithms. We next use the histogram-based method from [PSG16] to this end. Ideally, the (log) key rank must be computed for a few independent attacks, to estimate its median and quantiles.

3 Reverse engineering

In this section, we first detail the targets that we analyze and describe the setup we developed to measure their current consumption. Next, we discuss the side-channel reverse engineering steps that we carried out in order to gain preliminary understanding about the underlying architectures of the STM32F4 and STM32U5 AES coprocessors.

3.1 Targets and setup description

Our first target is the STM32F415: a microcontroller from STMicroelectronics' STM32F4 series, which is well-known for delivering high performance along with a wide range of integrated features. The STM32F415 is built around an ARM Cortex-M4 core, combining advanced processing capabilities with Digital Signal Processing (DSP) features and a Floating Point Unit (FPU). In addition to its computational performance, the STM32F415 MCU stands out with its fully parallel (unprotected) AES cryptographic accelerator.

Our second target is the STM32U585: a microcontroller from STMicroelectronics' STM32U5 series, which is designed with a strong emphasis on ultra-low power consumption. Built around the ARM Cortex-M33 core, the STM32U585 benefits from the integrated TrustZone technology for hardware-based security. One of the key features of the STM32U585 is its combination of security tools, such as a True Random Number

Generator (TRNG) and a secure AES encryption hardware engine, enhancing its suitability for secure and connected applications. In order to perform a secure AES encryption with the protected hardware accelerator, one must ensure that the TNRG is feeding fresh randomness (which we did) and select a way to load the key. Among the possibilities to load the key in the coprocessor, we chose to use the software loading mechanism, since it ensured a better acquisition throughput than the others. We got confirmation through STM32's forum that the SCA countermeasure(s) remain(s) active in that scenario.

Both MCUs are packaged in an LQFP-64 format.

In terms of measurement capabilities, we replicate our setup for both targets in order to process them in parallel, ensuring similar conditions and measurement capacities for each. The targets are soldered on a CW308T-STM32F: a circuit board supporting several STM32F devices in the LQFP-64 package. These boards are then plugged in a CW308 UFO board that allowed us to drive the MCU clock with an 8MHz external crystal and to supply power through a potentiometer, which we set at 1.8V in our experiments.

We run the unprotected AES core at 8MHz to obtain a low operating clock as indicated in the guidelines from [BUS21]. In the case of the protected AES core, we have to operate at 48MHz through a secure clock. The signal is collected with a CT1 current probe on the CW308 UFO shunt jumpers and sampled with a PicoScope 6424E, operating at 5[GSamples/s] with a 500 MHz analog bandwidth, using 10-bits of vertical resolution.

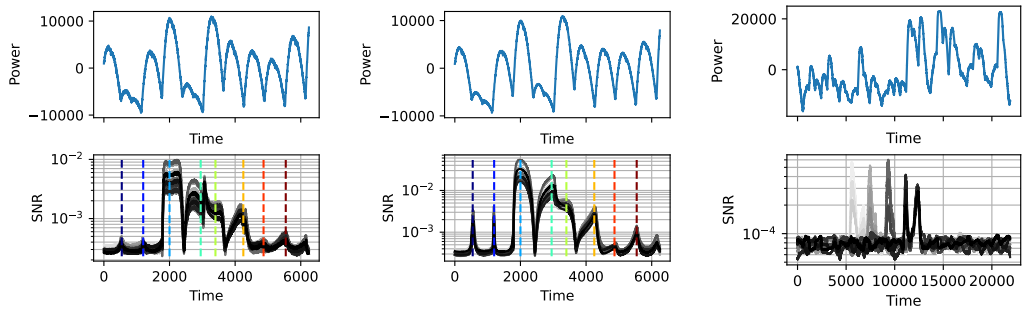
For each MCU, we finally apply the methodology of [SM16] to speed up measurements. Namely, we use the AES coprocessor as a Pseudo-Random Number Generator (PRNG) to generate plaintexts and keys, thereby significantly reducing the communication overheads with a desktop system. This involves encrypting a zero vector in CBC mode under a chosen initialization vector (IV). We compute the SNRs in real time to quickly adjust acquisition parameters, such as signal windowing and acquisition range. The configuration that yields the highest SNR estimation is selected. SNR calculations, the LDA modeling and the rank estimation are performed using the open-source SCALib library [CB23].

3.2 Architectural assumptions

The main information that specifications provide about the target's respective architectures is a cycle count of 11 for the STM32F4 and 528 for the STMU5 (suggesting a more parallel architecture for the former one). As a preliminary before evaluating the physical security of these targets, we therefore performed some side-channel reverse engineering in order to infer/confirm architectural assumptions. We use SNR estimations for the 16 S-boxes of the first AES round for this purpose. They are illustrated in Figure 2.

Starting with the unprotected implementation, we can confirm a loop architecture processing a 128-bit round per cycle from the (left and middle) plots with the traces on top, where the 16 SNR peaks overlap, suggesting a parallel execution of the S-boxes.

Following with the protected core, we observe noisier traces reflected by lower SNR values (suggesting active countermeasures). Yet, we can still distinguish five distinct spikes. The first four of them correspond to each AES column being manipulated, hinting towards a 32-bit serial implementation for the SubBytes operation that we target in our security evaluations, while all the 16 bytes are springing up for the last peak, hinting towards a 128-bit implementation for the MixColumns operation. Interestingly, the (right) plot of Figure 2 considers a first-order signal estimation. This may either correspond to hardware-level countermeasures or to a 2-share masking scheme with a small first-order flaw due to glitches [MPG05, MPO05], transitions [CGP+12, BGG+14] or couplings [CBG+17, CEM18]. We posit that we more likely face the second option (i.e., that the implementation is Boolean masked) because, as discussed in Appendix B, we also observe higher SNR values for the key scheduling, which could suggest that this (less sensitive) part of the



(a) STM32F415, without repetition ($N_r = 1$) for one million observations ($N_p = 10^6$).

(b) STM32F415, with repetitions ($N_r = 1024$) for one million observations ($N_p = 10^6$).

(c) STM32U585, with $N_r = 1024$ repetitions for 3 million observations ($N_p = 3 \times 10^6$).

Figure 2: Comparison of the SNR for both targets and impact of the repetition factor N_r. The 16 SNR curves with different shades of gray correspond to different S-boxes.

implementation is not masked. In both cases, these assumptions admittedly remain educated guesses since no information on the HDL is specified in the datasheets.

Remark. More advanced Side-Channel Analysis Reverse Engineering (SCARE) techniques could be used to gain a finer-grain understanding of our targets [DLMV05, RR13], for example regarding the type of countermeasures implemented in the STM32U5. We leave them as scopes for further investigations and next detail how we estimate the physical security of these targets based on the aforementioned architectural assumptions.

4 Evaluation rationale

As hinted in the introduction, the security of a leakage-resilient PRF and the one of a masked block cipher rely on different physical assumptions, which are formalized by different leakage models. LR-PRFs are proven in the bounded leakage (or related) model(s) [DP10, SPY+10, FPS12, YS13]. Concretely, bounded leakage requires minimizing the side-channel signal, which typically benefits from parallelism [BSH+14]. Boolean masking is proven in the noisy leakage model [PR13], under some noise and independence assumptions. The independence assumption is usually tested by assessing a "statistical security order", which ensures that the statistical moments of the leakage distribution are independent of the key up to some order [SM16]. The noise condition can be quantified thanks the the SNR (in the univariate case) or the PI (in the multivariate case) [DFS15, IUH22, BCG+23].

More generally, and in both cases, the quantitative evaluation of cryptographic implementations against side-channel attacks is a challenging problem. Informally, this is because it faces a tradeoff between the facts that (*i*) as with any cryptanalytic effort, the confidence in an evaluation only grows over time and (*ii*) evaluators are constrained in time. This difficulty is further amplified when evaluators face a closed-source target, which prevents a direct identification of the best attack vectors and requires relying on unconfirmed assumptions. So compared with the situation of cryptographic algorithms, there are analogies (the best evaluations are open and continuous) and differences (there are many implementations per algorithm, so the time that can be devoted to each implementation is reduced). This situation motivates so-callsed "shortcut" approaches to physical security evaluations, leveraging the scaling trends of side-channel analysis metrics backed up by theoretical investigations, in order to bound and extrapolate the adversaries' success [Sta24].

The two AES implementations we consider in this work are typically reflecting these

different assumptions and evaluation tradeoffs. One corresponds to an unprotected but parallel AES core, aimed to ensure low side-channel signal. Despite it is a closed-source target, inferring its architecture based on specifications and leakage analyzes can be done with quite good confidence and a limited number of traces. The other is a protected AES core, for which we can only guess the countermeasures implemented with limited confidence. As a consequence, profiling an accurate leakage model for this second device is expected to be more difficult. Not only because of a lower SNR, but also because this (non worst-case) profiling cannot leverage the knowledge of the randomness used by the (presumably, Boolean masking) countermeasure of the STM32U5 chip [BDMS22]. Overall, this means that the amount of data and time needed to reach a similar confidence in the security evaluations of the two targets may vastly differ. Besides, the theoretical extrapolation of the results is also expected to be significantly harder for the protected AES core (due to a less precise understanding of its internal functioning).

These differences motivate different approaches for the evaluation of our two target implementations. For the protected implementation, we decided to follow a more certification-style approach. Namely, and after a few weeks of preliminary investigations (specifications reading, setup preparation, reverse engineering), we measured the target for one full month and spent another two months to identify a first baseline attack. For the unprotected implementation (to be integrated in a leakage-resilient PRF) we rather followed the backwards security evaluation approach proposed in [ABB+20]. For this purpose, we first measured the target for one full week, then more exhaustively investigated strong profiling attacks in a (closer-to-worst-case) evaluation setting and tried to bound the information that could be further extracted with more profiling efforts. In both cases, we additionally discuss the risks of security overstatements, and argue that they are lower for the (easier-to-evaluate) leakage-resilient PRF. We finally complete the backwards security evaluation of the STM32F4 implementation by examining the possible gap between our worst-case evaluations and more practical ones, in particular exhibiting the challenges that attacks with strong averaging raise in terms of robust profiling [WO15, CK18].

Remark. Easier-to-evaluate does not mean easy-to-evaluate. The argumentation above (and the following risk analysis sections) aim to justify that we see more room for improving our baseline attack against the protected AES implementation than for improving the (more optimized) profiled attack against the unprotected AES implementation. Yet, our main message is that side-channel security evaluations are a continuous process which benefits from open designs. Therefore investigating possible improvements is a worthwhile goal in both cases: improving the measurement setups [BUS21] or, as already mentioned, using advanced statistical methods [PPM+23], are natural directions for this purpose.

5 Protected AES Core (STM32U5)

In this section, we detail the methodology followed for evaluating the physical security of the STM32U5 target, together with experimental results. We begin with the profiling phase and explain how we optimized our LDA models in Subsection 5.1. Next, we present the performances achieved by our models, and demonstrate that they lead to successful attack against an independent validation set in Subsection 5.2. Finally, we discuss the limitations of this analysis and its potential improvements in Subsection 5.3.

5.1 Model profiling

The protected core is based on undisclosed countermeasures possibly including masking. Yet, and as already pointed out in Figure 2, a first-order leakage is present in the traces.

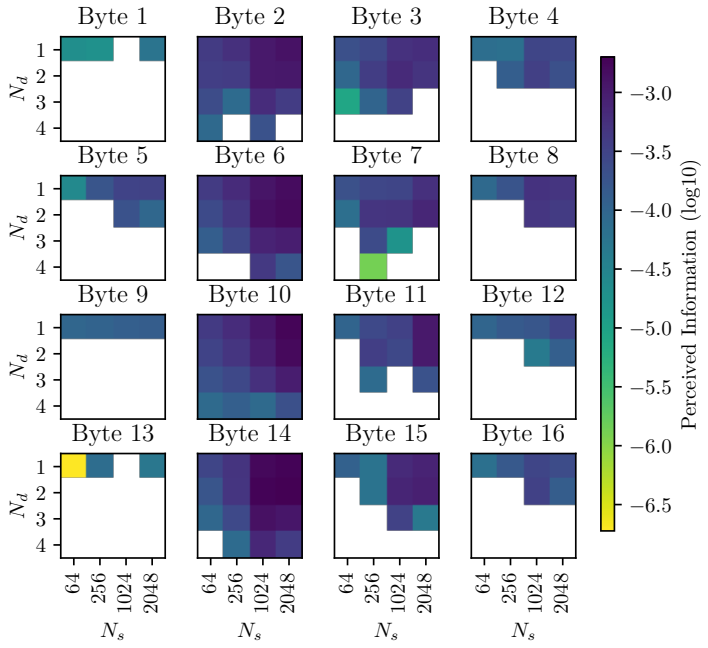


Figure 3: PI of the STM32U5 target for all the bytes and the whole parameter space with $N_r = 1024$, displayed in log scale. As X-axis we display the number of selected Points-of-Interest (N_s) and as Y-axis the number of output dimensions (N_d) for the LDA. The PI is computed for three million traces corresponding to different plaintexts/keys.

We therefore decided to estimate Gaussian templates in a linear subspace optimized thanks to LDA, for the informative time samples (located around the 5 peaks of the figure).

As usually observed with the estimation of multidimensional models, the choice of good parameters is particularly important. The standard way to find them is to explore the parameters' space in a somewhat exhaustive manner, to evaluate the resulting models, and to retain the best ones. In our case study, we evaluate the quality of a model based on its PI (i.e., the amount of information that can be extracted with it) computed thanks to k-fold cross-validation, so that the traces used for profiling a model and the ones used for testing this model are always independent (to prevent overfitting). The information values obtained for all the models constructed with the considered parameter space are summarized in Figure 3, where the white boxes denote negative PI values (i.e., models that are not informative). This figure leads to the following main observations.

First, and most importantly, it highlights that we are able to estimate informative models for all the key bytes, confirming the presence of exploitable leakage hinted at by the previous SNR-based investigations. Such models should therefore enable successful attacks, which we will confirm in the next section. Second, it is noticeable that the best models are in quite low-dimensional subspaces (i.e., $N_d \in [1, 2]$) and use a lot of POIs (i.e., large N_s values). This may suggest that the month of measurements used for profiling is not sufficient to fully characterize the information leakage of the target, which we will discuss in Subsection 6.3. Third, Figure 3 shows some column-based patterns (e.g., a more informative second column), which is consistent with the previous SNR plots, potentially suggesting some architectural feature making these bytes (slightly) more leaky.

Remark. The PI plots of Figure 3 are for average traces with $N_r = 1024$. The motivation of this averaging is only to reduce the time and memory complexity of our (already expensive)

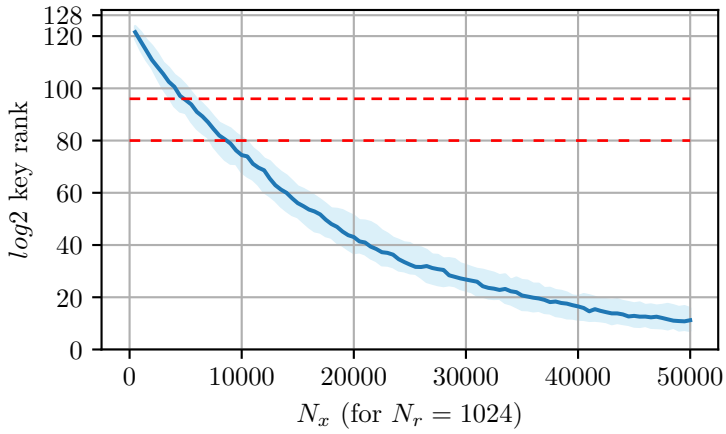


Figure 4: Median and quartiles of the key ranks for 100 independent attacks on the STM32U5 target (attack dataset computed after the interleaved training/test sets). Red horizontal lines correspond to target security levels / key ranks of 2^{80} and 2^{96}.

attacks. However, it may be that averaging is not an optimal strategy to exploit the leakage of the STM32U5 target (e.g., if there is an informative second-order leakage).

5.2 Security evaluation

We confirmed the previous observations by running concrete attacks using the models with the highest PI for all the key bytes, identified thanks to Figure 3. To this end, we captured an attack set corresponding to random plaintexts and a fixed key, with a repetition factor of 1024. In order to emulate more realistic attack conditions, the collection of the attack set was not interleaved with the one of the training and test sets. As illustrated in Figure 4, we can nevertheless perform successful key recoveries in this context. Namely, we observe that the median rank drops below 80 bits of side-channel security after 10,000 average traces. This number is further reduced to 5000 traces if we target 96 bits of security.

5.3 Risk assessment

Given the time constraints of evaluators, an important part of any security evaluation is to assess the risk of security overstatements. In the present case, this risk is admittedly high as we only demonstrate a first baseline attack, and the identification of better attacks is delayed by the closed-source target and non-worst-case profiling conditions.

More precisely, one generic source of risk naturally relates to the measurement setups. Improving the setups may lead to higher SNRs, more informative leakages and better attacks. A more specific source of risk relates to the statistical processing of the measurements, which leads to two main questions. First, can we improve the models of Figure 3 with more profiling efforts (in data and time)? In order to answer this question, Figure 5 illustrates the convergence of the PI and the TI bound of our LDA models, for two exemplary bytes. Clearly, these models have not converged yet (especially the byte in the left plot) and more profiling should therefore lead to better attacks. Second, can more advanced statistical tools such as surveyed in [PPM+23] lead to stronger attacks? While a formal/definitive answer to this question is out of reach, the fact that the LDA models we leverage only exploit first-order signal for a presumably masked implementation at least leaves room for advanced models that may or may not improve the attack results.

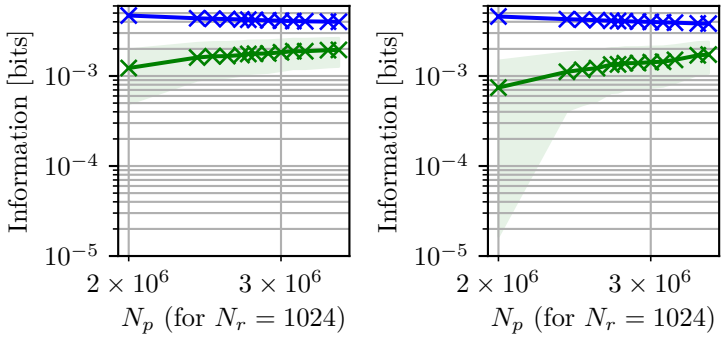


Figure 5: PI (green) and TI (blue) for the most informative bytes on the STM32U5.

6 Leakage-resilient PRF (STM32F4)

In this section, we detail the methodology followed for evaluating the physical security of the STM32F4 target integrated in the leakage-resilient PRF of Figure 1. We begin with the profiling phase and explain how we optimized our LDA models in Subsection 6.1. Next, we present the performances achieved by our models, and assess the security of the LR-PRF in Subsection 6.2. Finally, we discuss the limitations of this analysis and its potential improvements in Subsection 6.3. We also put our results in the perspective of a backwards security evaluation by discussing model robustness issues in Subsection 6.4.

Before describing these results, we recall that the LR-PRF bounds the number of plaintexts N_x^s that can be observed by the side-channel adversary, for each of its stage keys. This bound is the most important parameter to select since it directly determines the cost vs. security tradeoff of the construction. In order to estimate it, the main difficulty is that an adversary against the LR-PRF can actually repeat the observation of each plaintext N_r times and, for example, average the leakage traces in order to make them less noisy (or even noise-free). So a critical step of our evaluations is to gain confidence that the impact of increasing the number of repetitions N_r at some point "saturates", indicating that most of the (bounded) side-channel signal has indeed been captured by the evaluation.

6.1 Model profiling

As for the STM32U5, the POIs' selection is based on the SNR results given in Figure 2 from Section 3. We observed significant SNR levels for each (first-round) key byte, spanning over 2500 time samples, when no repetitions are considered (i.e., $N_r = 1$). This situation does not change much when a high amount of repetitions is considered (e.g., $N_r = 1024$), apart from higher SNR values. So, again, we decided to estimate Gaussian templates in a linear subspace optimized thanks to LDA, for the informative POIs.

As in the previous section, we conduct an exhaustive exploration of the parameter space considered, in order to identify the parameters achieving a model that maximizes the PI for each byte. Nevertheless, this time we do it in function of the N_r parameter. The results obtained are summarized in Figure 6 and Figure 7 for exemplary values $N_r = 1$ and $N_r = 1024$. In both cases, we observe that all models lead to positive PI values that are higher than for the protected coprocessor in the previous section. Given the (presumably) more parallel nature of the unprotected coprocessor, this again confirms that implementation-level countermeasures were active for the STM32U5 target.

The most interesting observation of these figures appears when comparing the impact of an increased averaging. Namely, a notable effect of moving from Figure 6 to Figure 7 is that

the increased N_r of the second figure enables the characterization of higher-dimensional models, reflected by larger PI values for larger number of dimensions after projection N_d. This suggests that, as the traces used for profiling (and, later on, attacking) become less noisy, the LDA models are able to capture smaller details of the leakage distribution.

As a side observation, we note that in both cases, there are key bytes that stand up due to an increased PI. Yet, the key bytes that stand up differ for the $N_r = 1$ and $N_r = 1024$ cases. This also suggests that different parts of the traces can be exploited by the LDA models depending on the level of noise/averaging. For example, one possible explanation is that the leakage of bytes 3 and 15 is concentrated on less POIs with higher SNR values, explaining that they lead to the most informative models when $N_r = 1$. By contrast, the leakage of bytes 6 and 8 could be more spread in time, including POIs with lower SNR values that more significantly benefit from the increased averaging in Figure 7.

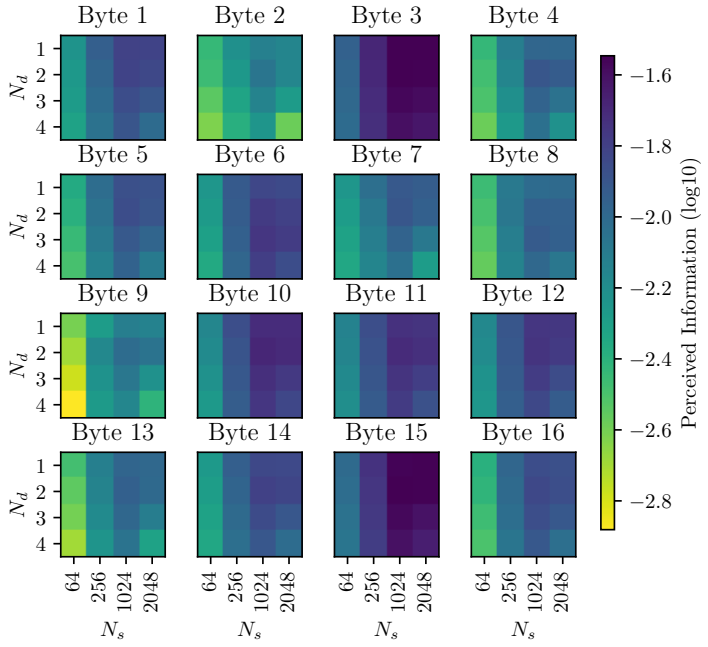


Figure 6: PI of the STM32F4 target for all the bytes and the whole parameter space with $N_r = 1$, displayed in log scale. As X-axis we display the number of selected Points-of-Interest (N_s) and as Y-axis the number of output dimensions (N_d) for the LDA. The PI is computed for one million traces corresponding to different plaintexts/keys.

6.2 Security evaluation

In order to embrace the more worst-case goal of our security evaluation for the LR-PRF, we first ran attacks on both the training set and the test set. Running attacks on the training set (i.e., with overfitting) is reminiscent of the TI metric and is aimed to provide a conservative bound on the adversary's success. Running attacks on the test set is rather connected to the PI metric and is therefore reflective of the models' generalization. We will additionally run attacks against an independent attack set (measured without interleaving) as part of the backwards security evaluation discussion of Subsection 6.4.

In order to assess the impact of the N_r parameter, we ran attacks considering three levels of repetition (i.e., $N_r \in [1, 32, 1024]$), using the models with the highest PI for all

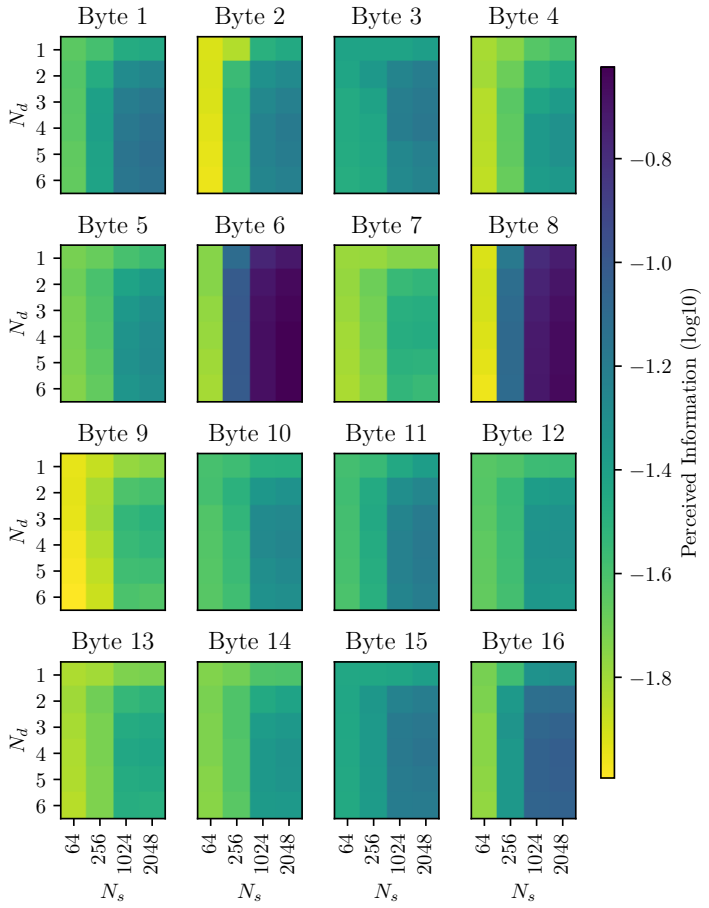


Figure 7: PI of the STM32F4 target for all the bytes and the whole parameter space with $N_r = 1024$, displayed in log scale. As X-axis we display the number of selected Points-of-Interest (N_s) and as Y-axis the number of output dimensions (N_d) for the LDA. The PI is computed for one million traces corresponding to different plaintexts/keys.

the key bytes, identified thanks to Figure 6 and Figure 7 for $N_r \in [1, 1024]$ and following a similar strategy for $N_r = 32$. The key ranks obtained on the training dataset and on the test dataset are given in Figures 8 and 9, respectively. The median and quartiles are estimated based on 100 independent attacks for each case (i.e., $N_r = 1, 32, 1024$).

The more realistic results of Figure 9 confirm attack complexities in the range of what is predicted by the inverse of the PI values in Subsection 6.1. The more conservative results of Figure 8 suggest that additional profiling efforts (with the LDA models we use) should only lead to limited improvements, which we will further discuss in the next section.

Most importantly, these figures highlight that for small number of observable plaintexts (i.e., n_a values), high key ranks are maintained despite increasing N_r. Assuming that we target a 96-bit security level, and that the gap between the measurement campaign of the two targets (i.e., one week vs. one month in our experiments) is not sufficient to perform 2^{96} AES encryptions (which seems reasonable with current computing power), it means that the LR-PRF maintains a key rank similar to the one of the protected STM32U5 target with n_a up to 4. Given that the protected AES core runs in 528 cycles and the LR-PRF runs in $11 \cdot \frac{128}{n_x}$ cycles, the cost vs. security tradeoff of the LR-PRF becomes

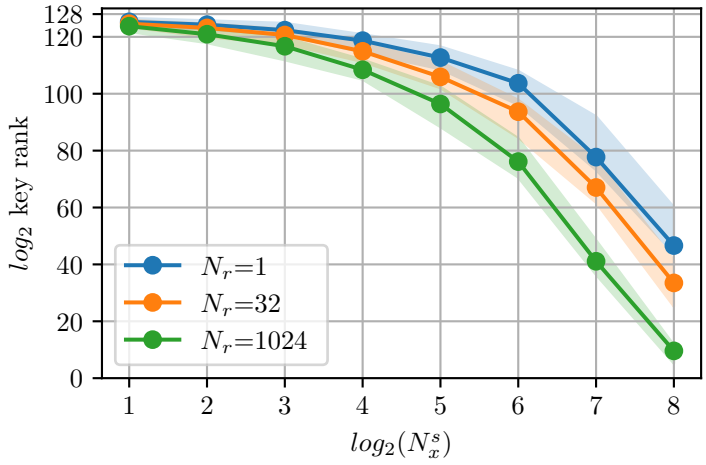


Figure 8: Median and quartiles of the log key rank estimated on the training set for the unprotected AES coprocessor of the STM32F4 (estimated from 100 independent attacks).

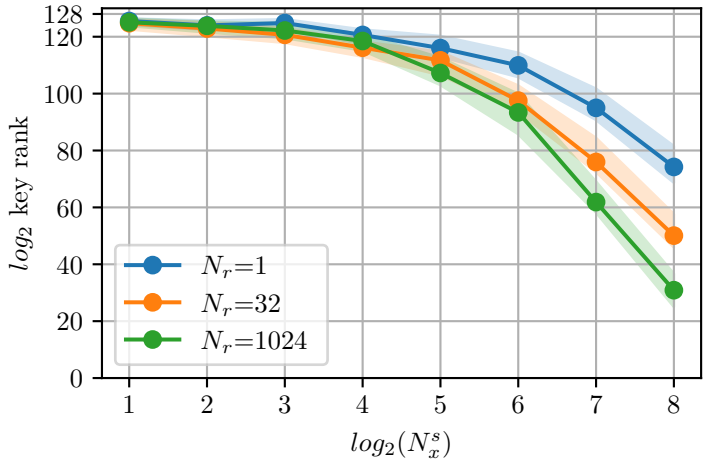


Figure 9: Median and quartiles of the log key rank estimated on the test set for the unprotected AES coprocessor of the STM32F4 (estimated from 100 independent attacks).

better starting from $n_x = 3$ and is therefore similar to slightly better in our study.[6]

For completeness, we succinctly report in Appendix A the results of an evaluation similar to the one of Figure 8 for the (32-bit) unprotected AES implementation available on the STM32U5. It confirms that, despite a slightly higher physical noise level (presumably due to a more recent technology), this coprocessor leads to lower estimated security levels when integrated in a leakage-resilient PRF. This confirms that the 128-bit implementation of the STM32F4 is a reasonable sweet spot for leveraging leakage-resilient designs, following theoretical expectations and experimental scaling trends put forward in [CDSU23].

6.3 Risk assessment

As in Section 5.3, we now discuss the risk of security overstatements in our evaluations.

Starting with the risks of improved measurement setups, they are similar. Improving them may lead to higher signal, more informative leakages and better attacks. So the LR-PRF does not bring improvements on this side. By contrast, we next argue that it comes with lower risks of improved statistical processing for two main reasons.

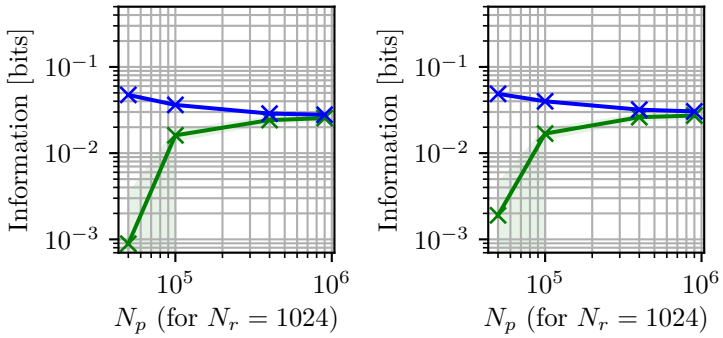
First, a look at the PI/TI convergence plots of Figure 10 highlights that the models used to evaluate the unprotected coprocessor of the STM32F4 are closer to profiling saturation than the models used to evaluate the protected coprocessor of the STM32U5, in Figure 5. We also confirm in the left part of Figure 11 that for relevant POIs used in the models of the STM32F4, the SNR approaches saturation with respect to repetition/averaging. The same holds for the PI metric and sets of 128 POIs around the SNR peaks in the right part of the figure. So our LDA-based attacks against this unprotected target could only be marginally improved with more evaluation efforts (i.e., larger N_p or N_r values). Second, this target is unlikely to exhibit fancy leakage distributions that would strongly benefit from advanced statistical methods as surveyed in [PPM+23]. In other words, it is likely that LDA models are well suited to estimate the side-channel security of an unprotected implementation and we do not expect significant gaps on this side either.

We therefore conclude that our evaluation of the STM32F4 AES coprocessor integrated in a LR-PRF is overall more confident than the one of the STM32U5 target.

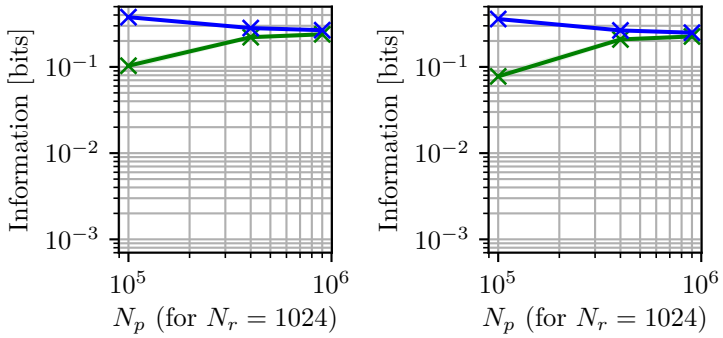
6.4 Backwards analysis

We conclude our investigations by analyzing a difference in the way we estimated security for our two targets. Namely, the investigations of Subsection 5.2 were based on more realistic attack conditions, in which the collection of the attack set was not interleaved with the one of the training and test sets By contrast, the attacks in this section were performed on the training set (as a bound) and on a test set collected in an interleaved manner. Yet, due to the strong incentive to increase the averaging factor N_r, one can wonder whether the very accurate models estimated on strongly averaged traces could lead to robustness issues. That is, as put forward in [RSV+11], it may happen that in case of discrepancies between a profiled model and the attack traces, keeping the traces (slightly) noisy can make them more robust to variations of the measurements' conditions. Informally, this is because the gap between the profiling sets and the test sets that could cause model overfitting could then be covered by a larger noise in the model.

[6] A bit more precisely, this comparison ignores the fact that the implementation of the STM32U5 includes some redundancy which we assume being due to countermeasures against fault attacks [JT12]. On the other hand, a leakage-resilient PRF integrated in a mode of operation may also offer security against certain fault attacks, as for example discussed by the ISAP designers [DEM+20]. Overall, our comparisons should anyway not be taken as strict indications of better or worse, but as evidence that the LR-PRF can be a competitive option when security against side-channel attacks is necessary.

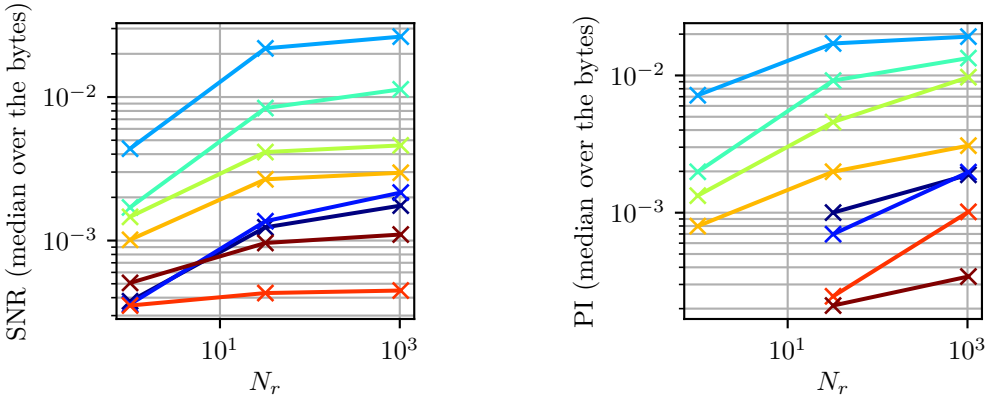


(a) STMF415, $N_r = 1$.



(b) STMF415, $N_r = 1024$.

Figure 10: PI (green) and TI (blue) for the most informative bytes on the STM32F4.



(a) Impact of N_r on the SNR.

(b) Impact of N_r on the PI.

Figure 11: Saturation of the median SNR (resp., PI) metrics when increasing N_r, for relevant POIs (resp., sets of POIs) of the STM32F4. Colors are the same as in Figure 2.

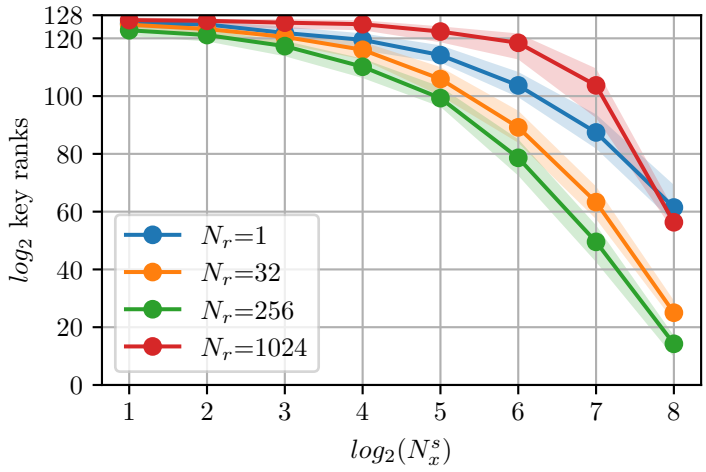


Figure 12: Median and quartiles of the log key rank estimated on the attack set for the unprotected AES coprocessor of the STM32F4 (estimated from 100 independent attacks).

We provide a preliminary investigation of this question with Figure 12, where the attacks against the STM32U5 target are performed on an independent attack dataset, of which the collection was not interleaved with the one of the training and test sets. It clearly illustrates that moving from $N_r = 256$ to $N_r = 1024$ affects the attack negatively in this more realistic scenario. Such an observation essentially backs up the backwards evaluation approach and shows that worst-case evaluations can come with mild additional security margins due to the powerful adversarial conditions they consider. Yet, the results of Figure 12 do not fundamentally affect our conclusions, neither qualitatively nor quantitatively. Hence, they essentially confirm that worst-case security evaluations as a useful shortcut to reach more confident conclusions on the side-channel security of a cryptographic implementation. It is an interesting open problem to investigate whether and how much robust profiling can cope with this model variability issue for larger N_r values [WO15, CK18].

7 Discussion

The results in the previous sections exhibit an interesting case study where the simplicity of the LR-PRF leveraging an unprotected AES coprocessor does not come at the cost of a reduced security compared to a coprocessor protected with implementation-level countermeasures. It naturally raises as next question: *could this case be generalized?*

Such a generalization is unfortunately hard to formalize with existing tools. The main reason is that both the LR-PRF and Boolean masking crucially rely on technology-dependent hardware features for which the cost vs. security tradeoff is hard to extrapolate. For the LR-PRF (resp., Boolean masking), security roughly depends on the amount of information leaked on the key by average leakage traces obtained thanks to repetition (resp., noisy leakage traces). But reaching sufficiently low information values so that these countermeasures are effective can be achieved with a variety of means, which do not come with simple (cost vs. security) scaling trends. For the LR-PRF, the parallelism on which we rely in this paper is a natural choice but, for example, dual-rail logic styles could be an alternative [TV03]. For masked implementations, parallelism and dual-rail logic styles can help as well, but the design space is larger: noise/jitter engines as in the STM32U5 chip are an option, but other randomization techniques like RDIs [CK09, CK10] or shuffling [VMKS12] are eligible too. So we do not believe there is currently a sound

way to conclude that using LR-PRFs or masking is always preferable in terms of cost vs. security tradeoff, and the reason to choose one or the other are to find elsewhere.

For the masking countermeasure, the advantages and disadvantages are well-known. First, it is in general (i.e., outside the context of this paper where we rely on existing COTS devices only) a flexible solution which does not require a change of primitive or mode of operation. Masking is applicable to existing standards and it enables devices with different side-channel security levels to communicate with overheads adapted in function of whether adversaries can access them physically. This is in contrast with LR-PRFs for which the overheads always have to be paid. Second, and as just mentioned, masking comes with a wide design space: it can be combined with a broad literature aiming at reducing the side-channel signal or increasing the noise, and it has been optimized for various cost metrics such as randomness [BBP+16, DSM22] or latency [KM22, CSV24]. Its main disadvantage is that the design and evaluation of masked implementations are challenging and require high expertise. From the design viewpoint, it implies dealing with the various physical defaults that can break masking's independence assumption such as glitches [MPG05, MPO05] or transitions [CGP+12, BGG+14], which leads to additional overheads [NRS11, CS21]. From the evaluation viewpoint, it implies assessing a security order [SM16] and estimating a mixture distribution corresponding to this security order, which can be difficult tasks, especially in a closed-source / non-worst-case evaluation context [BDMS22].

LR-PRFs essentially offer a complementary picture. Namely, they provide a welcome simplicity and can directly leverage the unprotected hardware (e.g., AES) coprocessors that are already widely deployed in the embedded security market. Their design is simple and the gap between a functionally correct implementation of a LR-PRF and a secure one is expected to be smaller when relying on hardware coprocessors. Their evaluation is simple as well since it can be based on first-order statistics and it benefits from useful extrapolation tools. So while there are certainly contexts in which the flexibility of masking is a must (e.g., if side-channel countermeasures are only needed for a few devices in a system), we believe there are also many contexts in which LR-PRFs could contribute to improve physical security. IoT applications where secure coprocessors and/or the expertise required to implement masking securely may be too expensive are a good example.

So overall, the informal conclusion of the paper is that masking is not the only option in the context of end users willing to leverage COTS devices for side-channel secure symmetric cryptography. In this respect, and despite more results to confirm our observations with other unprotected coprocessors would be welcome, we finally note that the lack of standard solutions is an important hurdle preventing the more widespread use of primitive-level countermeasures like LR-PRFs. We therefore believe that considering the standardization of such modes could be a good trigger towards more work on this important topic, and is anyway a necessary step towards making them usable in an industrial context. It could also motivate COTS devices' manufacturers to design new implementations suited to be integrated leakage-resistant schemes. Modes like the one-pass LR-BC-2, which provides strong integrity against leakage, or the two-pass LR-BC-3, which additionally provides confidentiality against leakage, could serve as an AES-based basis for this purpose [BMPS21]. Similarly, ISAP could serve as a basis for a permutation-based alternative, when coprocessors for the Ascon permutation become available [DEM+20].

Acknowledgments

F.-X. Standaert is a research director of the Belgian Fund for Scientific Research (FNRS-F.R.S.). This work has been funded in parts by European Union through the Horizon project 1010706275 (acronym REWIRE) and the ERC Advanced Grant 101096871 (acronym BRIDGE). Views and opinions expressed are those of the authors and do not necessarily

reflect those of the European Union or the ERC. Neither the European Union nor the granting authority can be held responsible for them.

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.

[ABF13] Michel Abdalla, Sonia Belaïd, and Pierre-Alain Fouque. Leakage-resilient symmetric encryption via re-keying. In *CHES*, volume 8086 of *Lecture Notes in Computer Science*, pages 471–488. Springer, 2013.

[BBC+20] Davide Bellizia, Olivier Bronchain, Gaëtan Cassiers, Vincent Grosso, Chun Guo, Charles Momin, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Mode-level vs. implementation-level physical security in symmetric cryptography - A practical guide through the leakage-resistance jungle. In *CRYPTO (1)*, volume 12170 of *Lecture Notes in Computer Science*, pages 369–400. Springer, 2020.

[BBP+16] Sonia Belaïd, Fabrice Benhamouda, Alain Passelègue, Emmanuel Prouff, Adrian Thillard, and Damien Vergnaud. Randomness complexity of private circuits for multiplication. In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 616–648. Springer, 2016.

[BCG+23] Julien Béguinot, Wei Cheng, Sylvain Guilley, Yi Liu, Loïc Masure, Olivier Rioul, and François-Xavier Standaert. Removing the field size loss from duc et al.'s conjectured bound for masked encodings. In *COSADE*, volume 13979 of *Lecture Notes in Computer Science*, pages 86–104. Springer, 2023.

[BDMS22] Olivier Bronchain, François Durvaux, Loïc Masure, and François-Xavier Standaert. Efficient profiled side-channel analysis of masked implementations, extended. *IEEE Trans. Inf. Forensics Secur.*, 17:574–584, 2022.

[BGG+14] Josep Balasch, Benedikt Gierlichs, Vincent Grosso, Oscar Reparaz, and François-Xavier Standaert. On the cost of lazy engineering for masked software implementations. In *CARDIS*, volume 8968 of *Lecture Notes in Computer Science*, pages 64–81. Springer, 2014.

[BGS15] Sonia Belaïd, Vincent Grosso, and François-Xavier Standaert. Masking and leakage-resilient primitives: One, the other(s) or both? *Cryptogr. Commun.*, 7(1):163–184, 2015.

[BHM+19] Olivier Bronchain, Julien M. Hendrickx, Clément Massart, Alex Olshevsky, and François-Xavier Standaert. Leakage certification revisited: Bounding model errors in side-channel security evaluations. In *CRYPTO (1)*, volume 11692 of *Lecture Notes in Computer Science*, pages 713–737. Springer, 2019.

[BMPS21] Olivier Bronchain, Charles Momin, Thomas Peters, and François-Xavier Standaert. Improved leakage-resistant authenticated encryption based on hardware AES coprocessors. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):641–676, 2021.

[BSH+14] Sonia Belaïd, Fabrizio De Santis, Johann Heyszl, Stefan Mangard, Marcel Medwed, Jörn-Marc Schmidt, François-Xavier Standaert, and Stefan Tillich. Towards fresh re-keying with leakage-resilient prfs: cipher design principles and analysis. *J. Cryptogr. Eng.*, 4(3):157–171, 2014.

[BUS21] Davide Bellizia, Balazs Udvarhelyi, and François-Xavier Standaert. Towards a better understanding of side-channel analysis measurements setups. In *CARDIS*, volume 13173 of *Lecture Notes in Computer Science*, pages 64–79. Springer, 2021.

[CB23] Gaëtan Cassiers and Olivier Bronchain. Scalib: A side-channel analysis library. *Journal of Open Source Software*, 8(86):5196, 2023.

[CBG+17] Thomas De Cnudde, Begül Bilgin, Benedikt Gierlichs, Ventzislav Nikov, Svetla Nikova, and Vincent Rijmen. Does coupling affect the security of masked implementations? In *COSADE*, volume 10348 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2017.

[CDSU23] Gaëtan Cassiers, Henri Devillez, François-Xavier Standaert, and Balazs Udvarhelyi. Efficient regression-based linear discriminant analysis for side-channel security evaluations towards analytical attacks against 32-bit implementations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(3):270–293, 2023.

[CEM18] Thomas De Cnudde, Maik Ender, and Amir Moradi. Hardware masking, revisited. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):123–148, 2018.

[CGP+12] Jean-Sébastien Coron, Christophe Giraud, Emmanuel Prouff, Soline Renner, Matthieu Rivain, and Praveen Kumar Vadnala. Conversion of security proofs from one leakage model to another: A new issue. In *COSADE*, volume 7275 of *Lecture Notes in Computer Science*, pages 69–81. Springer, 2012.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.

[CK09] Jean-Sébastien Coron and Ilya Kizhvatov. An efficient method for random delay generation in embedded software. In *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 2009.

[CK10] Jean-Sébastien Coron and Ilya Kizhvatov. Analysis and improvement of the random delay countermeasure of CHES 2009. In *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 95–109. Springer, 2010.

[CK18] Marios O. Choudary and Markus G. Kuhn. Efficient, portable template attacks. *IEEE Trans. Inf. Forensics Secur.*, 13(2):490–501, 2018.

[CRBO24] Aakash Chowdhury, Arnab Roy, Carlo Brunetta, and Elisabeth Oswald. Leakage certification made simple. In *CRYPTO (6)*, volume 14925 of *Lecture Notes in Computer Science*, pages 427–460. Springer, 2024.

[CRR02] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.

[CS21] Gaëtan Cassiers and François-Xavier Standaert. Provably secure hardware masking in the transition- and glitch-robust probing model: Better safe than sorry. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(2):136–158, 2021.

[CSV24] Gaëtan Cassiers, François-Xavier Standaert, and Corentin Verhamme. Low-latency masked gadgets robust against physical defaults with application to ascon. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2024(3):603–633, 2024.

[dCGRP19] Eloi de Chérisey, Sylvain Guilley, Olivier Rioul, and Pablo Piantanida. Best information is most successful mutual information and success rate in side-channel analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):49–79, 2019.

[DEM+20] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, Robert Primas, and Thomas Unterluggauer. Isap v2.0. *IACR Trans. Symmetric Cryptol.*, 2020(S1):390–416, 2020.

[DEMS21] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2: Lightweight authenticated encryption and hashing. *J. Cryptol.*, 34(3):33, 2021.

[DFH+16] Stefan Dziembowski, Sebastian Faust, Gottfried Herold, Anthony Journault, Daniel Masny, and François-Xavier Standaert. Towards sound fresh re-keying with hard (physical) learning problems. In *CRYPTO (2)*, volume 9815 of *Lecture Notes in Computer Science*, pages 272–301. Springer, 2016.

[DFS15] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete - or how to evaluate the security of any leaking device. In *EUROCRYPT (1)*, volume 9056 of *Lecture Notes in Computer Science*, pages 401–429. Springer, 2015.

[DKM+15] Christoph Dobraunig, François Koeune, Stefan Mangard, Florian Mendel, and François-Xavier Standaert. Towards fresh and hybrid re-keying schemes with beyond birthday security. In *CARDIS*, volume 9514 of *Lecture Notes in Computer Science*, pages 225–241. Springer, 2015.

[DLMV05] Rémy Daudigny, Hervé Ledig, Frédéric Muller, and Frédéric Valette. SCARE of the DES. In *ACNS*, volume 3531 of *Lecture Notes in Computer Science*, pages 393–406, 2005.

[DMMS21] Sébastien Duval, Pierrick Méaux, Charles Momin, and François-Xavier Standaert. Exploring crypto-physical dark matter and learning with physical rounding towards secure and efficient fresh re-keying. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(1):373–401, 2021.

[DP10] Yevgeniy Dodis and Krzysztof Pietrzak. Leakage-resilient pseudorandom functions and side-channel attacks on feistel networks. In *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2010.

[DR20] Joan Daemen and Vincent Rijmen. *The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition.* Information Security and Cryptography. Springer, 2020.

[DSM22] Siemen Dhooghe, Aein Rezaei Shahmirzadi, and Amir Moradi. Second-order low-randomness d + 1 hardware sharing of the AES. In *CCS*, pages 815–828. ACM, 2022.

[FPS12] Sebastian Faust, Krzysztof Pietrzak, and Joachim Schipper. Practical leakage-resilient symmetric cryptography. In *CHES*, volume 7428 of *Lecture Notes in Computer Science*, pages 213–232. Springer, 2012.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *FOCS*, pages 464–479. IEEE Computer Society, 1984.

[GP99] Louis Goubin and Jacques Patarin. DES and differential power analysis (the "duplication" method). In *CHES*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer, 1999.

[HMM+23] Clément Hoffmann, Pierrick Méaux, Charles Momin, Yann Rotella, François-Xavier Standaert, and Balazs Udvarhelyi. Learning with physical rounding for linear and quadratic leakage functions. In *CRYPTO (3)*, volume 14083 of *Lecture Notes in Computer Science*, pages 410–439. Springer, 2023.

[ISW03] Yuval Ishai, Amit Sahai, and David A. Wagner. Private circuits: Securing hardware against probing attacks. In *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.

[IUH22] Akira Ito, Rei Ueno, and Naofumi Homma. On the success rate of side-channel attacks on masked implementations: Information-theoretical bounds and their practical usage. In *CCS*, pages 1521–1535. ACM, 2022.

[JT12] Marc Joye and Michael Tunstall, editors. *Fault Analysis in Cryptography*. Information Security and Cryptography. Springer, 2012.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[KM22] David Knichel and Amir Moradi. Low-latency hardware private circuits. In *CCS*, pages 1799–1812. ACM, 2022.

[Man02] Stefan Mangard. A simple power-analysis (SPA) attack on implementations of the AES key expansion. In *ICISC*, volume 2587 of *Lecture Notes in Computer Science*, pages 343–358. Springer, 2002.

[Man04] Stefan Mangard. Hardware countermeasures against DPA ? A statistical analysis of their effectiveness. In Tatsuaki Okamoto, editor, *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004.

[MCHS23] Loïc Masure, Gaëtan Cassiers, Julien M. Hendrickx, and François-Xavier Standaert. Information bounds and convergence rates for side-channel security evaluators. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(3):522–569, 2023.

[MPG05] Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-channel leakage of masked CMOS gates. In *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005.

[MPO05] Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully attacking masked AES hardware implementations. In *CHES*, volume 3659 of *Lecture Notes in Computer Science*, pages 157–171. Springer, 2005.

[MSGR10] Marcel Medwed, François-Xavier Standaert, Johann Großschädl, and Francesco Regazzoni. Fresh re-keying: Security against side-channel and fault attacks for low-cost devices. In *AFRICACRYPT*, volume 6055 of *Lecture Notes in Computer Science*, pages 279–296. Springer, 2010.

[MSJ12] Marcel Medwed, François-Xavier Standaert, and Antoine Joux. Towards super-exponential side-channel security with efficient leakage-resilient prfs. In *CHES*, volume 7428 of *Lecture Notes in Computer Science*, pages 193–212. Springer, 2012.

[MSNF16] Marcel Medwed, François-Xavier Standaert, Ventzislav Nikov, and Martin Feldhofer. Unknown-input attacks in the parallel setting: Improving the security of the CHES 2012 leakage-resilient PRF. In *ASIACRYPT (1)*, volume 10031 of *Lecture Notes in Computer Science*, pages 602–623, 2016.

[NRS11] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of nonlinear functions in the presence of glitches. *J. Cryptol.*, 24(2):292–321, 2011.

[PPM+23] Stjepan Picek, Guilherme Perin, Luca Mariot, Lichao Wu, and Lejla Batina. Sok: Deep learning-based physical side-channel analysis. *ACM Comput. Surv.*, 55(11):227:1–227:35, 2023.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 142–159. Springer, 2013.

[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.

[RPD09] Matthieu Rivain, Emmanuel Prouff, and Julien Doget. Higher-order masking and shuffling for software implementations of block ciphers. In *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2009.

[RR13] Matthieu Rivain and Thomas Roche. SCARE of secret ciphers with SPN structures. In *ASIACRYPT (1)*, volume 8269 of *Lecture Notes in Computer Science*, pages 526–544. Springer, 2013.

[RSV+11] Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A formal study of power variability issues and side-channel attacks for nanoscale devices. In *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 109–128. Springer, 2011.

[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.

[SIH+23] Emanuele Strieder, Manuel Ilg, Johann Heyszl, Florian Unterstein, and Silvan Streit. ASCA vs. SASCA - A closer look at the AES key schedule. In *COSADE*, volume 13979 of *Lecture Notes in Computer Science*, pages 65–85. Springer, 2023.

[SM16] Tobias Schneider and Amir Moradi. Leakage assessment methodology - extended version. *J. Cryptogr. Eng.*, 6(2):85–99, 2016.

[SMY09] François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.

[SPY+10] François-Xavier Standaert, Olivier Pereira, Yu Yu, Jean-Jacques Quisquater, Moti Yung, and Elisabeth Oswald. Leakage resilient cryptography in practice. In *Towards Hardware-Intrinsic Security*, Information Security and Cryptography, pages 99–134. Springer, 2010.

[Sta24] François-Xavier Standaert. *Side-Channel Analysis and Leakage-Resistance*. Version 1.2, September 2024.

[TV03] Kris Tiri and Ingrid Verbauwhede. Securing encryption algorithms against DPA at the logic level: Next generation smart card technology. In *CHES*, volume 2779 of *Lecture Notes in Computer Science*, pages 125–136. Springer, 2003.

[UBS21] Balazs Udvarhelyi, Olivier Bronchain, and François-Xavier Standaert. Security analysis of deterministic re-keying with masking and shuffling: Application to ISAP. In *COSADE*, volume 12910 of *Lecture Notes in Computer Science*, pages 168–183. Springer, 2021.

[UHIM24] Rei Ueno, Naofumi Homma, Akiko Inoue, and Kazuhiko Minematsu. Fallen sanctuary: A higher-order and leakage-resilient rekeying scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2024(1):264–308, 2024.

[VMKS12] Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. Shuffling against side-channel attacks: A comprehensive study with cautionary note. In *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 740–757. Springer, 2012.

[WO15] Carolyn Whitnall and Elisabeth Oswald. Robust profiling for dpa-style attacks. In *CHES*, volume 9293 of *Lecture Notes in Computer Science*, pages 3–21. Springer, 2015.

[YS13] Yu Yu and François-Xavier Standaert. Practical leakage-resilient pseudorandom objects with minimum public randomness. In *CT-RSA*, volume 7779 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2013.

A Evaluation of the STM32U5's unprotected core

The STM32U5 embeds an unprotected AES hardware accelerator in addition to the protected one. It is hinted to be more serialized due to the disclosed latency of 51 cycles for a single execution of AES-ECB. It is therefore expected to be less amenable to integration in a LR-PRF design than the more parallel STM32F4 core (running in 11 cycles). We confirm this claim in Figure 13, which displays an analysis similar to the one of Figure 8 for this more serial unprotected core. It exhibits consistently lower security levels. Note that these results hold despite the presence of more physical noise than for the measurements of the STM32F4, presumably due to a more recent technology. To confirm this, we computed the noise variance of the two targets, both without algorithmic noise (i.e., by fixing the full AES state) and with algorithmic noise (i.e., by fixing only a byte of the AES state). Without algorithmic noise, the variance of the STM32U5 is 1.6 times larger than the one of the STM32F4. With algorithmic noise, this factor drops to 0.65. This experiment confirms that the STM32F4 coprocessor is a better candidate for integration in a LR-PRF thanks to higher parallelism and highlights that comparisons between masking and LR-PRFs for COTS devices can only be based on a best effort selection of implementations.

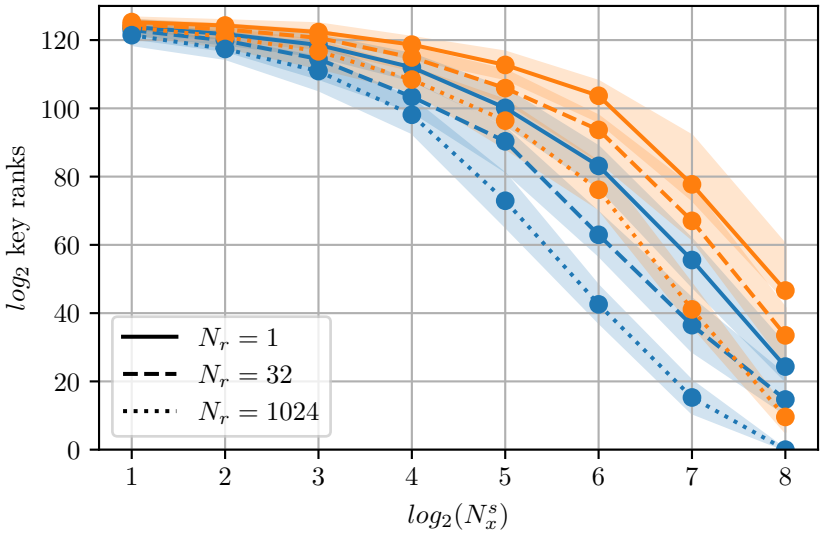


Figure 13: Median and quartiles of the log key rank estimated (from 100 independent attacks) on the training set for the STM32F4's unprotected coprocessor (in orange) and the STM32U5's unprotected coprocessor (in blue), both to be integrated in a LR-PRF.

B SNR of the STM32U5 key scheduling

For completeness, we also provide a wider view of the SNR values that we computed for all the rounds and the key rounds of the protected AES core in Figure 14. It leads to a few additional observations. First, the SNR values of the key rounds are significantly larger than the ones of the rounds. It may be caused by additional algorithm-level countermeasures (presumably, Boolean masking) that are only added for the more sensitive round operations of the AES. This is because round operations are DPA targets whereas the key round leakage can only be exploited by a Simple Power Analysis (SPA). Since our work is focused on DPA, it is an interesting open problem to find out whether the key scheduling leakage of the STM32U5 can be exploited with attacks like [Man02, SIH+23]. Second, we see that the SNR peaks are repeated twice in both plots. We assume that this is due to redundant computations performed in order to mitigate fault attacks [JT12], and the lower SNR values for the right patterns in the figure is due to jitter in the traces.

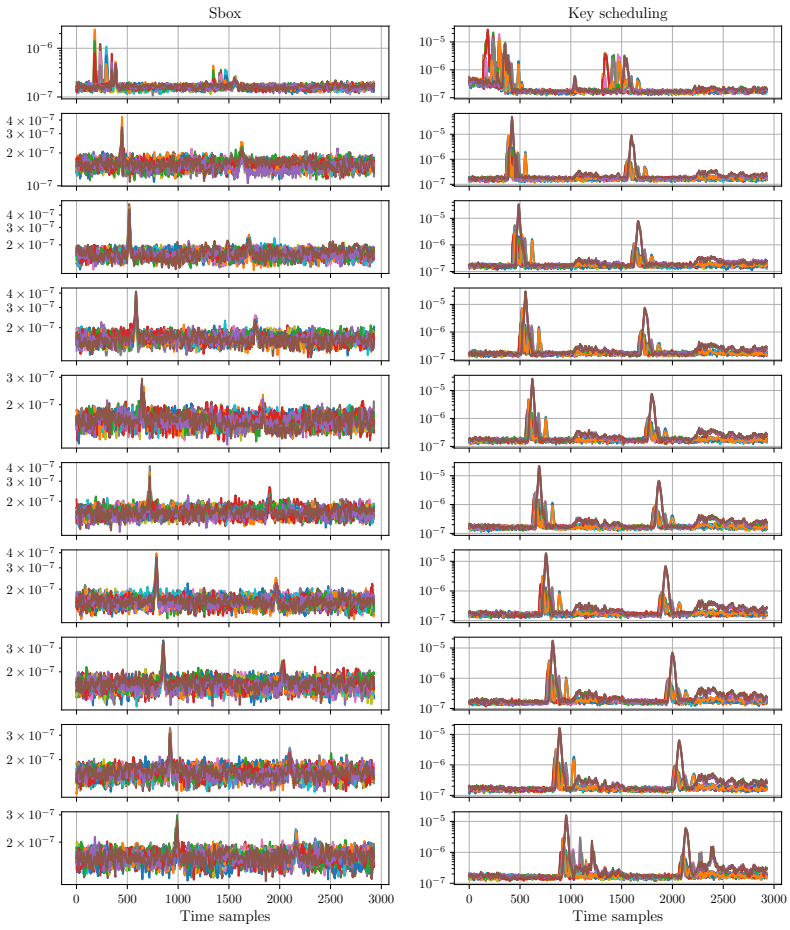


Figure 14: SNR of the rounds vs. key rounds (STM32U5, using $1.5 \cdot 10^9$ traces).