On the Security of Off-the-Shelf Microcontrollers: Hardware is not Enough

Balazs Udvarhelyi, Antoine van Wassenhove,
Olivier Bronchain, and François-Xavier Standaert

Crypto Group, ICTEAM Institute, UCLouvain, Louvain-la-Neuve, Belgium.
firstname.lastname@uclouvain.be

Abstract. We complete the state-of-the-art on the side-channel security of real-world devices by analysing two 32-bit microcontrollers equiped with an unprotected co-processor. Our results show that (i) the lack of understanding of their hardware architecture can be circumvented with standard detection tools – for this purpose, we combine a simple variation of the Test Vector Leakage Assessment methodology with Signal-to-Noise Ratio estimations, which enables the efficient identification of attack vectors; (ii) standard distinguishers then lead to powerful key recoveries with less than 5,000 traces; and (iii) preprocessing like the continuous wavelet transform can be useful in such a black box evaluation context.

1 Introduction

Side-channel analysis is known to be a threat to the security of embedded systems. It has been the topic of intensive academic research over the last two decades and many powerful attacks have been put forward. Yet, most academic results are performed in well understood and controlled environments, and public security analyses of real-world deployed products is more sporadic. Among the examples of such more realistic attacks we are aware of, we note the ones against Keeloq [13], the Xilinx bitstream encryption [18,19], SIM cards [29,15], Hue Smart lamps [21] and Thread communication stacks [11]. Their common denominator is the presence of a long-term key shared among many devices, so that its recovery can be used to decrypt secret communications, forge software updates or clone devices. Technically, performing such attacks typically requires a (possibly long and tedious) reverse engineering effort, because of either badly documented components or limited public implementation details, which is followed by a side-channel attack. In all the aforementioned cases, the products turned out to be unprotected, leading to quite straightforward weaknesses.

We complete this state-of-the-art and extend the investigation of side-channel attacks against real-world devices to 32-bit microcontrollers (MCU), that are becoming increasingly popular for lightweight embedded systems. In particular, we study the side-channel resistance of two off-the-shelf Cortex-M4 devices that include a hardware co-processor. We focus on MCUs from two different manufacturers, namely NXP Semiconductors (NXP) and STMicroelectronics (STM). We insist that none of them claimed to provide a high physical security level (although the NXP device includes some light countermeasure [23]). So our main

goal is to evaluate the technical difficulty to identify attack vectors despite lacking a precise description of the target hardware architectures.

Our investigations show that this limited information does not prevent the identification of simple yet powerful attacks (while a better understanding of the targets could lead to further optimizations [6]). For this purpose, we first propose a variant of the Test Vector Leakage Assessment (TVLA) methodology, which we denote as "one-hot" and is well suited to the analysis of hardware implementations: it allows us to locate some target operations and to infer the co-processor architecture (e.g., the degree of parallelism) with low data complexity. We next show that standard distinguishers are sufficient to recover the full encryption key in less than 5,000 (power or EM) measurements. As an additional contribution, we put forward the interest of the continuous wavelet transform for such a black box security evaluation (as previously proposed in [10]).

2 Background

This section contains the necessary background used in the rest of the paper. We first introduce the notations. Second, two widely spread detections tools are recalled. Third, two profiled side-channel attacks are described. Finally, we detail our methodology for the pre-processing of the traces.

2.1 Notations

For the rest of the paper, random variables are denoted by a capital letter X and their realisations with x. The statistical expectation is denoted as $\mathsf{E}[\cdot]$ and estimators are denoted with a hat. A leakage trace will be written as $\boldsymbol{l}$.

2.2 Detection tools

The first step in side channel analysis is the detection of Points-Of-Interest (POIs). It consists in learning the location of the sensible information within the leakage traces. To do so, we make use of two methods.

The first one is a slightly tweaked version of the TVLA in [14,9] which is based on Welch's t-test [28]. The t-test is a statistical test used to highlight a difference between the means of two populations respectively denoted as μ_1 and μ_2. The test is performed by computing

$$t = \frac{\hat{\mu}_1 - \hat{\mu}_2}{\sqrt{\frac{\hat{\sigma}_1^2}{N_1} + \frac{\hat{\sigma}_2^2}{N_2}}}, \tag{1}$$

where σ_i and N_i are the standard deviation and the sample size of the population i. If the maximum $|t|$ is larger than 4.5, a difference between the means is very likely to be present (with a p-value smaller than 10^{-5}). In a side-channel context, it is generally used to highlight dependencies between the mean of the traces

and the manipulated variables. To do so, the two tested populations are leakage traces $\boldsymbol{l}_i$ for two different sets of inputs. The t-test is then applied to all the time samples of the leakage traces independently.

The second detection tool we use is the Signal-to-Noise Ratio (SNR) [16]. It aims at quantifying the available signal about one (secret) intermediate variable within the side-channel measurements. Concretely, the SNR of a target intermediate variable X is estimated as

$$\hat{\mathrm{SNR}} = \frac{\hat{\mathsf{Var}}[\hat{\mu}_x]}{\hat{\mathsf{E}}[\hat{\sigma}_x^2]}\,, \tag{2}$$

where the estimated mean and variance of each possible value of X are denoted as $\hat{\mu}_x$, $\hat{\sigma}_x^2$. Similarly to the TVLA, the SNR is computed for every time sample. This metric is significant only at the locations where X (or values injectively depending of it) is (are) manipulated, which are its associated POIs.

2.3 Side-channel distinguishers

In this paper, key recovery attacks are performed against two targets. To do so, we make use of two distinguishers, namely Gaussian Template Attacks (TA) [7] and Correlation Power Analysis (CPA) [5]. Both are *Divide & Conquer*: they target the 16 bytes of the master key independently.

First, the TA is performed in two steps. It starts with a profiling phase. During this step, leakage traces and the corresponding plaintexts and keys are given to the adversary. Based on these, it estimates a Probability Density Function (PDF) of the leakage $\boldsymbol{l}$ given an intermediate state x at the POI. The leakage distribution is assumed to be Gaussian, so that this conditional distribution is

$$\hat{\mathsf{f}}[l|x] = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}_x|}} \exp\left(-\frac{1}{2}(\boldsymbol{l}-\boldsymbol{\mu}_x)'\boldsymbol{\Sigma}_x(\boldsymbol{l}-\boldsymbol{\mu}_x)\right)\,, \tag{3}$$

where $\boldsymbol{\mu}_x$ and $\boldsymbol{\Sigma}_x$ are the estimated mean vector and covariance matrix for the leakages of x. Next, during the attack phase, leakage traces and only the corresponding plaintext are given. Based on these and the estimated PDF, the adversary uses Bayes' theorem to estimate $\Pr[x|\boldsymbol{l}]$ for each trace independently. From n measurements, he infers the key byte with maximum likelihood as

$$\hat{k} = \underset{k^*}{\mathrm{argmax}} \prod_{i=1}^{n} \Pr[p_i, k^*|\boldsymbol{l}_i]\,, \tag{4}$$

where p_i is the plaintext corresponding to the $\boldsymbol{l}_i$ trace.

Second, the CPA is exploiting Pearson's Correlation $\hat{\rho}(\cdot,\cdot)$ between the observed traces $\boldsymbol{l}$ and a key-dependent leakage model $\boldsymbol{M}_{k^*,p}$ [5]. The inferred key byte is the one leading to the highest correlation such that

$$\hat{k} = \underset{k^*}{\mathrm{argmax}} \quad \hat{\rho}(\boldsymbol{M}_{k^*,p}, \boldsymbol{l}). \tag{5}$$

Informally, it is expected that the most accurate model will be the one of the correct key. The model can be selected based on engineering intuition (e.g., assuming the leakages to be proportional to the Hamming weight of x), which we denote as the non-profiled CPA. In this paper, the model is rather profiled and corresponds to the estimated mean of the output of the first Sbox (i.e., the vectors $\boldsymbol{\mu}_x$ of the TA). Compared to the profiled CPA, TA have the ability to exploit multivariate leakages. In a univariate setting with a sufficiently noisy environment, these two are equivalent [17].

2.4 Pre-processing tools

Before launching the previously mentioned attacks on actual leakage traces, some pre-processing stages can be implemented. These can be used both for *noise reduction* and/or for *dimensionality reduction*.

Continuous Wavelet Transform. In order to reduce the impact of noise on the measurements, a Continuous Wavelet Transform (CWT) can be used [10]. Similarly to Fourier Transforms, the CWT is a representation of a given signal in another domain. The CWT domain can be interpreted as the frequency content (f) across time (t'). Formally, the CWT of a time signal $x(t)$ is written as

$$X_\omega(f, t') = \frac{1}{|f|} \int_{-\infty}^{\infty} x(t) \cdot \psi\left(\frac{t - t'}{f}\right) dt\,, \tag{6}$$

where $\psi(\cdot)$ is a given wavelet. It is therefore the convolution of the signal with a scaled wavelet. Several wavelets have been tested. The best results were obtained with the Ricker wavelet which we use for the rest of the paper. For computational reasons, we limited the wavelet width to the smallest one that was maintaining the signal. In the side-channel context, a CWT can be applied to the traces before any other processing.[1] Since the signal manipulated is then in the CWT domain, an SNR computed on it will highlight where the key-dependent signal lies across both time and frequency. This allows removing frequencies and time samples that are not signal-dependent, making it useful for noise reduction.

Principal Component Analysis. In order to reduce the number of samples over which an attack is executed, a Principal Component Analysis (PCA) can be used [1]. It is a profiled dimensionality reduction tool that takes high dimensional signals and reduces them to a smaller, chosen number of dimensions. In the side-channel context, it is typically applied to the mean vectors, which maximizes the inter-class variance. Practically, before applying PCA, the SNR can be used (and was used in our experiments) to find and apply the PCA only on the POIs. This allows to speed up the convergence of the PCA.

[1] Concretely, we only evaluate Equation (6) at a finite number of coordinates since exploring the entire continuous domain is unpractical.

3 Targets and setup

First, this section gives a rationale behind the choice of two MCUs as well as their specificities. Second, it describes the conditions under which these were monitored during their security evaluation.

3.1 Targets

In this study, we are interested in the security of AES co-processors in low-cost off-the-shelf components. We found out that `ARM Cortex-M4` devices are among the cheapest components with widely spread AES hardware acceleration. For diversity, we chose one component fulfilling these criteria from two well-known manufacturers in the MCU industry, namely NXP and STM.

NXP Kinetis. The selected MCU from NXP is the Kinetis K82 `MK82FN256-VLL15` [23]. It comes with two cryptographic co-processors. The one under investigation is the `LP Trusted Cryptography` module. It reports a countermeasure that inserts noise into the power consumption with a random mask [24]. A `DPAMaskSeed` register is present to reseed the core, which is advised after 50,000 encryptions. This target has been mounted on a custom Printed Circuit Board (PCB) in order to limit potential noise due to additional components.

STM32. The selected MCU from STM is the `STM32L422CB` [25]. The AES co-processor of this processor does not have countermeasures against side-channel attacks mentioned. The target has been mounted on a custom PCB too.

3.2 Evaluation setup

In order to cover a good range of threat models, we evaluated the two aforementioned targets under different conditions, also reflecting the fact that such low-cost MCUs can be used for a wide range of applications, going from low-energy to more computationally intensive tasks.

The first parameter of our evaluations is the clock frequency. Targets were evaluated with a low clock frequency of 8[MHz] as well as close to their maximum frequency (i.e., 100[MHz] for the NXP target and 80[MHz] for the STM target). In both cases, the clock is derived from a 8[MHz] on-board crystal.

Second, both the electromagnetic (EM) emanations and the power consumption were recorded for the evaluations. These two signals are simultaneously measured by a `Picoscope 5244d` at 500[MSample/s] with an 8-bit resolution. The EM leakage was obtained using an H near-field probe from the HZ-15, probe set from Rohde&Schwartz. The EM probe used was impedance matched and preamplified using the Rohde&Schwartz HZ-16 preamplifier. Several positions were tested by hand for the probes and the position with the highest signal was kept. Therefore, the measurement was done above the target for the NXP MCU.

For the STM, no exploitable emanations were observed above the chip. Hence, the measurements were made thanks to the emanations of a power line.

The power consumption was measured through a shunt resistor and without amplification. The resistor is of 5[Ohm] for the NXP target and 10[Ohm] for the STM target. These values were chosen as high as possible, leading to a greater signal, but without triggering a brown out reset. The MCUs are accessing the AES cores using the hardware abstraction layer (HAL) published by the manufacturers. The scope is triggered just before the HAL call.

4 Architecture inference

In absence of detailed specifications of the target hardware architectures, a first step in the following side-channel attacks is to infer a sufficient understanding enabling us to identify good target intermediate variables. We next describe our methodology for this purpose, followed by its results on the two targets.

At a high-level, we use a (fast) variant of the TVLA to locate the encryption and the execution of the first-round Sboxes. Then, we use the (slower) SNR to precisely identify the time samples corresponding to each key byte.

4.1 Methodology

In order to detect POIs as well as inferring the level of parallelism used within the targets, we combined the two detection tools from subsection 2.2.

1. The first one is a variant of the non-specific TVLA which we next denote as one-hot TVLA. We perform 16 well-chosen fixed-vs-fixed t-tests [12], such that the two sets of inputs induce a difference of a single byte in the first round of the AES encryption (hence the one-hot terminology). An independent t-test is then executed for each byte of the AES state. Overall, this requires only 17 sets of measurements as one of the sets is common across all the t-tests. An illustration of this method is given in Figure 1.

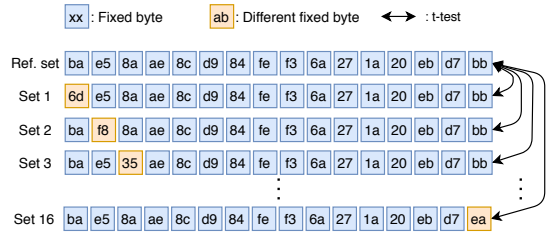


Fig. 1: Illustration of the one-hot TVLA.

2. Second, and as a complement, the SNR is evaluated for the first AES round (detected thanks to the first step), in order to obtain the POIs for each of the Sbox outputs, that are then considered for profiling.

Note that the one-hot TVLA lies between specific and non-specific t-tests. A non-specific t-test leaks everywhere on the leakage trace and is not suitable for POI detection. Non-specific t-tests can be fixed-vs-random as in [14,9] or fixed-vs-fixed as in [12]. They typically allow faster leakage detection thanks to their reduced number of classes. By contrast, specific t-tests (like SNR computations) allow POI detection at the cost of a higher number of classes to estimate. Due to the structure of the one-hot TVLA, leakage is detected only for the single byte which is different during the first operations of a block cipher. Then, for the later rounds, leakage is spotted everywhere due to the diffusion property. As a result, the one-hot TVLA is more specific for the first AES round and non-specific for the later rounds. By comparing the position of significant TVLA peaks for each byte, we can deduce the position of the first round of the AES.

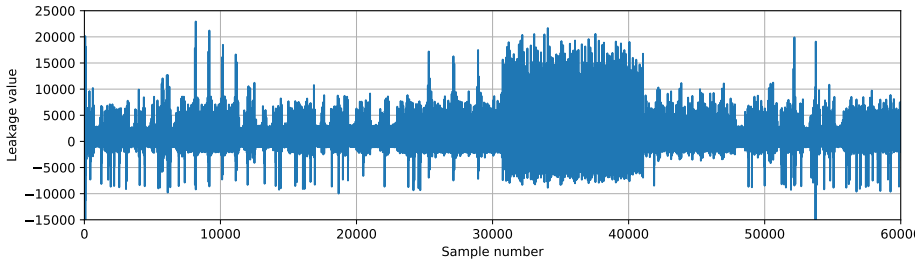
Note also that such an intermediate between non-specific and specific tests was already proposed. For example, the semi-fixed vs. random test in [9] combines a random set and one set with "well-chosen" values for similar reasons. Yet, the one-hot approach has two advantages compared to this previous proposal. First, the semi-fixed vs. random test will become specific as the size of the semi-fixed test increases (while the one-hot TVLA works with two classes per target byte, which can reduce its data complexity). Second, the semi-fixed vs. random test works as long as the model assumptions used to select the "well-chosen" values are correct (while such good model assumptions may not be available at this stage of an evaluation and are anyway not desirable for detection).

The combination of the two proposed steps can be an interesting tradeoff for evaluators. While a good part of the most informative points' positions can be identified with the one-hot TVLA, it remains that the corresponding peaks do not have the quantitative meaning that the SNR carries, as for example discussed in [12]. Besides, while the first step could directly be based on the SNR, the TVLA has the advantage of requiring a small data and time complexity to be evaluated [22]. Since the SNR is not computed on the whole trace but only on the first round, it reduces the time/memory complexity required for the detection. For example, when performed on traces of the same size, the one-hot TVLA requires about 100 times less memory than the SNR computed for 256 classes.

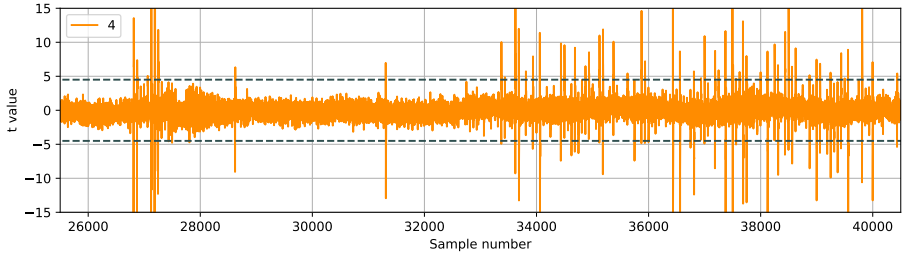
4.2 NXP Kinetis

The results of the methodology presented above are shown in Figure 2 with the mean trace on Figure 2a. We observe that a greater signal is present from samples 30,000 to 40,000. This is equivalent to 160 MCU cycles, leading to the suggestion of an implementation serialized on 1 Sbox. This will be verified with measurements as NXP does not provide the cycle count for this AES core.

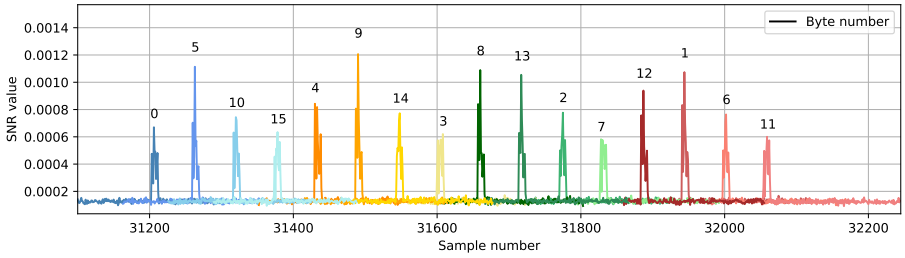
We observe that no leakage is spotted with the one-hot TVLA up to sample 25,000. This first part of the trace corresponds to the key scheduling of the AES (which we confirmed by using a set of inputs with different keys). More interestingly, three distinctly leaking parts can then be highlighted, confirming the interest of the one-hot TVLA: first, the loading of the plaintext around



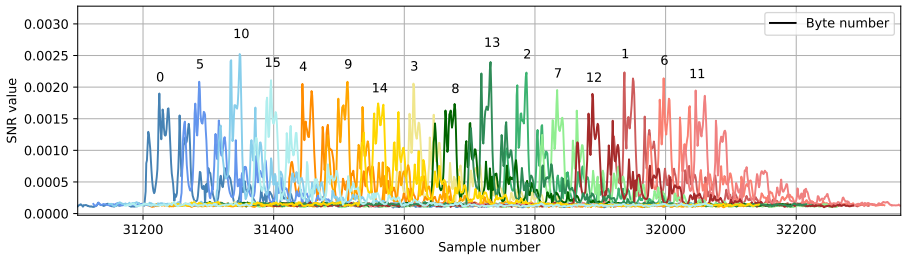
(a) Mean power leakage trace.



(b) Exemplary t-test on EM traces for byte 4.



(c) SNR on EM traces. Each peak is annotated with its corresponding byte number.



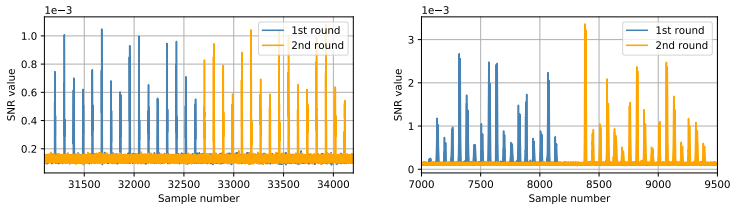
(d) SNR on power traces. Each peak is annotated with its corresponding byte number.

Fig. 2: Architecture inference methodology on the NXP target (low clock freq.).

sample 27,000; second, the key addition and the Sbox execution (here given for an exemplary byte) which correspond to the two peaks between samples 28,000 and 32,000; finally, all the peaks after sample 33,000 which is where the test starts to be non-specific due to the diffusion happening after the first AES round.

We next estimated the SNR of the 16 AES Sboxes for all the samples corresponding to the first AES round identified with the one-hot TVLA. The results of Figure 2c and Figure 2d show that each Sbox leads to well identified peaks and these peaks are spaced by a single cycle. Information about all the bytes is therefore available. By computing the SNR on the second round Sbox, we finally observe on Figure 3a that no cycles are lost between rounds. The MixColumns and key addition operations are therefore interleaved with the Sboxes.

As a result, the architecture is inferred to be serialized on 8 bits for the Sboxes and 32 bits for MixColumns (possibly performed in parallel to the Sboxes).



(a) NXP (b) STM

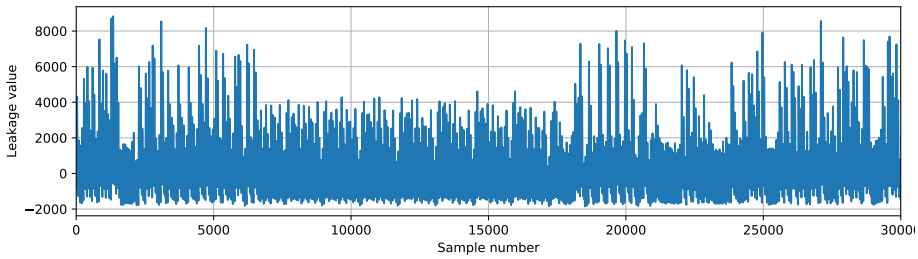
Fig. 3: SNR on first and second rounds for both devices

4.3 STM32

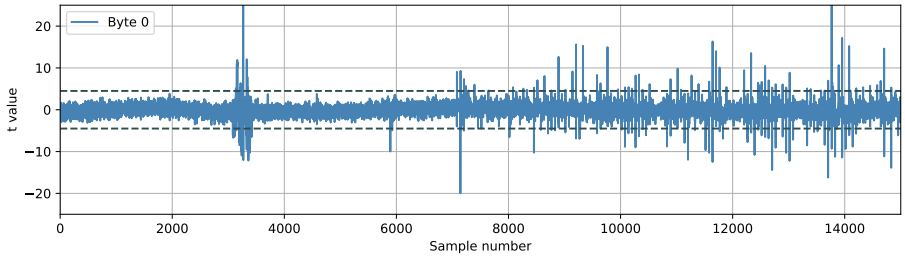
The same methodology was applied to the STM target and is reported in Figure 4. The mean power trace is given in Figure 4a. A repeating pattern is present and its length corresponds to the 214 MCU cycles presented in the reference manual [26].[2] This again suggests an implementation serialized on 1 Sbox. The application of the one-hot TVLA in Figure 4b leads to similar intuitions as for the NXP target and we can identify the plaintext loading (around sample 3500), the key addition and S-boxes (around samples 6000 and 7000) and the non-specific leakages after the first AES round (starting after sample 8,200).

These preliminary results are then confirmed with the SNR estimations of Figure 4c for EM measurements and Figure 4d for power ones. Compared to the NXP target, we notice a more significant disparity between the peak SNR levels of the different bytes (this difference is observed for both power and EM leakages). The largest SNRs on each byte are spaced by exactly one cycle. The SNR on a byte is also significant (but smaller) one cycle after the first peak (except for bytes 10 and 15 with power leakage). Eventually, by computing the SNR on the second round, we deduce that the first round and the second one are spaced by 5 cycles, as represented on Figure 3b.

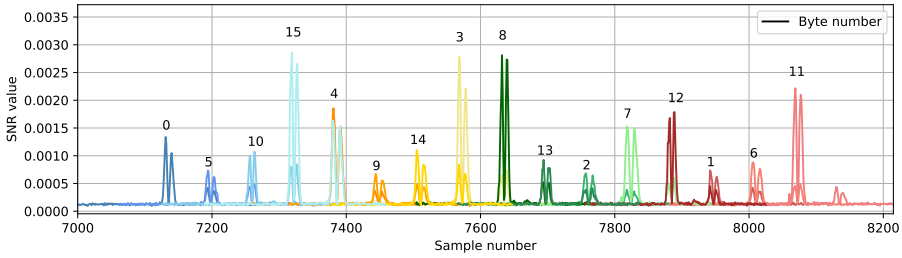
[2] This number corresponds to the AES core and excludes data loadings.



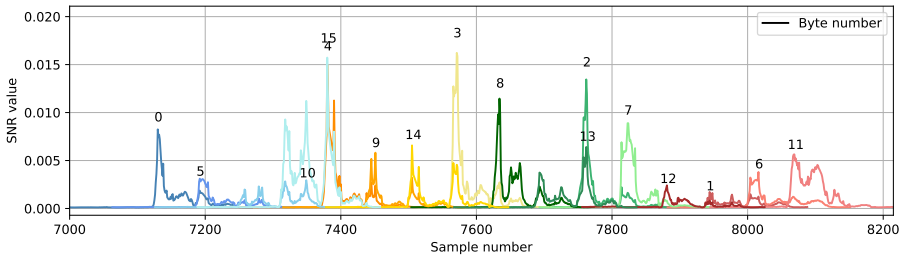
(a) Mean power leakage trace.



(b) Exemplary t-test on EM traces for byte 0.



(c) SNR on EM traces. Each peak is annotated with its corresponding byte number.



(d) SNR on power traces. Each peak is annotated with its corresponding byte number.

Fig. 4: Architecture inference methodology on the STM target (low clock freq.).

Putting things together, the architecture can be inferred to be serialized on 8 bits for the Sbox layer. The MixColumns operation and key addition are serialized on 32 bits and are not interleaved with Sboxes. Such an architecture might also leak information through the distance between two consecutive Sboxes. We tried to exploit such leakages but did not obtained better SNRs.

Overall, by comparing the STM and the NXP targets, we can conclude that the two manufacturers propose quite similar architectures.

5 Attacks

In order to evaluate the side-channel security provided by the two targets, we performed key recovery with CPA with various leakage models and TA. Next, we first describe our attack strategy and then discuss attack results.

5.1 Attack strategy

The different steps of the presented attacks we performed are illustrated in Figure 5. All these steps (including the pre-processing) are performed independently on the key bytes. More precisely, these steps are described as follows.

- First, we use the results of the architecture inference from section 4 to identify the most significant SNR peaks. For each attack, we reduce the trace length by two cycles before and four cycles after this peak.
- Second, the CWT (see Equation (6)) is optionally applied to the shorter traces. When applied, the next steps are performed in the wavelet domain.
- If applicable, and in order to identify the POIs in the wavelet domain, we compute again the SNR on the previously obtained signal.
- In all cases, we only keep the points that have significant SNR values. This is done by filtering all the points with SNR smaller than at $2 \cdot 10^{-4}$.
- Eventually, the two attacks from subsection 2.3 are performed on the pre-processed traces. The CPA is directly applied. For the TA, a PCA in additionally performed in order to reduce the signal to five dimensions. Five dimensions were chosen as a compromise between the computational efficiency of the resulting attack and the amount of information extracted.

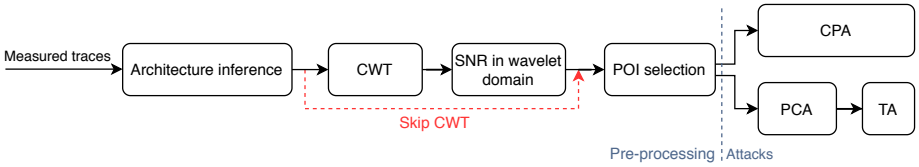


Fig. 5: General attack strategy.

The above attacks all use a profiled leakage model. For the CPA, it corresponds to the mean leakage of each output value of the Sboxes. CPAs with non profiled leakage models were also performed and are discussed below.

5.2 Attacks results and discussions

We now compare different attacks against our two targets for both low and high clock frequencies and for both EM and power traces. In all cases, we used $2 \cdot 10^6$ profiling traces. The results of these attacks are summarized in Table 1. It contains the number of traces we need to reduce the rank of the 128-bit master key below 2^{32}, using the rank estimation algorithm of [20].[3] As a complement, we also provide the graphs reporting the evolution of the guessing entropy for the best attack against each target in the full version of this paper.

We next discuss the influence of some parameters by looking at the minimum number of traces to reach a key rank lower than 2^{32}. We start with the targets' parameters and measurement setup, and follow with the attack strategy.

Device		2^{32} key rank		Full key recovery	
		CPA	TA	CPA	TA
NXP 8MHz	EM	27800	5200	N.A.	27400
	EM with CWT	3800	2400	7400	8800
	Power	6400	2000	15600	6200
	Power with CWT	12600	1200	27600	5400
NXP 100MHz	EM	26000	N.A.	N.A.	N.A.
	EM with CWT	4200	4400	11000	26600
	Power	N.A.	N.A.	N.A.	N.A.
	Power with CWT	10000	N.A.	N.A.	N.A.
STM32 8MHz	EM	16800	N.A.	N.A.	N.A.
	EM with CWT	3800	N.A.	11000	N.A.
	Power	4200	N.A.	17400	N.A.
	Power with CWT	1800	38000	5400	N.A.
STM32 80MHz	EM	30000	N.A.	N.A.	N.A.
	EM with CWT	2200	N.A.	7200	N.A.
	Power	1600	N.A.	7400	N.A.
	Power with CWT	1600	N.A.	6200	N.A.

Table 1: Number of traces needed for every attack. The maximum number of traces was set to 40k traces. The best attack for each target is highlighted in a different color. N.A. signifies the need for more than 40k traces to succeed.

The **clock frequency** has a limited influence: for the NXP target, the best attack requires 3.5 more traces at high frequency than at low frequency; for the STM one, the best attacks are nearly equivalent at low and high frequencies. This is most likely due to the fact that these frequencies remain moderate and do not lead to particular challenges in terms of sampling frequencies.

[3] For CPA attacks, as they do not output probabilities but correlation scores, the rank estimation algorithm is not optimal and may not represent the worst-case [8]. We then use it as a heuristic bound on the security level of our targets.

The **power and EM** measurements do not differ strongly either (in terms of best attack complexity): both channels can lead to powerful key recoveries. Yet, the attack exploiting power leakage is roughly twice faster than the one based on EM traces in most of the cases. EM is only the best for the NXP target at high frequency, where it requires about 2.5 less traces than its power counterpart.

By contrast, **TA and the CPA** do not always lead to similar results. In most cases, the best attack is a CPA. This can be explained by the limited number of traces we used for profiling. That is, the CPA is a univariate attack and only requires estimating mean values. The TA is a multivariate attack (remember we kept 5 dimensions after PCA) and in most cases, exploiting these additional dimensions is not useful given our profiling effort. Interestingly, the only case where TA outperform the CPA is with the power traces of the NXP target. This quite nicely matches the intuition of Figure 2d where we see that the signal is less sparse in this case. So for this target, the additional information of the additional dimensions is sufficient to compensate a more expensive profiling.

Note that with more profiling efforts, TA should at least reach the level of performance of CPA [17]). In view of the low complexity of the simple attacks we put forward in Table 1, we did not look for such further optimizations.

The **CWT** pre-processing is almost always improving attack complexity. The disparities observed between the different leakages are indeed reduced thanks to the CWT. We also note that the gain of the CWT in the EM case is of a factor between 4 and 10, while it never exceeds 2 in the power case. We assume this difference is due to the richer frequency content of the EM signals.

Although partially successful, our attacks without profiled leakage models were much worse. A combination of Hamming weight and Hamming distance models was necessary to recover the key on the STM target. Yet, the corresponding attack required 40,000 traces. As for the NXP target, a profiled model was necessary with our measurements and we were not able to recover any byte of the key with 75,000 traces. This suggest that both accelerators have leakages that are only weakly correlated to a Hamming Weight predictions.

Finally, we note that the **NXP core countermeasure** was activated. Namely, all the experiments against the NXP target reported in this section were performed while reseeding the `DPA Mask Seed` register after 50,000 encryptions, as stated in the reference manual. This setting was compared with two other ones where we either did not reseed the register or we reseeded it with the same value before each encryption in order to cancel the randomization. While slightly modifying the shape of the measurements, these variations did not appear to imply noticeable variations of the corresponding security levels.[4]

6 Conclusions

Our results first exhibit the limited security against side-channel attacks that unprotected hardware co-processors in mid-range 32-bit MCUs provide. Those co-processors should therefore be viewed as performance improvers, not as a

[4] We are aware of two independent teams who observed similar results.

direct solution for securing an embedded implementation. They also highlight that a lack of precise understanding of the target architectures does not prevent the simple identification of target intermediate variables in an implementation (though more optimized attacks could certainly be designed with a better understanding of the targets). The "one-hot" variation of the TVLA we propose is quite handy for this purpose. A similar statement holds for the CWT which seems to be an interesting addition in the side-channel analysis toolbox, as is quite systematically improved the attack results in our investigations. As an interesting direction for further research we mention the combination of hardware co-processors such as analyzed in this work with leakage-resistant modes of operation such as [4,3,27], as recently surveyed in [2].

Acknowledgments. We thank Colin O'Flynn for useful feedback about the paper. François-Xavier Standaert is a Senior Research Associate of the Belgian Fund for Scientific Research (FNRS-F.R.S.). Work funded in parts by the European Union through the ERC project SWORD (724725) and the European Union & Walloon Region FEDER USERMedia project 501907379156.

References

1. Archambeau, C., Peeters, E., Standaert, F., Quisquater, J.: Template attacks in principal subspaces. In: CHES. LNCS, vol. 4249, pp. 1–14. Springer (2006)
2. Bellizia, D., Bronchain, O., Cassiers, G., Grosso, V., Guo, C., Momin, C., Pereira, O., Peters, T., Standaert, F.: Mode-level vs. implementation-level physical security in symmetric cryptography - A practical guide through the leakage-resistance jungle. In: CRYPTO (1). LNCS, vol. 12170, pp. 369–400. Springer (2020)
3. Berti, F., Guo, C., Pereira, O., Peters, T., Standaert, F.: Tedt, a leakage-resist AEAD mode for high physical security applications. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2020**(1), 256–320 (2020)
4. Berti, F., Pereira, O., Peters, T., Standaert, F.: On leakage-resilient authenticated encryption with decryption leakages. IACR Trans. Symmetric Cryptol. **2017**(3), 271–293 (2017)
5. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: CHES. LNCS, vol. 3156, pp. 16–29. Springer (2004)
6. Bronchain, O., Standaert, F.: Side-channel countermeasures' dissection and the limits of closed source security evaluations. vol. 2020, pp. 1–25 (2020)
7. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: CHES. LNCS, vol. 2523, pp. 13–28. Springer (2002)
8. Choudary, M.O., Poussier, R., Standaert, F.: Score-based vs. probability-based enumeration - A cautionary note. In: INDOCRYPT. LNCS, vol. 10095, pp. 137–152 (2016)
9. Cooper, J., Mulder, E.D., Goodwill, G., Jaffe, J., Kenworthy, G., Rohatgi, P.: Test vector leakage assessment (TVLA) methodology in practice. ICMC 2013
10. Debande, N., Souissi, Y., Elaabid, M.A., Guilley, S., Danger, J.: Wavelet transform based pre-processing for side channel analysis. In: MICRO Workshops. pp. 32–38. IEEE Computer Society (2012)
11. Dinu, D., Kizhvatov, I.: EM analysis in the iot context: Lessons learned from an attack on thread. In: IACR Trans. Cryptogr. Hardw. Embed. Syst. vol. 2018, pp. 73–97 (2018)

12. Durvaux, F., Standaert, F.: From improved leakage detection to the detection of points of interests in leakage traces. In: EUROCRYPT (1). LNCS, vol. 9665, pp. 240–262. Springer (2016)
13. Eisenbarth, T., Kasper, T., Moradi, A., Paar, C., Salmasizadeh, M., Shalmani, M.T.M.: On the power of power analysis in the real world: A complete break of the keeloqcode hopping scheme. In: CRYPTO. LNCS, vol. 5157, pp. 203–220. Springer (2008)
14. Goodwill, G., Jun, B., Jaffe, J., Rohatgi, P.: A testing methodology for side channel resistance validation. NIST non-invasive attack testing workshop (2011)
15. Liu, J., Yu, Y., Standaert, F., Guo, Z., Gu, D., Sun, W., Ge, Y., Xie, X.: Small tweaks do not help: Differential power analysis of MILENAGE implementations in 3g/4g USIM cards. In: ESORICS (1). LNCS, vol. 9326, pp. 468–480. Springer (2015)
16. Mangard, S.: Hardware countermeasures against DPA ? A statistical analysis of their effectiveness. In: CT-RSA. LNCS, vol. 2964, pp. 222–235. Springer (2004)
17. Mangard, S., Oswald, E., Standaert, F.: One for all - all for one: unifying standard differential power analysis attacks. vol. 5, pp. 100–110 (2011)
18. Moradi, A., Kasper, M., Paar, C.: Black-box side-channel attacks highlight the importance of countermeasures - an analysis of the xilinx virtex-4 and virtex-5 bitstream encryption. In: CT-RSA. LNCS, vol. 7178, pp. 1–18. Springer (2012)
19. Moradi, A., Schneider, T.: Improved side-channel analysis attacks on xilinx bitstream encryption of 5, 6, and 7 series. In: COSADE. LNCS, vol. 9689, pp. 71–87. Springer (2016)
20. Poussier, R., Standaert, F., Grosso, V.: Simple key enumeration (and rank estimation) using histograms: An integrated approach. In: CHES. LNCS, vol. 9813, pp. 61–81. Springer (2016)
21. Ronen, E., Shamir, A., Weingarten, A., O'Flynn, C.: Iot goes nuclear: Creating a zigbee chain reaction. In: IEEE Symposium on Security and Privacy. pp. 195–212. IEEE Computer Society (2017)
22. Schneider, T., Moradi, A.: Leakage assessment methodology - extended version. J. Cryptographic Engineering **6**(2), 85–99 (2016)
23. Semiconductors, N.: Kinetis k82 datasheet (2015), `https://www.nxp.com/docs/en/data-sheet/K82P121M150SF5.pdf`
24. Semiconductors, N.: Kinetis k82 reference manual (2015), `https://www.nxp.com/docs/en/reference-manual/K82P121M150SF5RM.pdf`
25. ST: Stm32l422cb datasheet (2018), `https://www.st.com/resource/en/datasheet/stm32l422cb.pdf`
26. ST: Stm32l422cb reference manual (2018), `https://www.st.com/resource/en/reference_manual/dm00151940-stm32l41xxx42xxx43xxx44xxx45xxx46xxx-advanced-armbased-32bit-mcus-stmicroelectronics.pdf`
27. Unterstein, F., Schink, M., Schamberger, T., Tebelmann, L., Ilg, M., Heyszl, J.: Retrofitting leakage resilient authenticated encryption to microcontrollers. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2020**(4), 365–388 (2020)
28. Welch, B.L.: The generalization of 'student's' problem when several different population varlances are involved. vol. 34, pp. 28–35 (1947)
29. Zhou, Y., Yu, Y., Standaert, F., Quisquater, J.: On the need of physical security for small embedded devices: A case study with COMP128-1 implementations in SIM cards. In: Financial Cryptography. LNCS, vol. 7859, pp. 230–238. Springer (2013)