REASSURE

Project ID: 731591, Horizon 2020
http://www.reassure.eu

REASSURE

Deliverable D1.5

White Paper: Assurance in Security Evaluations

Editor:	F.X. Standaert (UCL)
Deliverable nature:	R
Dissemination level: (Confidentiality)	PU
Delivery date:	March 31st, 2020
Version:	1.0
Total number of pages:	21
Keywords:	Assurance, Security Evaluations, CC, FIPS 140, REASSURE

Horizon 2020
European Union funding
for Research & Innovation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 731591.

Executive summary

Security evaluations are complex, and two distinct approaches exist today: conformance style testing as in FIPS 140 and attack driven testing as in CC. Within the REASSURE project we studied attack vectors with regards to their optimality and potential for automation with the aim to improve existing evaluation regimes. By optimality we mean that we can prove that any practical instantiation of an attack reaches its theoretical limit; by potential for automation we mean that it can be executed with minimal user interaction. We comment on which steps offer some potential for automation in this white paper.

Considering **conformance style testing as an evaluation methodology**, it is clear that it **cannot offer any guarantees regarding optimality**: leakage detection in a black box setting is extremely challenging to set up correctly. **We have found that the current standard ISO 17825 requires improvement**, and this white paper provides some information regarding more sensible parameter choices.

Our research points towards the fact that any optimality can only ever be achieved when considering worst-case adversaries. These are adversaries that get full access to implementation details, can select secret parameters, and thereby control countermeasures during an initial profiling phase. The reason for this is that it is only in this setting that we can utilise statistical (or machine/deep learning) tools which are well understood and for which we can assess/argue their optimality. Any attack vector which requires dealing with higher order or multivariate data leads to a loss of theoretical guarantees.

Within the **REASSURE proposal for a so-called "backwards" approach** for evaluations, we postulate that any evaluation should attempt first to instantiate a worst-case adversary (even if this requires open samples and/or samples with known secrets, or even the developers in their own environment to execute these attacks). In this setting we argue that we have the strongest guarantees for optimality. If necessary an evaluation should then also instantiate a more "practical" attack by relaxing some assumptions. **The difference in effort required between the worst case and the best practical attack then gives an indication of the (potential) security gap.**

We recommend that any reporting that is based on a points-based system like [46] should **make the ratings explicit for the worst-case adversary** (separating out the points for identification/profiling and the exploitation) **and the "best practical adversary"** (again making explicit how points are awarded for the different phases), and it should also be reported how likely it is that the results from identification/profiling translate to other devices.

Finally we observe that the role of formal verification in the context of side channel evaluations is perhaps different from the role that it plays in other contexts. Formal methods prove properties of implementations on an abstract level: this requires assumption of a leakage model. In practice however, devices show multiple, context-dependent leakage characteristics. Therefore **formal verification in the context of leakage evaluations only shows that the necessary conditions are fulfilled, but they do not provide evidence about the sufficiency of these conditions.** These sufficient conditions can only be ascertained via testing. **Consequently, formal verification is suitable for low assurance levels.** By contrast, high assurance essentially relates to the risk that (e.g., backwards) security evaluations, even when the evaluator is provided with worst-case capabilities, may be sub-optimal due to inherently heuristic and hard to analyze theoretically attack steps.

List of authors

Company	Author
UCL	D. Bellizia
UNI-KLU	E. Oswald
UCL	F.X. Standaert
UNIVBRIS	C. Whitnall

Other contributions gratefully received from

Company	Author
IDM	N. Debande
Riscure	I. Buhan
SGDSN	E. Jaulmes

Contents

1 Introduction: bounding the unknown

Testing for side channel vulnerabilities is a central aspect of security evaluations of implementations featuring cryptography. The effort that goes into testing is considerable, and the stakes for companies are high.

Stakeholder groups such as JHAS are concerned with achieving a balance between sound evaluation practices and the cost of evaluations. Their approach is to discuss and in some sense categorise attacks (they maintain a list of attack vectors that need to be attempted during an evaluation), and come to a shared understanding of the difficulty of mounting attacks via a specific rating system [46]. EMVCO, a stakeholder group alike in its composition to JHAS, proceeds in a similar fashion. Both groups operate predominantly in the Common Criteria (CC) framework.

In contrast, the FIPS 140 approach is to keep the cost of evaluation to an absolute minimum, and to mandate no more than conformance style testing as specified in ISO 17825:2016. FIPS 140-3 (which has been agreed on in 2019 and will become effective later in 2020) adopts a variation of the so-called Test Vector Leakage Assessment (TVLA) framework to gauge the threat of side channel attacks.

In this white paper we are concerned with the central question of what is the assurance[1] that such security evaluations provide. By assurance we mean the confidence that we can have in the true security level of a product.

Concretely, answering this question requires considering the optimality of the general attack strategy selected by the evaluator, and the various steps to instantiate it. For this purpose, we draw on the REASSURE work published as Deliverable D1.3. (`www.reassure.eu`) and discuss whether state-of-the-art solutions for the different steps of a side-channel security evaluation offer bounds or guarantees of optimality, or if they are inherently heuristic.

We use this discussion to provide an informal rating of the steps' optimality and to put forward where risks of overstated security levels remain. We additionally discuss the role of formal verification, which is required to reach high certification levels in the Common Criteria framework [51].

We observe that formal verification in its current use in the side channel community (see e.g. [3]) works best when analysing abstract implementations. We therefore suggest that abstract formal verification is better used as a first step to consider even for low security levels, as it may reveal implementation flaws in the countermeasures implemented early in their design process. By contrast, high assurance is better obtained by increasing the confidence that the general evaluation strategy and each step used to attack an implementation are close enough to optimal, while explicitly stating where irreducible sources of risk lie, as discussed next.

1.1 Organisation and Outline of this White Paper

We provide a brief explanation of the two prevalent evaluation schemes (Common Criteria and FIPS 140) in Section 2, followed by a succinct summary of the evaluation approach proposed by REASSURE in Section 3. Then we consider the optimality of the steps or components that are the constituent parts of the three evaluation approaches and comment on the overall assurance that contemporary evaluations offer in Section 4.

2 State-of-the-Art Evaluation Approaches

Whilst there are a number of security evaluation approaches possible, two schemes (and derivatives thereof) dominate in practice. Common Criteria evaluations are "attack driven" and aim to systematically capture and categorise attack vectors. FIPS 140 evaluations are "conformance style" evaluations that rely on checking

[1] CC uses the term "Assurance level" in its terminology, and attributes a specific meaning to each assurance level. We will make it explicit in our text if we mean such an assurance level.

some minimum criteria.

FIPS 140-2 is mandated in the US (FIPS 140-3 will replace FIPS 140-2 late in 2020), and it is also mandated in Canada. Some other countries (e.g. Japan), have begun adopting it as well. FIPS 140 is represented by a set of ISO standards (ISO/IEC 19790:2012(E) and ISO/IEC 24759:2017(E)), and the difference between FIPS 140-2 and FIPS 140-3 is the inclusion of testing against side-channel attacks (the methodology for this is given in ISO/IEC 17825:2016, with setups and calibration defined in ISO/IEC 20085-1 and 20085-2).

Common Criteria is an international standard. It is represented by ISO 15408. Common criteria features a range of assurance levels (so called EALs), and to reach the higher level requires more rigorous testing. In particular, from EAL 4 onwards, some testing against side-channel attacks are required.

Both programs require that the product is tested by an accredited testing laboratory and a government agency oversees this process.

2.1 CC

CC evaluations are complex and governed by a range of specifications. The product which is being certified is called the Target of Evaluation (TOE). For a TOE three documents are of relevance: the Protection Profile, the Security Target, and the Security Functional Requirements.

The Protection Profile is often created by a user community and provides an implementation independent specification of security requirements for a "class of devices": it lists threats, security objectives, assumptions, security functional requirements (SFRs), security assurance requirements (SARs) and rationales.

The Security Target (ST) details the secure implementation of the TOE. Some manufacturers use a corresponding Protection Profile as a reference when manufacturing a particular type of device. Also, the Protection Profile may feed into the Security Target used for certification. The Security Target also represents the exact configuration for the certified environment. Vendors often make the Security Target details available to their customers.

Security Functional Requirements are the functions that a product will provide. Common Criteria provides a list of standard functions that products can provide.

During the Common Criteria evaluation process, vendors must state an envisioned security level. This is called the Evaluation Assurance Level (EAL). The EAL is supposed to indicate the rigor of the evaluation.

There are seven levels, with EAL 1 being the most basic and level 7 being the most rigorous:

- EAL 1: Functionally tested
- EAL 2: Structurally tested
- EAL 3: Methodically tested and checked
- EAL 4: Methodically designed, tested, and reviewed
- EAL 5: Semiformally designed, and tested
- EAL 6: Semiformally verified, designed, and tested
- EAL 7: Formally verified, designed, and tested

It is possible to pick one level and "augment" it with specific requirements from a higher level.

It is imperative to understand that higher EALs do not imply a higher level of security, they imply that the claimed security assurance of the TOE has been more rigorously verified.

2.2 ISCI – JHAS

In the specific case of smart cards, the International Security Certification Initiative (ISCI) brings together stakeholders from every aspect of smart card security evaluation: certification bodies, evaluation laboratories, hardware vendors, software vendors, card vendors and service providers.

ISCI has two working groups: ISCI-WG1, which aims to define methodology and best practice for smart security device evaluation, and ISCI-WG2 (also known as JHAS), which defines and maintains the state of the art in potential attacks against smart security devices.

Two documents are essential for the evaluation of smart cards. The "Application of Attack Potential to Smart Cards" [46] provides a "rating system" for attacks. The "Attack Methods for Smart Cards and Similar Devices" [47] is a confidential document and describes attack vectors that are considered "relevant".

The purpose of the rating system is grounded in the need to be able to compare the "security strength" of different products. The rating system is designed to reduce subjectivity and it results in a total score. This score is the sum of several factors during both the "Identification" and the "Exploitation" phase of an attack (for reference: identification is broadly speaking about finding, and characterising, leaks; exploitation refers to the actual attack). The factors that are considered are: Elapsed time, Expertise, Knowledge of TOE, Access to TOE, Used equipment, Open samples[2].

2.3 FIPS 140-3

This Federal Information Processing Standard (140-2, and, from late 2020 on, FIPS 140-3) specifies the security requirements for cryptographic Modules. It has four increasing, qualitative levels intended to cover a wide range of potential applications and environments.

FIPS 140-3 covers side-channel attacks. Among the existing ISO standards, a side-channel test regime is given in ISO/IEC 17825:2016, with setups and calibration defined in ISO/IEC 20085-1 and 20085-2 (NIST special publications SP800-140 A-F may modify these in the future).

For testing against basic power analysis attacks, ISO/IEC 17825:2016 relies on using leakage detection procedures instead of attempting attacks. Leakage detection involves producing evidence for the presence of leaks using statistical hypothesis testing. It has been advertised as a "cheaper process" than running full blown attacks, and ISO/IEC 17825:2016 suggests it may be done *instead* of attacks.

ISO/IEC 17825:2016 adopts a modified version of the Test Vector Leakage Assessment (TVLA), which is a methodology to test side-channel resistance. As such, it is a black-box tool that gathers evidence for the presence of leaks.

3 The REASSURE Approach: Worst-Case Evaluation

In the REASSURE white paper on "Evaluation Strategies for AES and ECC", Azouaoui et al. suggest a backwards approach that advocates to structure any evaluation starting with the worst-case adversary.

Whilst the definition of a worst-case adversary is still somewhat implementation dependent, it is considerably more well defined and stable than selecting a (subjectively) best attack from a (changing) list of "relevant" attacks.

The worst-case adversary is assumed to be able to measure one or multiple side-channels from the target, and have full control over all inputs (plaintexts or ciphertexts), full control over the secret parameters (keys, randomness). He can turn off any countermeasures, and has detailed implementation knowledge (e.g., source code in

[2]For the sake of succinctness we refer the reader to the JHAS documentation for a precise definition of these factors[46].

the case of software implementations, or a hardware level description in the case of hardware implementations).

The REASSURE approach advocates that if at all possible, then an attack for the worst-case adversary should be demonstrated.

After the feasibility of a worst-case attack has been considered, and if there are sound reasons that explain why this may not be possible, then the REASSURE approach explains that adversarial assumptions or capabilities can be gradually relaxed, and attack be considered and demonstrated for the considered relaxed assumptions.

The impact of relaxing these strong adversarial capabilities on the attack complexity should be discussed, in order to assess the possible complexity gaps between worst-case attacks and ones with fewer assumptions.

Because every evaluation requires a number of (potentially iterative) steps, it is important to consider and spell out assumptions for each of the steps, which will ultimately determine the assurance of the evaluation.

3.1 Evaluation steps

Within REASSURE we also proposed to consider any evaluation as a composition (possibly iterative) of the following key steps:

1. **Measurement and preprocessing.** This step provides the adversary/evaluator with leakages (e.g., the power consumption or electromagnetic radiation of a chip, or their simulation in case simulated analyses are considered) based on his input control, and possibly performs data-independent preprocessing in order to improve the quality of these measurements.
2. **Leakage detection and mapping.** In leakage detection, the adversary/evaluator aims to detect the presence of any data-dependent leakage (independent of whether this data-dependency is exploitable in a realistic attack). Leakage mapping further aims to connect the detected samples to specific operations performed by the target implementation.
3. **Leakage exploitation.** In this last step, the adversary/evaluator aims to exploit the leakages in order to perform an attack (e.g., a key recovery). It is usually divided in three phases:
   (a) **(Optional) modelling phase.** In this phase, the adversary/evaluator takes advantage of his profiling abilities to estimate a key-dependent model for his leakages.
   (b) **Information extraction phase.** In this phase, the adversary/evaluator extracts information about intermediate values manipulated by his target implementation thanks to a model (that can be obtained from a profiling phase or assumed a priori).
   (c) **Information processing.** In this final phase, the adversary/evaluator combines the partial information he extracted from his target information and aggregates this information in order to recover some secret parameter (e.g., a master key).

4 Risk assessment & examples

We now discuss the optimality of the state-of-the-art tools that can be used for these attack steps.

4.1 Measurement and preprocessing

In general, a measurement setup is composed of several elements, such as a probe, preamplifiers, physical filters and a digital storage oscilloscope, that has to deploy some specific characteristics, such as low-noise capability, suitable bandwidth and sampling rate, as also reported in ISO/IEC 20085-1.

The choice of those components and how they interact with each other impact sensibly on the final outcome of the practical evaluation of a device. Standing on the knowledge of the device's operating parameters (e.g., clock frequency, range of admitted operating power supply voltage, etc.), the measurement setup has to be designed in order to fulfil the expected leakage characteristics in order to deploy a sound evaluation.

4.1.1 Setup Design

The first step in the setup design for a specific side-channel observation is the choice of the probe, which represents the interface between the target of evaluation and the measurement chain. Its role is to convert the physical observation into a voltage signal. The overall side-channel trace and its quality are, in general, strongly affected by the choice of the probe (e.g., for power analysis it can be a shunt resistor or an inductive probe, for electromagnetic analysis an E-field or H-field probe) and its physical characteristics (e.g., bandwidth, conversion gain, etc.), remarking here the necessity of engineering expertise.

Following components of the measurement setup, as preamplifiers and/or physical filters, can be considered as data-independent signal preprocessing/conditioning blocks. They are used to improve the quality of the measurements by, respectively, amplifying the signal and removing unwanted frequency contents to speed up the evaluation process, as also shown in [40]. As well as the probe, these components have to be chosen carefully to avoid degrading the quality of the signal and engineering expertise may be required.

The last component of the measurement setup is usually represented by a digital storage oscilloscope, having the role to digitise and quantize the side-channel signal into a digital representation. The particular choice of digital storage oscilloscope and its parameters, such as vertical resolution, voltage range, front-end analog bandwidth, impedance matching and sampling rate, are usually application specific, being dependent on the target of evaluation as well as previous blocks of the measurement chain. As a general consideration, the measurement setup design is hard to automate, due to the fact that most critical aspects are device and technology dependent.

4.1.2 Optimality of Measurements and Preprocessing

Due to its physical nature, the measurement and preprocessing attack step is inherently carrying hard to quantify risks. The quality of a measurement setup is indeed mostly dependent on hard to evaluate engineering expertise. It may lead to higher noise in the time and amplitude domains that directly affects the attacks complexity [32], and the impact of which exponentially increases whenever combined with countermeasures such as masking [17].

Preprocessing is similarly heuristic. Many published solutions exist to filter the noise [38, 40] and to resynchronise the traces [45, 52], but their effectiveness is typically application dependent. Based on this state-of-the-art, the best mitigation plan currently is to make measurement setups and preprocessing steps as open and reproducible as possible so that the quality of the measurements they provide can be compared thanks to simple and established metrics (e.g., the SNR for univariate evaluations [32, 23] and information theoretic metrics for multivariate evaluations [50]).

4.1.3 Potential for automation

Measurement is thus a process which entails considerable expertise as well as experimental trial-and-error. It is possible to envision elements of automation within the individual tasks:

- Peak recognition could be used to find interesting frequencies in spectra.
- Matching against pre-acquired (and labelled) pattern snippets could help identify interesting regions and suggest possible activity.
- An EM probe could be provided with candidate coordinates to search and programmed to target ‘promising’ regions for taking larger acquisitions, based on preliminary on-the-fly analysis (for example, high levels of clock frequency activity might be a valid indicator that one is ‘not too far’ from exploitable leakage).
- A rolling average could be fed back to the set-up to help automatically adjust the offset towards zero; on-the-fly histograms could similarly be used to find a suitable resolution.

However, all these improvements are expected to require application-specific configurations for which (expert) human supervision is currently required and no optimal choice can be mathematically demonstrated.

Besides, we also remark that quantifying the 'quality' of the measurement setup independent of the following steps is not possible. This is because whether or not the traces yield information about sensitive targets only becomes apparent once leakage detection or attacks have been attempted. In addition, the sample size (that is, the number of traces) required to detect the information present depends on features which are not apparent in the measurement stage and must be learned from further (or previous) analyses. If the acquisition proves unsuitable to the task (in either quality or quantity), a type of iterative backtracking becomes necessary.

By contrast, once effective setup parameters have been ascertained for a given implementation (and detection/attack task), it then becomes possible to automate repeated experiments via appropriate configuration scripts.

4.2 Detection and mapping

The term leakage in the context of leakage detection refers to the presence of sensitive data dependency in the trace measurements. Leakage can be detected using statistical hypothesis tests for independence. There are two potential end results aimed at by a detection test:

Certifying vulnerability: Find a leak in **at least one** trace point. In such a case it is important to control the number of false positives (that is, concluding there is a leak where there is none).

Certifying security: Find **no leaks** having tested thoroughly. Here false negatives (failure to find leaks that are really there) become a concern.

The statistical methods used for leakage detection cannot "prove" that there is no effect, they can at best conclude that there is evidence of a leak or that there is no evidence of a leak. Hence it is especially important to design tests with **'statistical power'** in mind – that is, to make sure the sample size is large enough to detect a present effect of a certain size with reasonable probability. Then, in the event that no leak is discovered, these constructed features of the test form the basis of a reasoned interpretation. A further, considerable challenge implicit to this goal is the necessity to be convincingly exhaustive in the range of tests performed – that is, to target "all possible" intermediates and all relevant higher-order combinations of points. (This suggests analogues with the idea of *coverage* in software testing).

Typically leakage detection is a pre-curser to leakage exploitation. However in conformance style testing as detailed in ISO 17825:2016, leakage detection is seen as a replacement for leakage attacks in specific circumstances (in particular in the case of testing block ciphers against standard DPA attacks).

We therefore consider the case of an evaluation with detection as precursor to attack, and the case of an evaluation that uses detection only.

4.2.1 Detect and Then Attack

It is *impossible to eliminate* errors in statistical hypothesis testing; the aim is rather to understand and minimise them. The decision to reject a null hypothesis when it is in fact true is called a Type I error, a.k.a. 'false positive' (e.g. finding leakage when in fact there is none). The acceptable rate of false positives is explicitly set by the analyst at a significance level α. A Type II error, a.k.a. 'false negative' is a failure to reject the null hypothesis when it is in fact false (e.g. failing to find leakage when in reality there is some). The Type II error rate of an hypothesis test is denoted β and the **power** of the test is $1 - \beta$, that is, the probability of correctly rejecting a false null in favour of a true alternative.
For the simple case of a t-test with equal sample sizes and population variances σ_1 and σ_2[3], the following formula can be derived:

[3]We consider these conditions to approximately hold in the case of leakage detection, where the partitions are determined by uniformly distributed intermediates.

$$N = 2 \cdot \frac{(z_{\alpha/2} + z_{\beta})^2 \cdot ({\sigma_1}^2 + {\sigma_2}^2)}{(\mu_1 - \mu_2)^2} \quad (1)$$

where $\mu_1 - \mu_2$ is the true difference in means between the two populations (this relationship can be found in any standard statistics textbook). Note that Eq. (1) can be straightforwardly rearranged to alternatively compute any of the significance level, effect size or power in terms of the other three quantities.
Equation (1) implies that the two errors can be traded-off against one another, and mitigated (but not eliminated) by:

- Increasing the **sample size** N, intuitively resulting in more evidence from which to draw a conclusion.
- Increasing the minimum **effect size** of interest $\mu_1 - \mu_2$, which in our case implies increasing the magnitude of leakage that one would be willing to dismiss as 'negligible'. This is possible via an improved setup.
- Choosing a different statistical test that is more efficient with respect to the sample size. In the case of first order leakage analysis, the t-test is already the most trace efficient technique.

If detection is followed by attacks, then the purpose of detection is in line with "certifying vulnerability": i.e. we want to find any leaks and are particularly interested to avoid false positives. Recall that false positives are trace points that indicate a leak but there is none. If attacks are based on false positives, they are likely to be inconclusive, and they waste evaluators' time.

Controlling false positives in the context of leakage traces (which have many potentially correlated leakage points) is all but straightforward. The principal difficulty is that for any methods that are not detrimental to the detection power, something has to be already known about the distribution of leaks in the leakage traces. This obviously represents a catch-22 if detection precedes further analysis.

However in the case were attacks follow detection, the consequences of missing out on some leaks (because of a lack of statistical power) is not as severe (as a lower powered test is still o.k.), because any detected leak that is confirmed via an attack leads to the rejection of the security claim about the device.

4.2.2 Detect and Then Stop: ISO 17825

The goal of evaluations typically is to "certify security", and if this is based on leakage detection only as in the case of ISO 17825, this is particularly difficult to achieve. In this case we cannot tolerate low powered tests as any missed leak may enable a device to pass certification.

As explained before, the confidence level of a test (i.e. the α), the power of a test (i.e. the $1 - \beta$), the number of traces N, the effect size $\mu_1 - \mu_2$ and the trace variance all play off each other in Equation (1). Setup manipulations may enable to increase $\mu_1 - \mu_2$ and decrease the trace variance, and an increase of the number of traces enables to achieve better confidence and power simultaneously. Consequently the trace "budgets" are very important factors in an evaluation that relies exclusively on leakage detection.

In ISO 17825:2016, the security levels 3 and 4 are separated by the resources (sample size = number of traces) available to perform the leakage detection, and the degree of data pre-processing. For level 3 10.000 traces are mandated; for level 4 100.000 traces are mandated. These criteria seem to be directly inherited from FIPS 140-2, which originally was based on attacks (like CC and EMVCo evaluations).

The standard leaves ambiguous whether the sample size specifications apply per acquisition or for both fixed and random trace sets combined; similarly whether they are intended per repetition or for both the first and the confirmatory analysis combined. We will see in the next section on optimality that the resources specified in ISO 17825 are inadequate.

4.2.3 Optimality of Detection and Mapping.

We studied methods to account for multiple testing and concluded that utilising the Bonferroni adjustment represents the best method to retain both detection power and deal with long traces [55].

Table 1 details how to select parameters for reliable leakage detection. The table details several scenarios that we consider relevant for evaluations. For each scenario we specify an indicative trace length, and an assumption about the number of leaks (many in software, few in hardware) in the traces. Then, for two different effect sizes (again chosen with practice in mind), and two difference confidence levels alpha, we report the number of measurements N to achieve high power.

Scenario type	**Trace length**	**# leaks**	**ISO** $\alpha = 0.05$		**TVLA** $\alpha = 0.00001$	
			$d = 0.04$	$d = 0.001$	$d = 0.04$	$d = 0.001$
Software (generic leaks)	100,000	100	2.5×10^4	3.9×10^7	6.8×10^4	1.1×10^8
Software (specific leaks)	100,000	10	4.8×10^4	7.7×10^7	1.2×10^5	1.9×10^8
Software (protected)	100,000	1	1.1×10^5	1.8×10^8	2.9×10^5	4.6×10^8
Hardware (unprotected)	1,000	10	2.9×10^4	4.6×10^7	9.6×10^4	1.5×10^8
Hardware (protected)	1,000	1	8.1×10^4	1.3×10^8	2.5×10^5	4.0×10^8

Table 1: Parameter combinations for reliably certifying vulnerability/security in different realistic leakage scenarios using the Bonferroni adjustment to control the false positive rate at an overall level α.

Table 1 shows that in the case of the ISO 17825, choice of $\alpha = 0.05$, and using the Bonferroni method to account for multiple testing (in any given scenario) more than 10^4 traces are required for a high powered test in case leaks of magnitude 0.04 need to be detected (we found leaks of that magnitude in Cortex M style processors with 10k traces in attack settings). For smaller leaks (magnitude of 0.001), considerably more traces are necessary (such leak would be present in typical hardware implementations). Clearly ISO 17825 needs to mandate more traces in the case of relying on detection only (which it does in the context of testing implementations of symmetric cryptography).

We have so far ignored implementations that perhaps do not show any leakage in the first moment. Generic leakage detection approaches that rely on mutual information [35, 5, 37], or tests that rely on preprocessing to make higher order leaks visible via first order statistics [44] can be utilised. However, these approaches typically require more traces per se, are lower power powered than first order statistics, or miss leakages that do not sit in central moments. A recent discussion on this topic can also be found in [10].

4.2.4 Potential for automation.

- In order to automate any statistical test, some parameters need to be known or decided beforehand.

  Either: The analyst fixes the desired rate of false positives (α), the desired rate of false negatives (β), and the minimum 'effect size' (i.e. the magnitude of the hypothetical difference) that the test needs to be sensitive to, and determines the number of samples required for the test to fit those parameters.

  Or: The analyst fixes α, β and the available number of samples, and uses these to determine the smallest effect size which the test will be sensitive to under those parameters.

  Or: The analyst fixes α, the effect size of interest, and the number of samples, and uses these to determine the power $1 - \beta$ of the test to detect an effect of the specified size or larger.

- Depending on the test to be automated, thought must also be given to the sample design, which needs to be representative of the variation that the test is designed to be sensitive to. For example, in the case of a 'specific' test [20], the signal for a particular intermediate can be improved by fixing (for a given key) all other bits/bytes/words of the state at that point and working backwards to identify the associated inputs.

When it comes to *mapping* detected leaks, automation is straightforward in the case that one is working with an instruction level simulator (such as in [36]), as the relationship between the indices in the trace and the instruction sequence is perfectly known. This holds true regardless of the type of test applied (i.e. specific versus non-specific). On the other hand, in the case that one is working with traces measured from a real device, mapping is only possible via specific tests, which inherently ensure that detected leakages are tied to known intermediate targets (or at least to highly correlated ones).

4.2.5 Focus: REASSURE Approach

In the context of a backwards evaluation, the starting point is that both evaluation details and countermeasures' randomness are public information for profiling the device.

Therefore both the detection and mapping are well understood problems that can generally be solved using first-order statistical tools as we discussed before. Alternatively, using the SNR or correlation-based tests provides a more quantitative view of the leakage and a more direct mapping to target intermediate values [18].

Note that in the open context of a backwards evaluation, the detection is expected to be successful and it is only these positive results that are easy to interpret. By contrast, and as we argued before, negative detection results in the context of a closed source protected (e.g., masked) implementation are not necessarily indicative of a secure implementation [48].

A similar observation holds for the mapping step, which can either be instantiated using the aforementioned SNR and correlation tests and combined with multivariate distinguishers, or can leverage simple dimensionality reduction tools such as Principal Component Analysis (PCA) [1] or Linear Discriminant Analysis (LDA) [49]: here as well, these tools are effective because the physical noise of an open implementation is typically assumed to be close to Gaussian.[4]

By contrast, the dimensionality reduction problem becomes hard with no optimal solutions in the context of higher-order and multivariate attacks [12].

4.3 Attacks and Exploitation

In the context of FIPS 140, leakage exploitation is foreseen only in the case of implementations of public key cryptosystems (see ISO 17825:2016). The attacks are somewhat categorised and an upper time limit is provided as well as an upper trace limit. The consideration of worst-case adversaries is not foreseen (limited profiling). Consequently, it is unlikely that in this context an evaluation would come close to an optimal, worst-case adversary.

In CC evaluations, considerably more rigour and effort goes into ascertaining the possibility of worst-case attacks. Interaction between evaluators and implementers/vendors is foreseen, and, thanks to JHAS, a list of up to date attack vectors is maintained. However, as there are no scientific grounds for inclusion (or exclusion) for this list provided, it is unclear if such a list can ever truly represent the state-of-the-art, or the worst-case adversary. Whilst evaluators select methods from this list (and their own expertise) it is also unclear if in any concrete evaluation the optimal practical adversary is indeed considered (what if that adversary is a combination of attack methods not yet on the list?).

The approach that REASSURE suggests **always** starts by considering the worst-case adversary and works "backwards" from it, relaxing assumptions one by one until an adversary is found that can be actually demonstrated.

In the remainder of this section we hence concentrate on arguing how confident we can be (in the context of the REASSURE approach) to actually reach the worst-case adversary with state-of-the-art methods.

[4] If needed, alternative tests and distinguishers do not make such an assumption [5, 37].

4.3.1 Optimality of Leakage Modeling (Profiling).

In the current state of the art, optimally modelling a (multivariate and higher-order) leakage function remains a complex problem even when the source code and randomness are given to the evaluator. The main reason for this is that the best model should be chosen in function of the implementation's security order (i.e., the lowest statistical moment of the leakage distribution that depends on the key) and finding this security order becomes expensive as the number of shares in a masking scheme increases. For low security orders, the best known approach is to try higher-order detection on selected tuples of samples (provided by the detection and mapping step) [44]. This is for example possible for the two additive shares of the ANSSI software implementation analysed by Bronchain and Standaert in [11]. For high security orders, this exhaustive approach remains expensive and may require considering security margins [28].

From another perspective, the problem of accurate and efficient leakage modeling is well illustrated by the numerous attempts to evaluate security with machine learning and deep learning algorithms [25, 30, 31, 13]. Such approaches generally work with minimum assumptions on the underlying leakage distribution (e.g., they do not assume the independence of consecutive leakage samples). But the cost of this generality is (in the current literature) a more expensive profiling step. Since the independence of leakage samples is also the origin of the security order reductions that make the optimal modelling of leakage distributions challenging, it is an important open problem to better understand the best tools to deal with this problem in a systematic manner.

4.3.2 Optimality of Information Extraction.

Given well detected Points-of-Interest (POIs) and well estimated templates that accurately model the leakage distribution, the extraction of information for the relevant target intermediate values in an implementation can simply be performed by evaluating the templates with fresh samples. This part of the attack is not expected to lead to sub-optimalities (and can be easily automated).

4.3.3 Optimality of Information Processing

For this last step, one should first distinguish between (what we next denote as) simple approaches and (what we next denote as) advanced ones.

Simple approaches include Divide-and-Conquer (D&C) attacks in the context of symmetric cryptography and Extend-and-Prune (E&P) attacks in the context of asymmetric cryptography (see the REASSURE Deliverable D1.3 for illustrations based on AES and ECC case studies). In this context, the information about different parts of the target secret are first combined in a maximum likelihood manner (which is optimal [16]). For symmetric algorithms, the remaining (full key) candidates can then be enumerated or their rank can be bounded (thanks to key knowledge). There is a large body of work on rank estimation that provide tight bounds, see for example [19, 34, 39, 33], and these state-of-the-art solutions should be close enough to optimal. The case of asymmetric cryptography is less covered but dedicated approaches have also been proposed there [29].

Advanced approaches include the algebraic (resp., analytical) attacks that target the secret key at once, as for example considered in [43] (resp., [53]) in the context of block ciphers, or in [41] for asymmetric cryptography. These attacks are in general more difficult to mount and to evaluate, due to their higher computational cost and sensibility to various inherently heuristic parameters (e.g., to deal with cycles in the circuit graphs) [22, 21]. It implies risks of security overstatements whenever such attacks provide a significant gain over the simpler D&C and E&P ones.
The different levels of understanding between simple and advanced approaches motivate the suggestion to study both approaches in a backwards evaluation, so that the distance between them can provide an indication of the risk related to the more heuristic nature of advanced approaches.

4.4 Summary

An informal summary of the state-of-the-art solutions that can be used for a **backwards security evaluation** is given in Table 2. As illustrated by the color code, some attack steps are quite well understood (in this context,

Table 2: REASSURE evaluation strategy: assurance summary.

Attack steps		Risk
Measurement and preprocessing		🟠
Detection and mapping		🟢
Leakage Modeling (Profiling)		🔴
Information extraction		🟢
Information Processing	Simple (D&C, E&P)	🟢
	Advanced (analytical)	🟠

where adversaries are given full access to randomness and keys) and there are various working solutions for them. This is typically the case of detection and mapping, information extraction, and simple (D&C and E&P) approaches to information processing, as discussed in this report.

The measurement and preprocessing step is introducing a first source of (moderate) risk, as there are no (and probably cannot be) theoretical ways to design optimal measurement setups, while this step is determining the noise level of the measurements, which is a key parameter for most algorithmic side-channel countermeasures (e.g., masking [15, 26], shuffling [24, 54], ...). Yet, this risk can and should be mitigated by the sound comparison of standard measurement boards and the sharing of good practices, possibly combined with some security margins for the expected measurement noise level.

Advanced information processing (with algebraic or analytical attacks) is bringing another source of (moderate) risk due to their more heuristic nature. Current practical evaluations however suggest that the security loss due to suboptimalities in these attacks is generally limited and can be captured by small security margins as well.

So eventually, we posit that the main source of risk in side-channel security evaluations remains in the modelling step. On the one hand, this is where the impact of strong adversarial capabilities is the most critical. On the other hand, even with strong adversarial capabilities, finding general solutions to accurately estimate higher-order and multivariate distributions is likely to remain a hard problem with a need of risk management to be further investigated.

In this respect, we mention the possibility to use bounds such as proposed in [9] to quantify the loss due to imperfect models. Yet, the worst-case bound based on the empirical distribution (i.e., that does not make any assumption on the leakage distribution) is becoming less and less tight when applied to more multivariate distributions, highlighting again the need of a sound risk assessment based on a good understanding of the more heuristic steps used in a security evaluation.

5 Connection with the CC approach

5.1 Rating the severity of attacks

CC now acknowledges with the use of "open samples" and "samples with known secrets" the importance of considering what we call worst-case adversaries. The CC approach tries to both assess the "severity" of attack vectors (i.e., how likely are they to succeed in a practical setting), as well as the assurance level that an evaluation offers. Obviously, the more countermeasures an implementation is based on, the more difficult it gets to truly judge both the severity of attacks as well as the assurance of an evaluation.

The problem with judging severity is evident because the document that describes the rating methodology as used by JHAS is ever evolving (and growing). Attacks are comprised of the same steps as evaluations (see Section 3.1) and therefore any innovation impacts on what is perceived to be "easy" or "hard". For instance, at the advent of power analysis and EM attacks, the standard equipment was typically a digital

storage oscilloscope, the associated probes, and other laboratory kit, all costing several ten thousand Euros. Nowadays, on the one hand, one can buy (for a fraction of the money) commercial kits (e.g., Chipwhisperer) that are sufficient for power analysing commodity microprocessors. On the other hand, the technology for oscilloscopes, probes, amplifiers, etc. has improved, as has our general understanding of what makes a good setup, so for the same tens of thousands of Euros one can now produce a setup (still based on off-the-shelf equipment) that is considerably better than in the early days. Consequently, setups are in some sense as "hard" as they were two decades ago, but there are perhaps more options available to get to a "good" setup much more easily these days. How should then this be accounted for in an evaluation?

Similarly, we now know more ways of profiling, and we understand better the security benefits of masking countermeasures, yet profiling in the presence of unknown masks remains computationally expensive: an adversary has to find ways to determine where/when masks are being processed, then they can potentially bias them, thereby rendering masking ineffective and key extraction possible; with that, and biased masks, profiling can start. Consequently, an adversary would need to do all these steps, and a determined adversary would/could; but how should this be dealt with in an evaluation, where evaluator time equates money?

JHAS, as mentioned above, now in principle acknowledges the need for profiled attacks (and to facilitate them open samples and samples with known secrets) in the context of smart cards and similar devices. But in its methodology, it then argues that if attacks only succeed because of open samples but would not succeed otherwise, then they should not be counted (in the rating system). However, we argue that it is difficult, if not impossible, to **prove** that an attack **must have** open samples/samples with known secrets to succeed.

Consequently we recommend that such worst-case attacks are always attempted and their results are always reported: with the specifics of the **initial profiling phase** (identification phase in the CC terminology) and the **actual exploitation phase** represented **separately** in any report. A question in all profiling attacks is the likelihood that templates that are computed on one device can be applied to another devices. We recommend to experimentally answer this question, in order to give more meaning to separation of rating for the identification and the exploitation ratings: if templates are highly portable, then it is not fair to rate attacks only via the sum of identification and exploitation. Instead, one then needs to consider an "initial" attack (which requires both identification and exploitation), and "follow on attacks" (which only require exploitation).

5.2 Reaching higher assurance levels

In order to reach higher certification assurance levels, the CC approach currently requires so-called formal verification. Yet, a look at the state-of-the-art solutions for formal verifications in the specific context of side-channel analysis (power analysis) suggests that they currently have limited impact on the assurance one can have in a final implementation.

More precisely, a few examples of recently published formal verification tools for masked implementations is given in Table 3. Informally, these tools range from abstract to more concrete and from direct to composition-based. More precisely, the most abstract tools directly work in Ishai et al.'s probing model (where the adversary can probe wires in an implementation) [26] while the more concrete ones try to incorporate some of the implementation defaults that can occur when compiling (in software) or synthetizing (in hardware) the circuits.

As for the type of verification, direct verification checks security properties that abstract circuits satisfy: its complexity rapidly grows with the complexity of the implementation and the number of shares. By contrast, composition-based verification assumes that the gadgets of a circuit satisfy some strong form of (composable) security and then checks that the composition rules are satisfied (e.g., after compilation or synthesis): its lower complexity allows analyzing full circuits.
The main observation derived from this table is that in the current state-of-the-art, even the most concrete tools still work in quite abstract variations of the probing model. So while they can be handy to anticipate some security flaws such as a lack of randomness, glitches or transition-based leakages, the benefits of these tools are quite ineffective to mitigate the evaluation risks listed in the previous section, which primarily relate to the

	Abstract	Concrete
Direct	Barthe et al. [3]	REBECCA [8] maskVerif [2]
Composition-based	maskComp [4] Tight Private Circuits [6]	fullVerif [14]

Table 3: Masking formal verification tools' overview.

need to model the noise distribution of the leakages accurately. We conclude from this observation that such tools are better suited to the preliminary testing of any implementation (to detect the aforementioned flaws early in the design process) rather than as a final step to enhance insurance (which they do not provide).

We note that nothing theoretically prevents developing verification tools that work in even more concrete models (e.g., the noisy leakage model [42]), which is an interesting scope for further investigations (since automation in such models is believed to be more complex). Yet, even in that case, the resulting tools will ultimately exhibit the same risk that higher-order multivariate distributions are hard to estimate as exhibited in the previous section. In other words, assurance and risk depend on whether the parts of a security evaluation that cannot be modelled formally can be sufficiently bounded. Hence, in the context of a security evaluation w.r.t side channels, all the parts of an evaluation that can be modelled formally are better used as an early requirement than as a final one.

References

[1] C. ARCHAMBEAU, E. PEETERS, F. STANDAERT, AND J. QUISQUATER, *Template attacks in principal subspaces*, in Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings, L. Goubin and M. Matsui, eds., vol. 4249 of Lecture Notes in Computer Science, Springer, 2006, pp. 1–14.

[2] G. BARTHE, S. BELAÏD, G. CASSIERS, P. FOUQUE, B. GRÉGOIRE, AND F. STANDAERT, *maskverif: Automated verification of higher-order masking in presence of physical defaults*, in Computer Security - ESORICS 2019 - 24th European Symposium on Research in Computer Security, Luxembourg, September 23-27, 2019, Proceedings, Part I, K. Sako, S. Schneider, and P. Y. A. Ryan, eds., vol. 11735 of Lecture Notes in Computer Science, Springer, 2019, pp. 300–318.

[3] G. BARTHE, S. BELAÏD, F. DUPRESSOIR, P. FOUQUE, B. GRÉGOIRE, AND P. STRUB, *Verified proofs of higher-order masking*, in Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I, E. Oswald and M. Fischlin, eds., vol. 9056 of Lecture Notes in Computer Science, Springer, 2015, pp. 457–485.

[4] G. BARTHE, S. BELAÏD, F. DUPRESSOIR, P. FOUQUE, B. GRÉGOIRE, P. STRUB, AND R. ZUCCHINI, *Strong non-interference and type-directed higher-order masking*, in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016, E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, eds., ACM, 2016, pp. 116–129.

[5] L. BATINA, B. GIERLICHS, E. PROUFF, M. RIVAIN, F. STANDAERT, AND N. VEYRAT-CHARVILLON, *Mutual information analysis: a comprehensive study*, J. Cryptology, 24 (2011), pp. 269–291.

[6] S. BELAÏD, D. GOUDARZI, AND M. RIVAIN, *Tight private circuits: Achieving probing security with the least refreshing*, in Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II, T. Peyrin and S. D. Galbraith, eds., vol. 11273 of Lecture Notes in Computer Science, Springer, 2018, pp. 343–372.

[7] B. BILGIN AND J. FISCHER, eds., *Smart Card Research and Advanced Applications, 17th International Conference, CARDIS 2018, Montpellier, France, November 12-14, 2018, Revised Selected Papers*, vol. 11389 of Lecture Notes in Computer Science, Springer, 2019.

[8] R. BLOEM, H. GROSS, R. IUSUPOV, B. KÖNIGHOFER, S. MANGARD, AND J. WINTER, *Formal verification of masked hardware implementations in the presence of glitches*, in Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II, J. B. Nielsen and V. Rijmen, eds., vol. 10821 of Lecture Notes in Computer Science, Springer, 2018, pp. 321–353.

[9] O. BRONCHAIN, J. M. HENDRICKX, C. MASSART, A. OLSHEVSKY, AND F. STANDAERT, *Leakage certification revisited: Bounding model errors in side-channel security evaluations*, in Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I, A. Boldyreva and D. Micciancio, eds., vol. 11692 of Lecture Notes in Computer Science, Springer, 2019, pp. 713–737.

[10] O. BRONCHAIN, T. SCHNEIDER, AND F. STANDAERT, *Multi-tuple leakage detection and the dependent signal issue*, IACR Trans. Cryptogr. Hardw. Embed. Syst., 2019 (2019), pp. 318–345.

[11] O. BRONCHAIN AND F. STANDAERT, *Side-channel countermeasures' dissection and the limits of closed source security evaluations*, IACR Trans. Cryptogr. Hardw. Embed. Syst., 2020 (2020), pp. 1–25.

[12] E. CAGLI, C. DUMAS, AND E. PROUFF, *Kernel discriminant analysis for information extraction in the presence of masking*, in Smart Card Research and Advanced Applications - 15th International Conference, CARDIS 2016, Cannes, France, November 7-9, 2016, Revised Selected Papers, K. Lemke-Rust and M. Tunstall, eds., vol. 10146 of Lecture Notes in Computer Science, Springer, 2016, pp. 1–22.

[13] E. CAGLI, C. DUMAS, AND E. PROUFF, *Convolutional neural networks with data augmentation against jitter-based countermeasures - profiling attacks without pre-processing*, in Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings, 2017, pp. 45–68.

[14] G. CASSIERS, B. GRÉGOIRE, I. LEVI, AND F. STANDAERT, *Hardware private circuits: From trivial composition to full verification*, IACR Cryptology ePrint Archive, 2020 (2020), p. 185.

[15] S. CHARI, C. S. JUTLA, J. R. RAO, AND P. ROHATGI, *Towards sound approaches to counteract power-analysis attacks*, in Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings, M. J. Wiener, ed., vol. 1666 of Lecture Notes in Computer Science, Springer, 1999, pp. 398–412.

[16] S. CHARI, J. R. RAO, AND P. ROHATGI, *Template attacks*, in Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers, B. S. K. Jr., Ç. K. Koç, and C. Paar, eds., vol. 2523 of Lecture Notes in Computer Science, Springer, 2002, pp. 13–28.

[17] A. DUC, S. FAUST, AND F. STANDAERT, *Making masking security proofs concrete (or how to evaluate the security of any leaking device), extended version*, J. Cryptology, 32 (2019), pp. 1263–1297.

[18] F. DURVAUX AND F. STANDAERT, *From improved leakage detection to the detection of points of interests in leakage traces*, in Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I, M. Fischlin and J. Coron, eds., vol. 9665 of Lecture Notes in Computer Science, Springer, 2016, pp. 240–262.

[19] C. GLOWACZ, V. GROSSO, R. POUSSIER, J. SCHÜTH, AND F. STANDAERT, *Simpler and more efficient rank estimation for side-channel security assessment*, in Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers, G. Leander, ed., vol. 9054 of Lecture Notes in Computer Science, Springer, 2015, pp. 117–129.

[20] G. GOODWILL, B. JUN, J. JAFFE, AND P. ROHATGI, *A testing methodology for side-channel resistance validation*, in NIST Non-invasive attack testing workshop, 2011.

[21] J. GREEN, A. ROY, AND E. OSWALD, *A systematic study of the impact of graphical models on inference-based attacks on AES*, in Bilgin and Fischer [7], pp. 18–34.

[22] V. GROSSO AND F. STANDAERT, *Asca, SASCA and DPA with enumeration: Which one beats the other and when?*, in Iwata and Cheon [27], pp. 291–312.

[23] S. GUILLEY, H. MAGHREBI, Y. SOUISSI, L. SAUVAGE, AND J. DANGER, *Quantifying the quality of side channel acquisitions*, COSADE, February, (2011).

[24] C. HERBST, E. OSWALD, AND S. MANGARD, *An AES smart card implementation resistant to power analysis attacks*, in Applied Cryptography and Network Security, 4th International Conference, ACNS 2006, Singapore, June 6-9, 2006, Proceedings, J. Zhou, M. Yung, and F. Bao, eds., vol. 3989 of Lecture Notes in Computer Science, 2006, pp. 239–252.

[25] A. HEUSER AND M. ZOHNER, *Intelligent machine homicide - breaking cryptographic devices using support vector machines*, in Constructive Side-Channel Analysis and Secure Design - Third International Workshop, COSADE 2012, Darmstadt, Germany, May 3-4, 2012. Proceedings, W. Schindler and S. A. Huss, eds., vol. 7275 of Lecture Notes in Computer Science, Springer, 2012, pp. 249–264.

[26] Y. ISHAI, A. SAHAI, AND D. A. WAGNER, *Private circuits: Securing hardware against probing attacks*, in Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings, D. Boneh, ed., vol. 2729 of Lecture Notes in Computer Science, Springer, 2003, pp. 463–481.

[27] T. IWATA AND J. H. CHEON, eds., *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, vol. 9453 of Lecture Notes in Computer Science, Springer, 2015.

[28] A. JOURNAULT AND F. STANDAERT, *Very high order masking: Efficient implementation and security evaluation*, in Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings, 2017, pp. 623–643.

[29] T. LANGE, C. VAN VREDENDAAL, AND M. WAKKER, *Kangaroos in side-channel attacks*, in Smart Card Research and Advanced Applications - 13th International Conference, CARDIS 2014, Paris, France, November 5-7, 2014. Revised Selected Papers, M. Joye and A. Moradi, eds., vol. 8968 of Lecture Notes in Computer Science, Springer, 2014, pp. 104–121.

[30] L. LERMAN, S. F. MEDEIROS, G. BONTEMPI, AND O. MARKOWITCH, *A machine learning approach against a masked AES*, in Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers, A. Francillon and P. Rohatgi, eds., vol. 8419 of Lecture Notes in Computer Science, Springer, 2013, pp. 61–75.

[31] L. LERMAN, R. POUSSIER, G. BONTEMPI, O. MARKOWITCH, AND F. STANDAERT, *Template attacks vs. machine learning revisited (and the curse of dimensionality in side-channel analysis)*, in Constructive Side-Channel Analysis and Secure Design - 6th International Workshop, COSADE 2015, Berlin, Germany, April 13-14, 2015. Revised Selected Papers, S. Mangard and A. Y. Poschmann, eds., vol. 9064 of Lecture Notes in Computer Science, Springer, 2015, pp. 20–33.

[32] S. MANGARD, *Hardware countermeasures against DPA ? A statistical analysis of their effectiveness*, in Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings, T. Okamoto, ed., vol. 2964 of Lecture Notes in Computer Science, Springer, 2004, pp. 222–235.

[33] D. P. MARTIN, L. MATHER, AND E. OSWALD, *Two sides of the same coin: Counting and enumerating keys post side-channel attacks revisited*, in Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings, N. P. Smart, ed., vol. 10808 of Lecture Notes in Computer Science, Springer, 2018, pp. 394–412.

[34] D. P. MARTIN, J. F. O'CONNELL, E. OSWALD, AND M. STAM, *Counting keys in parallel after a side channel attack*, in Iwata and Cheon [27], pp. 313–337.

[35] L. MATHER, E. OSWALD, J. BANDENBURG, AND M. WÓJCIK, *Does my device leak information? an a priori statistical power analysis of leakage detection tests*, in Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I, K. Sako and P. Sarkar, eds., vol. 8269 of Lecture Notes in Computer Science, Springer, 2013, pp. 486–505.

[36] D. MCCANN, E. OSWALD, AND C. WHITNALL, *Towards practical tools for side channel aware software engineering: 'grey box' modelling for instruction leakages*, in 26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017., E. Kirda and T. Ristenpart, eds., USENIX Association, 2017, pp. 199–216.

[37] A. MORADI, B. RICHTER, T. SCHNEIDER, AND F. STANDAERT, *Leakage detection with the x2-test*, IACR Trans. Cryptogr. Hardw. Embed. Syst., 2018 (2018), pp. 209–237.

[38] D. OSWALD AND C. PAAR, *Improving side-channel analysis with optimal linear transforms*, in Smart Card Research and Advanced Applications - 11th International Conference, CARDIS 2012, Graz, Austria, November 28-30, 2012, Revised Selected Papers, S. Mangard, ed., vol. 7771 of Lecture Notes in Computer Science, Springer, 2012, pp. 219–233.

[39] R. POUSSIER, F. STANDAERT, AND V. GROSSO, *Simple key enumeration (and rank estimation) using histograms: An integrated approach*, in Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings, B. Gierlichs and A. Y. Poschmann, eds., vol. 9813 of Lecture Notes in Computer Science, Springer, 2016, pp. 61–81.

[40] S. M. D. POZO AND F. STANDAERT, *Blind source separation from single measurements using singular spectrum analysis*, in Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings, T. Güneysu and H. Handschuh, eds., vol. 9293 of Lecture Notes in Computer Science, Springer, 2015, pp. 42–59.

[41] R. PRIMAS, P. PESSL, AND S. MANGARD, *Single-trace side-channel attacks on masked lattice-based encryption*, in Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings, 2017, pp. 513–533.

[42] E. PROUFF AND M. RIVAIN, *Masking against side-channel attacks: A formal security proof*, in Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings, T. Johansson and P. Q. Nguyen, eds., vol. 7881 of Lecture Notes in Computer Science, Springer, 2013, pp. 142–159.

[43] M. RENAULD, F. STANDAERT, AND N. VEYRAT-CHARVILLON, *Algebraic side-channel attacks on the AES: why time also matters in DPA*, in Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings, C. Clavier and K. Gaj, eds., vol. 5747 of Lecture Notes in Computer Science, Springer, 2009, pp. 97–111.

[44] T. SCHNEIDER AND A. MORADI, *Leakage assessment methodology - extended version*, J. Cryptographic Engineering, 6 (2016), pp. 85–99.

[45] S. SKOROBOGATOV, *Synchronization method for SCA and fault attacks*, J. Cryptographic Engineering, 1 (2011), pp. 71–77.

[46] SOG-IS, *Application of attack potential to smartcards and similar devices*, 2019.

[47] ——, *Attack methods for smartcards and similar devices*, 2020.

[48] F. STANDAERT, *How (not) to use welch's t-test in side-channel security evaluations*, in Bilgin and Fischer [7], pp. 65–79.

[49] F. STANDAERT AND C. ARCHAMBEAU, *Using subspace-based template attacks to compare and combine power and electromagnetic information leakages*, in Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings, E. Oswald and P. Rohatgi, eds., vol. 5154 of Lecture Notes in Computer Science, Springer, 2008, pp. 411–425.

[50] F. STANDAERT, T. MALKIN, AND M. YUNG, *A unified framework for the analysis of side-channel key recovery attacks*, in Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings, A. Joux, ed., vol. 5479 of Lecture Notes in Computer Science, Springer, 2009, pp. 443–461.

[51] THE COMMON CRITERIA. https://www.commoncriteriaportal.org/.

[52] J. G. J. VAN WOUDENBERG, M. F. WITTEMAN, AND B. BAKKER, *Improving differential power analysis by elastic alignment*, in Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings, A. Kiayias, ed., vol. 6558 of Lecture Notes in Computer Science, Springer, 2011, pp. 104–119.

[53] N. VEYRAT-CHARVILLON, B. GÉRARD, AND F. STANDAERT, *Soft analytical side-channel attacks*, in Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I, P. Sarkar and T. Iwata, eds., vol. 8873 of Lecture Notes in Computer Science, Springer, 2014, pp. 282–296.

[54] N. VEYRAT-CHARVILLON, M. MEDWED, S. KERCKHOF, AND F. STANDAERT, *Shuffling against side-channel attacks: A comprehensive study with cautionary note*, in Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings, X. Wang and K. Sako, eds., vol. 7658 of Lecture Notes in Computer Science, Springer, 2012, pp. 740–757.

[55] C. WHITNALL AND E. OSWALD, *A critical analysis of ISO 17825 ('testing methods for the mitigation of non-invasive attack classes against cryptographic modules')*, in Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III, S. D. Galbraith and S. Moriai, eds., vol. 11923 of Lecture Notes in Computer Science, Springer, 2019, pp. 256–284.

