# A Systematic Appraisal of Side Channel Evaluation Strategies*

Melissa Azouaoui[1,2], Davide Bellizia[2], Ileana Buhan[3], Nicolas Debande[4], Sèbastien Duval[2], Christophe Giraud[4], Èliane Jaulmes[5], François Koeune[2], Elisabeth Oswald[6,7], François-Xavier Standaert[2], and Carolyn Whitnall[7]

[1] NXP Semiconductors, Hamburg, Germany,
[2] UCLouvain, ICTEAM, Louvain-la-Neuve, Belgium,
[3] Radboud University, Nijmegen, Netherlands,
[4] Idemia, Paris, France,
[5] SGDSN, Paris, France,
[6] AAU, Klagenfurt, Austria, Elisabeth.Oswald@aau.at,
[7] UoB, Bristol, UK

**Abstract.** In this paper we examine the central question that is how well do side channel evaluation regimes capture the true security level of a product. Concretely, answering this question requires considering the optimality of the attack/evaluation strategy selected by the evaluator, and the various steps to instantiate it. We draw on a number of published works 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.

**Keywords:** Side channels, Evaluation, Certification

## 1 Introduction

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. There exist two testing/evaluation regimes at present. The first regime operates within the Common Criteria (CC) framework [23][8] whereby side channel (and other implementation related) attacks have been picked up early as a threat that warrants specialist

* This work has been funded in parts by the European Union (EU) via the H2020 project 731591 (acronym REASSURE), the ERC project 724725 (acronym SWORD) and the ERC project 725042 (acronym SEAL). François-Xavier Standaert is a senior research associate of the Belgian Fund for Scientific Research (FNRS-F.R.S.). Ileana Buhan was with Riscure at the time of conducting this research. The final authenticated version will appear in the proceedings of SSR 2020.

[8] The most recent version of all documents relating to CC evaluations can be found on wwww.commoncriteriaportal.com

consideration, in particular in the context of smart cards. The second regime operates within the framework of FIPS 140 [36]; there is a transition effort currently ongoing to move from 140-2 towards 140-3, the latter explicitly considers side channel attacks.

Within the context of CC, stakeholder groups such as JHAS[9], 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 confidential 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 [48].

In contrast, the FIPS 140 approach is to keep the *cost of evaluation to an absolute minimum* by mandating no more than conformance style testing as specified in ISO 17825:2016[24]. 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 [16] to assess the threat of side channel attacks.

*Contributions.* In this paper we are concerned with the central question of how well do such evaluation regimes capture the true security level of a product. Concretely, answering this question requires considering the optimality of the attack/evaluation strategy selected by the evaluator, and the various steps to instantiate it. We also point towards a third evaluation strategy, based on working backwards from the worst-case adversary, which has emerged in the academic literature. We draw on a number of published works 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.

### 1.1 Organisation and Outline of this Paper

We provide a brief explanation of the two evaluation regimes (Common Criteria and FIPS 140) in Section 2. We suggest a third technique (we call this the worst case adversary) in Section 3, and discuss some examples where such an approach was in fact used in the academic community. 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 Industrial Evaluation Approaches

Whilst there are a number of security evaluation approaches possible in principle, two schemes (and derivatives thereof) dominate in industrial practice. Common

[9] (JIL Hardware Attacks Subgroup), they operate within the International Security Certification Initiative (ISCI)

Criteria evaluations are "attack driven" and aim to systematically capture and categorise attack vectors. The Common Criteria methodology is adopted as an international standard via ISO 15408. Common criteria features a range of assurance levels (so called EALs), and to reach the higher level requires more rigorous testing. The goal of a Common Criteria evaluation is to check the security claims made by a manufacturer and testing against side channels is typically included.

FIPS 140 evaluations are "conformance style" evaluations that rely on checking some minimum criteria relating to the security of a product. FIPS 140-2 is mandated in the US (FIPS 140-3 will replace FIPS 140-2 late in 2020), it is also used in Canada and 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).

Both approaches 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 several documents. The product which is being certified is called the Target of Evaluation (TOE). For a TOE two documents are of relevance: the Protection Profile (PP) and the Security Target (ST). The Protection Profile is a generic document for a category of product (e.g. Travel documents, Java Cards, IC, *etc.* ), often created by a user community. It 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. Such document insures that a product is conform to a security goal and provides the expected security features. It is not mandatory to rely on a PP, but if one exists for a kind of products, it is recommended to use it. The Security Target details the secure implementation of the TOE and may use (or not) a PP as reference. It uniquely identifies the product and describes the assets, the threats, the security objectives (both on the TOE and on the environment), the perimeter of the evaluation, the SFRs and the life cycle. Vendors often make the Security Target details available to their customers.

During the Common Criteria evaluation process, vendors must state an envisioned security level. This is called the Evaluation Assurance Level (EAL). The EAL indicates a minimal level for each subclass (development process, guidance, conformity of security target, vulnerability assessment, *etc.* ) that will be taken into account during the evaluation. It reflects the rigour of the evaluation. There are seven levels of EALs, with EAL 1 being the most basic and level 7 being the most rigorous. One can pick a level and "augment" it with specific requirements from a higher level. It is imperative to understand that higher EALs do not necessarily imply a higher level of security, they imply that the claimed security assurance of the TOE has been more rigorously verified. Among all subclasses,

the more relevant for practical security is AVA_VAN (vulnerability assessment), with levels going from 1 (resistance to basic attackers) to 5 (resistance to attackers with high attack potential). It describes the search for vulnerabilities and define a rating scale for attacks, depending on the means of the adversary.

**Smart card/integrated circuit evaluations** In the specific case of smart cards, the International Security Certification Initiative (ISCI) brings together stakeholders from every aspect of smart card security evaluations: 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" [48] provides a "rating system" for attacks. The "Attack Methods for Smart Cards and Similar Devices" [49] 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 and corresponding attack vectors for the first time; exploitation refers to attacks utilising the results from identification). The factors that are considered are: Elapsed time, Expertise, Knowledge of TOE, Access to TOE, Used equipment, Open samples[10]. The same rating scheme is also used by EMVCo (a "derivative" of the CC approach that we discuss).

### 2.2 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 via a link to several 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 in the context of symmetric encryption, ISO/IEC 17825:2016 relies on using leakage detection procedures instead of attempting attacks. In all other scenarios it requires to test against standard DPA style attacks (the type of attacks/methodology are listed in the standard). Leakage detection involves producing evidence for the presence of leaks using statistical hypothesis testing. It has been advertised as a "cheaper

[10] For the sake of succinctness we refer the reader to the JHAS documentation for a precise definition of these factors[48].

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 against the absence/presence of leaks.

## 3 An Alternative: Backwards Evaluations

The goal of an evaluation is to ascertain the true security level of a product (either in absolute terms by checking explicit claims by the manufacturer or in relative terms via ensuring that it is at least as secure as given by some minimum criteria) and our research question for this work is how well existing evaluation regimes capture the true security level of a product. The NIST/FIPS approach for side channels (via ISO/IEC 17825:2016) sets the bar rather low by mandating a testing regime that captures a well resourced and capable adversary (we a provide a more in-depth critique in the next section). But it is far from mandating even a "best practical adversary" (as it happens in the CC approach). Defining the "best practical adversary" is hard because "practical" is somewhat subjective and tends to change over time. In contrast, the definition of a worst-case adversary (and working backwards, i.e. relaxing assumptions) is often less ambiguous and therefore academic works have increasingly utilised this approach. Such a worst-case adversary will utilize multiple leaking intermediate variables, a multivariate characterization of each leaking intermediate variable, divide-and-conquer or analytical information extraction and enumeration capabilities. For this, various types of capabilities, for example in terms of knowledge of the target implementation and profiling abilities, can be granted to the adversary. Academic research has featured this type of adversary in published works, and we will link to two concrete such examples in the next section.

### 3.1 Worst-Case Adversary

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) as well as over the secret parameters (keys, randomness). They can turn off any countermeasures (should the target allow turning them off), and has detailed implementation knowledge (e.g., source code in the case of software implementations, or a hardware level description in the case of hardware implementations). The worst-case adversary is pushing the separation between the identification of the attack and its exploitation to the extreme: *it essentially enables practically unbounded profiling efforts in order to reach the strongest online attack.*

For instance in case of an AES hardware implementation that employs a special logic style that does not require extra randomness, the worst-case adversary would have full information about the properties of that logic style, and he would be able to choose keys and inputs. They would also have full information

about the AES architecture. With this information, a profiling attack should be attempted (using either statistical modelling, machine learning or deep learning).

In case of an AES software implementation that employs software masking and shuffling, the worst-case adversary would have the source code, control over inputs (plaintexts/ciphertexts), key, and knowledge of randomness (for both masking and shuffling). This is because in software it is realistic to output randomness without significantly changing the leakage characteristics of the rest of the implementation (therefore the countermeasure can be made accessible during evaluation, but this access can be completely removed when the software is deployed). With these assumptions, the evaluator can again conduct a profiling attack and we describe in a subsequent section one such concrete example.

During an evaluation, a natural goal is therefore to come as close as possible to the worst-case adversary, by first granting them with the maximum (even if not always realistic) capabilities. Thanks to such advanced capabilities, it is in general possible to (*i*) identify (from the documentation) the predictable target values that may occur separated in the time domain, and the predictable target vaues that occur within each clock cycle, (*ii*) attempt characterization (potentially by using a biased trace set if documentation suggests when masks may leak)[11]. As a result, a backwards evaluation suggests to start from such a powerful (yet easier to specify) adversary and, once concretely analyzed, to discuss the consequences of relaxing different adversarial capabilities for the feasibility of the attack, and the additional (profiling or online) attack complexity this relaxation implies. Arguing from this angle provides at least a stable starting point, and a fairly well defined set of steps which fit to processes which are (to the best of our knowledge) already standard.

Thus this approach advocates that, if at all possible, 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 adversarial assumptions or capabilities can be gradually relaxed, and attacks 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.

### 3.2 Evaluation steps

We propose to consider any evaluation as a composition (possibly iterative) of the following key steps:

[11] A target value is an intermediate value that the adversary/evaluator can predict based on knowing (parts of) the input and guessing parts of the key

1. **Measurement and preprocessing.** This step provides the adversary (evaluators) with leakages (e.g., the power consumption or electromagnetic radiation of a chip, or their simulation in case simulated analyses are considered) based on their 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 their profiling abilities to estimate a model for the leakages.
   (b) **Information extraction phase.** In this phase, the adversary (evaluator) extracts information about intermediate values manipulated by their target implementation thanks to a model (that can be obtained from a modelling phase or assumed a priori).
   (c) **Information processing.** In this final phase, the adversary (evaluator) combines the partial information they extracted from their target implementation and aggregates this information in order to recover some secret parameter (e.g., a master key).

We now illustrate the backwards approach based on the Worst-Case Adversary with reference to two concrete papers that were published recently: a masked AES implementation proposed by the French ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information) [2], recently analyzed in [5], and an (unprotected) ECC scalar multiplication analyzed in [40, 1].

### 3.3 Case Study: Masked AES implementation

**Instantiation of a worst-case adversary.** The ANSSI implementation that was analysed by [5] is a protected implementation combining additive and multiplicative secret sharing into an affine masking scheme [14], which is additionally mixed with a shuffled execution [20]. It is running on an ARM Cortex-M4 architecture. Preliminary leakage assessments did not reveal data dependencies with up to 100,000 measurements (i.e. following a TVLA style leakage assessment). Bronchain and Standaert considered a worst-case adversary with no specific device preparation, a single device sample, full control of the AES inputs and outputs, full profiling capabilities (i.e., knowledge of the key and randomness during profiling), knowledge of the (open source) software implementation, limited knowledge of the hardware details (i.e., the general architecture of the ARM Cortex family), with a simple measurement setup worth a few thousands of euros. The attack steps listed in Section 3.2 of their worst-case attack can be detailed as follows.

*Measurement setup:* the target board has been modified by removing decoupling capacitors and measurements were taken at 1[Gsamples/s] with a PicoScope (while the chip was running at 48MHz). The probe position was optimized in function of the Signal-to-Noise Ratio (SNR) [32] of the multiplicative mask. No additional preprocessing (e.g., filtering) was performed on the traces. The SNR of the computation samples was typically in the 0.1 range, while it was significantly higher (more than 10) for the memory accesses needed during the precomputations of the multiplicative mask tables.

*Leakage detection and mapping:* most target intermediate variables are identified based on the SNR metric. In the case of the multiplicative mask precomputations, a dimensionality reduction based on principal component analysis (PCA) was additionally performed (which allowed recovering this mask in full).

*Modelling:* all the randomized target intermediate variables can be modeled with Gaussian mixtures as per [52, 55]. Thanks to the knowledge of the randomness during profiling, this was done straightforwardly by estimating first-order (sometimes multivariate) Gaussian templates [9].

*Information extraction:* Bronchain and Standaert considered the dissection of countermeasures. That is, they targeted the different countermeasures (i.e., the additive mask, the multiplicative mask and the shuffling) independently in order to reduce the physical noise amplification they respectively imply. Thanks to this approach, the multiplicative mask was recovered in full, the shuffling permutation was recovered with high probability, leaving the adversary with the need to attack a two-share Boolean masking scheme with multivariate templates.

Finally, the information extracted on the different target intermediate variables was accumulated on the long-term key using a standard maximum likelihood approach. Key information was then post-processed with a key enumeration algorithm [39]. As a result, the best attack was able to reduce the 128-bit key rank below $2^{32}$ with less than 2,000 measurements.

**Relaxing capabilities.** Compared to the leakage assessment in [2], the main improvement in the dissection attack described above is that it exploits multiple target intermediate variables and multiple leakage samples per target. For this purpose, the two most critical adversarial capabilities are ($i$) the implementation knowledge made available thanks to the open source library and ($ii$) the possibility to profile models efficiently thanks to the randomness knowledge. As discussed in [5], removing these capabilities makes the attack substantially harder.

On the one hand, purely black box approaches (e.g., based on machine learning) seem unable to efficiently identify the different countermeasures as exploited in a dissection attack [5]. So in absence of implementation knowledge, it is unlikely that an attack can directly target the additive and multiplicative masks and the shuffling separately, implying a significant (multiplicative) increase of the

overall attack cost. Such a difficulty could be overcome with advanced techniques such as [12] which are, however, less studied and understood than standard side-channel attacks.

On the other hand, profiling Gaussian mixtures without mask knowledge is known to be a hard task. A work by Lerman et al. discusses options for this purpose [29], but the profiling cost is significantly higher than in the known randomness case (another solution is [28]). Alternatively, one can attack using a non-profiled higher-order side-channel attack [43]. However such a strategy (based on the estimation of a higher-order statistical moment rather than a mixture) becomes increasingly suboptimal as the level of noise in the implementation decreases [50]. When combined together, the lack of implementation knowledge and the unknown randomness during profiling additionally imply that tuples of Points-of-Interest (POIs) must be detected exhaustively, which is also known to be a hard task [11, 6]. For illustration, the complexities of the worst-case attack put forward by Bronchain and Standaert and the single-target attack discussed in the preliminary security assessment of the ANSSI implementation differ by a factor $> \frac{100,000}{2,000} = 50$.

### 3.4 Case Study: ECC scalar multiplication

**Instantiating a worst-case adversary.** The ECSM implementation analyzed in [40, 1] is a constant-time Montgomery ladder using Jacobian coordinates on the NIST P-256 curve and the target device is an ARM cortex-M4 with no specific preparation. The worst-case adversary is assumed to have full control of the inputs and full profiling capabilities. The generic evaluation framework designed by Poussier et al. [40] is horizontal (i.e. it utilises multiple leakage points per leakage trace) and allows extracting most of the information in the leakage traces. The main vector of the attack is that for each scalar bit a regular ECSM performs a fixed and predictable sequence of operations. These operations lead to several leakages on intermediate values that depend on the scalar bit and the input point. Following an Extend and Prune (E&P) strategy, once one bit is recovered, the internal state of the ECSM is known and the following bit can be recovered in the same way. The general steps of the evaluation, as outlined in Section 3.2, are summarised below.

*Measurement setup:* the voltage variation was monitored using a 4.7 $\Omega$ resistor. The traces were acquired using a Lecroy WaveRunner HRO 66 ZI oscilloscope running at 200MHz. The target micro-controller runs at 100MHz. No preprocessing was applied to the traces. The average SNR achieved by the targeted ALU operations was around 0.018.

*Leakage detection and mapping:* POIs corresponding to target intermediate values are identified using classical selection techniques such as correlation [11] (based on a simple estimated model) or SNR based ones.

*Modelling:* once the time locations of all the target intermediates are found, they can be modelled using classical Gaussian templates [9], but a full profiling (i.e. assuming that all 32 bits in a variable can interact) on 32-bit variables is too measurement intensive. As a result, Poussier et al. rather use a regression based approach with only a linear 32-bit basis [45], which significantly speeds up the modelling phase of the 32-bit target registers.

*Information extraction:* using the previous regression based modeling and a single side-channel trace, the conditional probabilities of all the target intermediates are evaluated.

Finally, all the information is processed by simply multiplying all the intermediate's probabilities to evaluate the most likely value for the scalar bit. Based on the E&P strategy, to recover the following bit at index $i+1$, the intermediate values are not only predicted based on the value of the bit at index $i+1$ but also on the previously guessed value of the bit at index $i$. This is due to the recursive nature of ECSM algorithms. On the target implementation, a scalar bit is recovered with high confidence when 1000 or more intermediate values are exploited.

While all previous steps were described for a single scalar bit, they can be easily extended to consider words of the scalar. For example instead of considering only two possible sequences of intermediate values, the analysis can be extended to $n$-bit limbs ($n$ is typically small) and thus now the attack requires to predict $2^n$ intermediate value sequences instead of 2. After the previous attack, in the case of ECDH, computational power can be exploited in order to mitigate a possible lack of information using enumeration [27], and to recover the full value of the scalar. For ECDSA, a potential strategy is to partially attack the random nonces, recover their first few bits, and apply lattice cryptanalysis in order to recover the secret scalar [37]. Lattice attacks are hindered by errors on the nonces' bits. However based on the nonces' probabilities after a side-channel attack, it is possible to select only a few nonces' with a probability above a certain threshold, and discard the others to maximize the success of the lattice attack. Based on this combination of tools, the ECDSA key can be recovered using 4 bits of 140 nonces.

**Relaxing capabilities.** The previously described evaluation strategy is designed to exploit the leakage of all the intermediate values computed during the execution of the ECSM. This is made easy by the detailed knowledge of the code that an open evaluation enables. However, even when the attacker is not assumed to have access to this information, a similar evaluation strategy is still possible for a lower (yet still high) number of intermediate values that the attacker can guess. That is, while reverse engineering the ECSM code is a possible but tedious option, the structure of the elliptic curve and the fact that ECSM algorithms always perform point addition and point doubling routines make it possible for the adversary to test a few "natural" options for how point and field operations are implemented in practice. This step could be emulated

by the evaluator/adversary based on openly available ECC implementations, for example.

Typically, the original attack of Poussier et al [40] exploits 1,600 intermediate values based on the knowledge of the multiplication algorithm. By mapping some intermediate values to the side-channel traces, it is possible for an attacker to try identifying the multiplication, addition and modular reduction algorithms used. For instance (i.e., based on the above experiment), an attacker who has knowledge of the multiplication algorithm could exploit roughly 46% of the key dependent leakage, an attacker able to identify the addition algorithm (which is in most cases the easiest to recover) can exploit 3% of the key dependent leakage and an attacker having access to the modular reduction code can additionally exploit over 50% of the leakage.

Tools such as the shortcut formula given by Azouaoui et al. [1] can then help evaluators to predict the success rate of the previous attack for a varying number of intermediate values, without having to implement the attack in full and with minimal modeling.

Overall, we conclude that while the knowledge of the implementation details is helpful to rapidly reach a close to worst-case attack, strong horizontal attacks are still possible without this knowledge. This is in contrast with the case of a masked AES implementation in the previous section. The main reason of this observation is that an unprotected ECSM implementation has many targets that can be very efficiently identified with simple (correlation or SNR) tools.

## 4 Optimality of Evaluation Steps

We now discuss the optimality of the state-of-the-art tools that can be used for various attack steps in the context of the three evaluation approaches that we have introduced.

### 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 greatly on the final outcome of the practical evaluation of a device. Based 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.

Due to its physical nature, the optimality of the measurement and preprocessing step is hard to quantify. The quality of a measurement setup is indeed mostly dependent on hard to evaluate engineering expertise. A badly designed setup may lead to higher noise in the time and amplitude domains that directly

affect the attack complexity [32], and the impact of which exponentially increases whenever combined with countermeasures such as masking [10].

Preprocessing is similarly heuristic. Many published solutions exist to filter the noise [38, 41] and to resynchronise the traces [47, 53], 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, 19] and information theoretic metrics for multivariate evaluations [51]).

In the context of CC/JHAS and the worst case approach the emphasis is on working towards the best setup. In the FIPS 140 case, the corresponding public ISO standards advocate checking against some set target devices in order to argue the quality of a setup. Neither approach is able to substantially change the heuristic nature of setups and configurations, thus there cannot be any claims towards optimality.

### 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. Mapping of leaks is about associating identified leaks with intermediate values. Leakage can be detected using statistical hypothesis tests for independence. These can be based on (non-parametric) comparisons between generic distributional features or on (parametric) comparisons between moments and related quantities, and vary in complexity and scope depending on whether one is interested in univariate or multivariate settings. 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 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 precursor 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.

**Detect and Then Attack: CC and worst case approach** 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 $\alpha$. 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 $\beta$ 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.

It is well known that the two errors can be traded-off against one another, and mitigated (but not eliminated) by:

- Increasing the **sample size** , intuitively resulting in more evidence from which to draw a conclusion.
- Increasing the minimum **effect size** of interest, 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[35].

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 where attacks follow detection, the consequences of missing out on some leaks (because of a lack of statistical power) is not as severe (as a test with a lower power 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.

Note that in the open context of a worst-case adversary, 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 [50].

A similar observation holds for the mapping step, which can be instantiated using a variety of simple statistical tools [11]. By contrast, if one is not in the worst case adversary setting, the dimensionality reduction problem may become hard with no optimal solutions (in the context of higher-order and multivariate attacks [6]).

**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 (for symmetric encryption), 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, the power of a test, the number of traces, the effect size and the trace variance all play off each other. Setup manipulations may enable to increase the effect size and/or 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.

Whitnall and Oswald [56] 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. They show that 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], or tests that rely on preprocessing to make higher order leaks visible via first order statistics [46] 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 [4].

### 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?).

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

**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) [46]. This is for example possible for the two additive shares of the ANSSI software implementation analysed by Bronchain and Standaert in [5]. For high security orders, this exhaustive approach remains expensive and may require considering security margins [26].

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 [21, 30, 31, 7]. 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. Summarising, modelling is challenging and well understood techniques can often only be utilised by worst case adversaries.

**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).

**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. In this context, the information about different parts of the target secret are first combined in a maximum likelihood manner (which is optimal [9]). 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 provides tight bounds, see for example [15, 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 [27].

Advanced approaches include the algebraic (resp., analytical) attacks that target the secret key at once, as for example considered in [44] (resp., [54]) in the context of block ciphers, or in [42] for asymmetric cryptography. These attacks are in general more difficult to mount and to evaluate, due to their higher computational cost and sensitivity to various inherently heuristic parameters (e.g., to deal with cycles in the circuit graphs) [18, 17]. It implies risks of security overstatements whenever such attacks provide a significant gain over the simpler D&C and E&P ones.

The different depths 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.

## 5 Summary

An informal summary of the state-of-the-art solutions that can be used is given in Table 1. As illustrated by the colour code (red signals most uncertainty, followed by orange, yellow and green which indicates least uncertainty), some attack steps are quite well understood (in this context, 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 before.

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. This step is determining the noise level of the measurements, which is a key parameter for most algorithmic side-channel countermeasures (e.g., masking [8, 22], shuffling [20, 55], ...). Yet, this risk can and should be mitigated by the sound comparison of standard measurement

**Table 1.** Remaining uncertainty in evaluation steps.

| Attack steps | | CC | WCA | FIPS-SK | FIPS-PK |
|---|---|---|---|---|---|
| Measurement and preprocessing | | 🟠 | 🟠 | 🟠 | 🟠 |
| Detection and mapping | | 🟠 | 🟢 | 🔴 | 🔴 |
| Leakage Modeling (Profiling) | | 🔴 | 🟡 | - | - |
| Information extraction | | 🟢 | 🟢 | - | - |
| Information Processing | Simple (D&C, E&P) | 🟢 | 🟢 | - | 🟢 |
| | Advanced (analytical) | 🟠 | 🟠 | - | - |
| Overall | | 🟠 | 🟡 | 🔴 | 🔴 |

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 sub-optimalities in these attacks is generally limited and can be captured by small security margins as well.

## 6 Conclusions

Based on the previous summary we conclude 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 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.

Because leakage modelling is not within the scope of FIPS/ISO 17825, the lack thereof implies that any resulting evaluation only provides very loose guarantees.

A key difference between the CC approach and backwards evaluations (the approach that considers the worst case adversary first) is that in a backwards evaluation it is much more likely that simpler tools can be deployed during modelling and this lowers the risk of incorrectly estimating the true security level of a product (it also implies less guesswork and therefore faster/cheaper evaluations).

Our research suggest 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/modelling phase. The reason for this is that only in this setting, can we utilise 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 in relation to "best methods".

## References


1. M. Azouaoui, R. Poussier, and F. Standaert. Fast side-channel security evaluation of ECC implementations - shortcut formulas for horizontal side-channel attacks against ECSM with the montgomery ladder. In I. Polian and M. Stöttinger, editors, *Constructive Side-Channel Analysis and Secure Design - 10th International Workshop, COSADE 2019, Darmstadt, Germany, April 3-5, 2019, Proceedings*, volume 11421 of *Lecture Notes in Computer Science*, pages 25–42. Springer, 2019.
2. R. Benadjila, L. Khati, E. Prouff, and A. Thillard. https://github.com/ANSSI-FR/SecAESSTM32.
3. B. Bilgin and J. Fischer, editors. *Smart Card Research and Advanced Applications, 17th International Conference, CARDIS 2018, Montpellier, France, November 12-14, 2018, Revised Selected Papers*, volume 11389 of *Lecture Notes in Computer Science*. Springer, 2019.
4. O. Bronchain, T. Schneider, and F. Standaert. Multi-tuple leakage detection and the dependent signal issue. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):318–345, 2019.
5. O. Bronchain and F. Standaert. Side-channel countermeasures' dissection and the limits of closed source security evaluations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(2):1–25, 2020.
6. E. Cagli, C. Dumas, and E. Prouff. Kernel discriminant analysis for information extraction in the presence of masking. In K. Lemke-Rust and M. Tunstall, editors, *Smart Card Research and Advanced Applications - 15th International Conference, CARDIS 2016, Cannes, France, November 7-9, 2016, Revised Selected Papers*, volume 10146 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 2016.
7. E. Cagli, C. Dumas, and E. Prouff. Convolutional neural networks with data augmentation against jitter-based countermeasures - profiling attacks without preprocessing. In Fischer and Homma [13], pages 45–68.
8. S. Chari, J. R. Rao, and P. Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In M. J. Wiener editor, Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412, Springer, 1999.
9. S. Chari, J. R. Rao, and P. Rohatgi. Template attacks. In B. S. K. Jr., Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.
10. 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(4):1263–1297, 2019.
11. F. Durvaux and F. Standaert. From improved leakage detection to the detection of points of interests in leakage traces. In M. Fischlin and J. Coron, editors, *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*, volume 9665 of *Lecture Notes in Computer Science*, pages 240–262. Springer, 2016.

12. T. Eisenbarth, C. Paar, and B. Weghenkel. Building a side channel based disassembler. *Trans. Comput. Sci.*, 10:78–99, 2010.
13. W. Fischer and N. Homma, editors. *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*. Springer, 2017.
14. G. Fumaroli, A. Martinelli, E. Prouff, and M. Rivain. Affine masking against higher-order side channel analysis. In A. Biryukov, G. Gong, and D. R. Stinson, editors, *Selected Areas in Cryptography - 17th International Workshop, SAC 2010, Waterloo, Ontario, Canada, August 12-13, 2010, Revised Selected Papers*, volume 6544 of *Lecture Notes in Computer Science*, pages 262–280. Springer, 2010.
15. C. Glowacz, V. Grosso, R. Poussier, J. Schüth, and F. Standaert. Simpler and more efficient rank estimation for side-channel security assessment. In G. Leander, editor, *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, volume 9054 of *Lecture Notes in Computer Science*, pages 117–129. Springer, 2015.
16. 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.
17. 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 [3], pages 18–34.
18. V. Grosso and F. Standaert. Asca, SASCA and DPA with enumeration: Which one beats the other and when? In Iwata and Cheon [25], pages 291–312.
19. S. Guilley, H. Maghrebi, Y. Souissi, L. Sauvage, and J. Danger. Quantifying the quality of side channel acquisitions. *COSADE, February*, 2011.
20. C. Herbst, E. Oswald, and S. Mangard. An AES smart card implementation resistant to power analysis attacks. In J. Zhou, M. Yung, and F. Bao, editors, *Applied Cryptography and Network Security, 4th International Conference, ACNS 2006, Singapore, June 6-9, 2006, Proceedings*, volume 3989 of *Lecture Notes in Computer Science*, pages 239–252, 2006.
21. A. Heuser and M. Zohner. Intelligent machine homicide - breaking cryptographic devices using support vector machines. In W. Schindler and S. A. Huss, editors, *Constructive Side-Channel Analysis and Secure Design - Third International Workshop, COSADE 2012, Darmstadt, Germany, May 3-4, 2012. Proceedings*, volume 7275 of *Lecture Notes in Computer Science*, pages 249–264. Springer, 2012.
22. Y. Ishai, A. Sahai, D. A. Wagner. Private Circuits: Securing Hardware against Probing Attacks. In D. Boneh editor, Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings, volume 2729, of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.
23. ISO/IEC JTC 1/SC 27. ISO/IEC 15408-1: Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model. International Organization for Standardization, Geneva, CH, 2009.
24. ISO/IEC JTC 1/SC 27. ISO/IEC 17825: Information technology – Security techniques – Testing methods for the mitigation of non-invasive attack classes against cryptographic modules. International Organization for Standardization, Geneva, CH, 2016.
25. T. Iwata and J. H. Cheon, editors. *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*, volume 9453 of *Lecture Notes in Computer Science*. Springer, 2015.

26. A. Journault and F. Standaert. Very high order masking: Efficient implementation and security evaluation. In Fischer and Homma [13], pages 623–643.
27. T. Lange, C. van Vredendaal, and M. Wakker. Kangaroos in side-channel attacks. In M. Joye and A. Moradi, editors, *Smart Card Research and Advanced Applications - 13th International Conference, CARDIS 2014, Paris, France, November 5-7, 2014. Revised Selected Papers*, volume 8968 of *Lecture Notes in Computer Science*, pages 104–121. Springer, 2014.
28. K. Lemke-Rust and C. Paar. Gaussian mixture models for higher-order side channel analysis. In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 14–27. Springer, 2007.
29. L. Lerman and O. Markowitch. Efficient profiled attacks on masking schemes. *IEEE Trans. Information Forensics and Security*, 14(6):1445–1454, 2019.
30. L. Lerman, S. F. Medeiros, G. Bontempi, and O. Markowitch. A machine learning approach against a masked AES. In A. Francillon and P. Rohatgi, editors, *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, volume 8419 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2013.
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 S. Mangard and A. Y. Poschmann, editors, *Constructive Side-Channel Analysis and Secure Design - 6th International Workshop, COSADE 2015, Berlin, Germany, April 13-14, 2015. Revised Selected Papers*, volume 9064 of *Lecture Notes in Computer Science*, pages 20–33. Springer, 2015.
32. S. Mangard. Hardware countermeasures against DPA ? A statistical analysis of their effectiveness. In T. Okamoto, editor, *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004.
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 N. P. Smart, editor, *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, volume 10808 of *Lecture Notes in Computer Science*, pages 394–412. Springer, 2018.
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 [25], pages 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 K. Sako and P. Sarkar, editors, *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*, volume 8269 of *Lecture Notes in Computer Science*, pages 486–505. Springer, 2013.
36. National Institute of Standards and Technology. NIST FIPS 140-3. Information Technology Laboratory, NIST, Gaithersburg, MD 20899-8900
37. P. Q. Nguyen and I. E. Shparlinski. The insecurity of the elliptic curve digital signature algorithm with partially known nonces. *Des. Codes Cryptogr.*, 30(2):201–217, 2003.
38. D. Oswald and C. Paar. Improving side-channel analysis with optimal linear transforms. In S. Mangard, editor, *Smart Card Research and Advanced Applications -*

*11th International Conference, CARDIS 2012, Graz, Austria, November 28-30, 2012, Revised Selected Papers*, volume 7771 of *Lecture Notes in Computer Science*, pages 219–233. Springer, 2012.
39. R. Poussier, F. Standaert, and V. Grosso. Simple key enumeration (and rank estimation) using histograms: An integrated approach. In B. Gierlichs and A. Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 61–81. Springer, 2016.
40. R. Poussier, Y. Zhou, and F. Standaert. A systematic approach to the side-channel analysis of ECC implementations with worst-case horizontal attacks. In Fischer and Homma [13], pages 534–554.
41. S. M. D. Pozo and F. Standaert. Blind source separation from single measurements using singular spectrum analysis. In T. Güneysu and H. Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 42–59. Springer, 2015.
42. R. Primas, P. Pessl, and S. Mangard. Single-trace side-channel attacks on masked lattice-based encryption. In Fischer and Homma [13], pages 513–533.
43. E. Prouff, M. Rivain, and R. Bevan. Statistical analysis of second order differential power analysis. *IACR Cryptology ePrint Archive*, 2010:646, 2010.
44. M. Renauld, F. Standaert, and N. Veyrat-Charvillon. Algebraic side-channel attacks on the AES: why time also matters in DPA. In C. Clavier and K. Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, volume 5747 of *Lecture Notes in Computer Science*, pages 97–111. Springer, 2009.
45. W. Schindler, K. Lemke, and C. Paar. A stochastic model for differential side channel cryptanalysis. In J. R. Rao and B. Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.
46. T. Schneider and A. Moradi. Leakage assessment methodology - extended version. *J. Cryptographic Engineering*, 6(2):85–99, 2016.
47. S. Skorobogatov. Synchronization method for SCA and fault attacks. *J. Cryptographic Engineering*, 1(1):71–77, 2011.
48. SOG-IS. Application of attack potential to smartcards and similar devices, 2019.
49. SOG-IS. Attack methods for smartcards and similar devices, 2020.
50. F. Standaert. How (not) to use welch's t-test in side-channel security evaluations. In Bilgin and Fischer [3], pages 65–79.
51. F. Standaert, T. Malkin, and M. Yung. A unified framework for the analysis of side-channel key recovery attacks. In A. Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.
52. F. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper, and S. Mangard. The world is not enough: Another look on second-order DPA. In M. Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.

53. J. G. J. van Woudenberg, M. F. Witteman, and B. Bakker. Improving differential power analysis by elastic alignment. In A. Kiayias, editor, *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, volume 6558 of *Lecture Notes in Computer Science*, pages 104–119. Springer, 2011.
54. N. Veyrat-Charvillon, B. Gérard, and F. Standaert. Soft analytical side-channel attacks. In P. Sarkar and T. Iwata, editors, *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*, volume 8873 of *Lecture Notes in Computer Science*, pages 282–296. Springer, 2014.
55. N. Veyrat-Charvillon, M. Medwed, S. Kerckhof, and F. Standaert. Shuffling against side-channel attacks: A comprehensive study with cautionary note. In X. Wang and K. Sako, editors, *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*, volume 7658 of *Lecture Notes in Computer Science*, pages 740–757. Springer, 2012.
56. 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 S. D. Galbraith and S. Moriai, editors, *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*, volume 11923 of *Lecture Notes in Computer Science*, pages 256–284. Springer, 2019.