REASSURE

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

REASSURE

Deliverable D1.3

White Paper on Evaluation Strategies for AES and ECC

Editors:	M. Azouaoui (NXP) and F.-X. Standaert (UCL)
Deliverable nature:	R
Dissemination level: (Confidentiality)	PU
Delivery date:	March 2020
Version:	1.0
Total number of pages:	19
Keywords:	Evaluation, Framework, Side-channel analysis, AES, ECC

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

This deliverable is a white paper describing side-channel evaluation strategies of cryptographic implementations, particularly for AES and ECC. Through this paper, the REASSURE consortium provides general guidance and directions to improve how cryptographic implementations are evaluated both practically and soundly.

First, we begin by describing our alternative structured evaluation. It is a backwards approach based on first defining a worst-case adversary and relaxing particular capabilities to reach the best practical attack possible. This first step of the evaluation stems from the fact that defining a worst-case attack strategy is commonly easier than defining the best practical attack strategy. This allows for a sound and well-defined starting point for any implementation under investigation. Accordingly, we then detail the different capabilities that must be taken into account and described during an evaluation and additionally discussed when arguing upon the feasability of attack strategies with relaxed capabilities. Next, we describe the three main evaluation steps following the Detect-Map-Exploit framework.

We illustrate the proposed evaluation strategy with three concrete examples: an AES implementation protected with combined affine masking and shuffled execution, an unprotected ECC implementation and the ECC point randomization countermeasure.

We connect our approach to current evaluation strategies such as the BSI ECC evaluation guidelines and the CC scheme and describe how it can be used as a refinement of the latter to yield more efficient and representative evaluations. We finally discuss how our approach adresses the gap between concrete worst-case security and current evaluation processes.

List of authors

Company	Author
NXP	M. Azouaoui
UCL	F.-X. Standaert
UCL	F. Koeune
UNI-KLU	E. Oswald
SGDSN	E. Jaulmes

Other contributions gratefully received from

Company	Author
IDM	N. Debande
Riscure	I. Buhan

Revision history

Revision number	Date	Comment
1.0	March 2020	First version of document

Contents

AES Advanced Encryption Standard

ANSSI Agence Nationale de la SÃ©curitÃ© des SystÃ¨mes d'Information

BSI Bundesamt für Sicherheit in der Informationstechnik

CC Common Criteria

D&C Divide & Conquer

E&P Extend & Prune

ECC Elliptic Curve Cryptography

ECDH Elliptic Curve Diffie Hellman

ECDSA Elliptic Curve Digital Signature

ECSM Elliptic Curve Scalar Multiplication

FIPS Federal Information Processing Standard

IP Intellectual Property

ISCI International Security Certification Initiative

JHAS JIL Hardware Related Attack Subgroup

LIM Long Integer Multiplication

PCA Principal Component Analysis

POI Point Of Interest

SASCA Soft Analytical Side-Channel Attacks

SNR Signal-to-Noise Ratio

TLVA Test Vector Leakage Assessment

1 Introduction: chasing a moving target

Since their apparition in the late 1990s [31], side-channel attacks have considerably reshaped the understanding of cryptographic implementations and led to various (evaluation and design) challenges. Both from the evaluation and the design points-of-view, one central difficulty raised by these attacks is the need to capture physical quantities. That is, while modern cryptography generally relies on computational security assumptions [30] (leading to reasonably well understood security guarantees and predictions), physical attacks ultimately require the (usually statistical) modeling of physical objects which depend on many hard(er) to quantify assumptions such as the adversary's knowledge of the target implementations, the quality of his measurement setup, ...

The state-of-the-art solutions for side-channel security testing typically deal with this physical modeling issue in two different manners. First, *conformance-style* testing aims at developing cost-effective procedures to verify minimum properties for side-channel secure implementations. Examples include the FIPS 140-3 standard [15], and the popular TVLA methodology [21]. Second, *evaluation-style* testing rather aims at defining a common framework for evaluating implementations by analyzing all state-of-the-art attack strategies [54] – the latter being typically developed in research venues such as the IACR's CHES, EUROCRYPT, CRYPTO, ASIACRYPT [27]. This is for example the approach of the ISCI-WG2 working group (also known as JHAS) of the International Security Certification Initiative (ISCI), which brings together stakeholders from every aspect of smart card security evaluation to define and maintain the state of the art in potential attacks against smart security devices.

In practice, this state of affairs implies that there is a significant effort required to keep track of all the existing attack vectors, and to come to a shared understanding of what may be the "best practical attacks" among the ones that are considered relevant. We hypothesize that there exist attacks that were initially considered as a lesser threat, but over time they were assessed differently. This and the fact that new attack vectors keep getting discovered and existing attack vectors get improved means that a lot of time is needed to re-establish a common view on what are such "best practical attacks" [57], especially since practicality is in general hard to define and assess.

For instance, the document produced by the Bundesamt für Sicherheit in der Informationstechnik (BSI) [10] provides guidelines for security evaluators to test the resistance of ECC implementations against side-channel attacks with high attack potential according to version 3.1 of the Common Criteria (CC). However, albeit extremely useful and valuable, it is clearly stated in this document that it only lists a set of attacks and gives the baseline of minimum requirements that are by no means exhaustive. Concretely, despite having access to a document that lists several state-of-the-art attacks and publications, it is the responsability of the evaluator to keep their knowledge and expertise up to date. The BSI's disclaimers in [10] refer to the matter we point out in this paper. Particularly for ECC, each implementation can be unique since it is based not only on state-of-the-art publications but also on company IP (Intellectual Property). Based on the BSI document, the evaluator is required to comb through a significant portion of the side-channel literature and adapt existing attacks to suit the implementation under investigation. This implies a significant effort and certainly no full coverage or assurance that all possible attack vectors/vulnerabilities have been addressed.

In this paper, we suggest an alternative, possibly more effective, way to structure attack vectors and to conduct physical security evaluations. Our starting observation for this purpose is that while defining the "best practical attacks" is hard, it is however possible to first define a worst-case adversary and next argue why it may not apply to an implementation. Such a worst-case adversary will try to utilize multiple leaking intermediate variables, 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.

During an evaluation, a natural goal is therefore to come as close as possible to the worst-case adversary, by first granting him 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 targets that may occur separated in the time domain, and the predictable targets that occur within each clock cycle, (*ii*) attempt

characterization (potentially by using a biased trace set if documentation suggests when masks may leak). As a result, our proposal is 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. We next illustrate how this "backwards evaluation strategy" can be instantiated with symmetric (masked AES) and asymetric (ECC scalar multiplication and point randomization) cryptography case studies.

2 Adversary/evaluator's capabilities

We generally assume an adversary/evaluator who can access a cryptographic implementation and obtain physical leakages. For this purpose, an evaluation report should at least specify:

1. **Device preparation (and number of items).** Does the target device need to be prepared (e.g., depackaged) prior to the side-channel attack or can it be used non-invasively? In case of invasive attacks, how many items are needed for a successful preparation?
2. **Input control.** Does the adversary/evaluator know and/or control the algorithm's inputs (e.g., plaintexts) and outputs (e.g., ciphertexts) during the online phase of the attack?
3. **Profiling abilities.** In case profiled (aka supervised) attacks are considered, does the adversary/evaluator have access to devices for which he knows and/or controls the keys and (e.g., in case of protected implementations) the randomness generated by the implementation?
4. **Implementation knowledge.** Does the adversary/evaluator know the hardware and software source codes? And more generally, what does he know about the target?
5. **Setup cost.** What is the approximate cost of the hardware used to measure the leakages?

2.1 The worst-case adversary

We put forward the proposal to define the worst-case adversary as a starting point for an evaluation. Whilst this definition is implementation-dependent, we posit that it is considerably better defined than selecting a (subjectively) best attack vector 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) as well as over the secret parameters (keys, randomness). He 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).

In other words, the worst-case adversary is pushing the separation between the identification of the attack and its exploitation (both considered in CC evaluations, as briefly discussed in Section 6 and more detailed in the REASSURE Deliverable D1.5 `www.reassure.eu`) 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. He 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.

3 Evaluation steps

Concrete side-channel attacks can be viewed as a combination of steps [37]. While the description of these steps is usually informal and the border between them is flexible, we next make a proposal of attack based on the definition of three main steps, with as main goal to delineate the descriptions that would be required in a typical evaluation report. We do not enforce any specific tools to perform any of the steps, nor metrics to quantify the outcome of these tools, since multiple options are usually possible (and sometimes equivalent [38]). By contrast, we require that an evaluation provides propositions of tools and (when applicable) metrics for the different steps (we will give examples in the next sections).
The different steps of our side-channel evaluation strategy are described as follows:

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 analyzes 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) modeling 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).

We note that some tools may be used in different steps. For example, dimensionality reduction with Principal Component Analysis (PCA) can be viewed as a pre-processing step if applied on raw measurements and as a detection and mapping step if applied to average measurements [2].

4 The "backwards" evaluation approach

Performing a backwards side-channel security evaluation is based on first considering a worst-case adversary that we defined as having full knowledge of the code, full control over inputs, key material and randomness and additionally full (yet realistic) profiling abilities. Then a combination of tools to perform the steps mentioned in Section 3 in a worst-case manner is proposed by the evaluator, without limiting the adversarial capabilities. The first step of the evaluation is to demonstrate an attack performed by the worst-case adversary.

Concretely, it may be that a device does not allow for a such a worst-case adversary. For example, in case of a hardware implementation, access to randomness may require to change the target in such a way that its leakage characteristics would change. Thus if some of the previously defined abilities cannot be fulfilled for the target under investigation, following the backwards approach, they are progressively relaxed in order to

reach a more realistic attack that can be demonstrated.

Whenever applicable, an evaluation metric can additionally be estimated in order to reflect the attack (steps) complexity. Based on them, in the following parts of the evaluation, since the adversary's capabilities are redefined, the impact of the different assumptions on the attack complexity is discussed. By starting from the fixed worst-case adversary and documenting the gradual relaxation of adversarial assumptions it is possible to assess the gap between the worst-case adversary and a practical adversary.

5 Case studies

We now illustrate the proposed backwards approach with three concrete case studies: a masked AES implementation proposed by the French ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information) [5], recently analyzed in [9], an (unprotected) ECC scalar multiplication analyzed in [46, 4] and the ECC point randomization countermeasure analyzed in [3].

5.1 Masked AES implementation

5.1.1 Towards a worst-case attack.

The ANSSI implementation that we analyze is a protected implementation combining additive and multiplicative secret sharing into an affine masking scheme [18], which is additionally mixed with a shuffled execution [26]. It is running on an ARM Cortex-M4 architecture. Preliminary leakage assessments did not reveal data dependencies with up to 100,000 measurements.

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 of their worst-case attack can be detailed as follows:

Regarding the 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) [36] of the multiplicative mask (which is the most critical to recover in a worst-case attack). No additional preproecessing (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.

Regarding 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 PCA was additionally performed (which allowed recovering this mask in full).

Regarding modeling, all the randomized target intermediate variables are modeled with Gaussian mixtures as in [53, 56]. Thanks to the knowledge of the randomness during profiling, this was done straightforwardly by estimating first-order (sometimes multivariate) Gaussian templates [12].

Regarding information extraction, Bronchain and Standaert considered countermeasures' dissection. 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 [45]. As a result, the best attack was able to reduce the 128-bit key rank below 2^{32} with less than 2,000 measurements.

5.1.2 Relaxing capabilities.

Compared to the leakage assessment in [5], 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 [9], 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 [9]. 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 [14] 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 [35], but the profiling cost is significantly higher than in the known randomness case (another solution is [33]). Alternatively, one can attack using a non-profiled higher-order side-channel attack [47]. 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 [52]. 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 [13, 11]. 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 $> \frac{100{,}000}{2{,}000} = 50$.

5.2 ECC scalar multiplication

5.2.1 Towards a worst-case attack.

The ECSM implementation analyzed in [46, 4] 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 adversary is assumed to have full control of the inputs and full profiling capabilities. The generic evaluation framework designed by Poussier et al. [46] is horizontal 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, are described below:

Regarding the measurement setup, the voltage variation was monitored using a 4.7 Ω 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.

Regarding leakage detection and mapping, POIs corresponding to target intermediate values are identified using classical selection techniques such as correlation [13] or SNR based ones.

Once the time locations of all the target intermediates are found, they can be modeled using classical Gaussian templates [12], but such an exhaustive profiling is measurement intensive. As a result, Poussier et al. rather use a linear regression based approach with a 32-bit basis [51], which significantly speeds up the modeling phase of the 32-bit target registers.

Regarding 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 for 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 [32], 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 [40]. 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.

5.2.2 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 is 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 [46] 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. [4] 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.

5.3 ECC point randomization

5.3.1 Towards a worst-case attack.

The previous evaluation assumes that the attacker has knowledge of the ECSM input point. If the implementation is protected with point randomization, this is no longer possible. However it is possible for the attacker to additionally target the point randomization procedure. We analyzed the security of a point randomization countermeasure implemented using independent Long Integer Multiplication (LIM) and modular reduction on an AVR ATmega328p microcontroller [3].

For this purpose, we first considered advanced Soft Analytical Side-Channel Attacks (SASCA) [55] to potentially exploit all leakage samples of the target implementation (i.e., not only the ones that can be easily predicted). In order to gauge the impact of such techniques in this ECC point randomization context, we then compared the advanced SASCA with a classical Divide-and-Conquer (D&C) attack that only targets enumerable parts of the secret independently. Interestingly, these results show that in this single-trace attack context, the gain of SASCA over D&C attacks is negligible. This indicates that D&C attacks on the target point randomization countermeasure are close to worst-case and thus can be used for efficient and practical high security evaluations.

The evaluation steps apply independently and in the same way to all the limbs of the secret used to randomize the point, in a similar way as the previous horizontal attack on ECSM:

Regarding the measurement setup, the target microcontroller is mounted on an Arduino UNO board running at 16MHz. Using a custom probe, the traces are captured on a PicoScope5244D at a sampling rate of 125MHz. The traces are preprocessed using amplitude demodulation.

Regarding leakage detection and mapping, POIs relating to the multiplication of each limb are identified using the correlation test [13]. These time samples are then combined using PCA [2].

The modeling phase consists of estimating multivariate Gaussian templates [12] in order to characterize the leakage of each possible value of the limb. The information is then extracted by simply evaluating the probability of each limb value using a single trace.

As for post-processing, typically, the most likely values of the randomized point can be enumerated and used to attack the reminder of the ECSM (as in Section 5.2). Alternatively, rank estimation [45] can be used to assess the computational effort to reach the randomized point.

5.3.2 Relaxing capabilities.

Once again, the main adversarial capability needed to perform a worst-case attack (given that there is no high-order statistical distribution to estimate) is the knowledge of the code. The situation in this respect is similar to the ECSM case study (and different from the masked AES case study). The main reason for this is the preliminary observation that in the single-trace attack considered, SASCA (for which detecting POIs without code knowledge can be challenging) and D&C attacks (for which simple correlation based or SNR tests can be used for this purpose) have similar efficiencies. Concretely, a point randomization consists of one or two field multiplications. Hence, an attacker can predict intermediates targetable by the D&C strategy and identify them

on the traces (if he is able to control the input to the randomization procedure) – this would be difficult for the deep target intermediates exploited by a SASCA (since they cannot be enumerated).

6 Connection with the CC approach

In the CC approach, attacks are evaluated by rating the difficulty to identify the attack vector and the difficulty to perform the attack once identified.[1] This difficulty is measured in terms of elapsed time, expertise, knowledge of the target, number of targets (e.g., for depackaging) and equipment. While such an approach is in general sensible, the backwards evaluation approach described in this note suggests one possible refinement. Namely, to complete an attack potential calculation in the CC approach, the points for identification and exploitation have to be added as both phases together constitute the complete attack. However, in practice, most of the identification effort only needs to be performed a single time, while the exploitation has to be repeated for every target. As a result, it may be interesting to report not only the sum of the points, but also the separate scores obtained for these two (identification and exploitation) parts. While not fundamentally changing the CC approach, it would provide users with a better understanding of what the CC evaluation implies in terms of worst-case security. Indeed, worst-case security is determined by the exploitation points (only, or at least significantly more than the identification points).

7 Connection to BSI's minimum requirements for ECC evaluations

Regarding our first ECC case study, the most recent version of the BSI document [10] lists some profiled attacks on unprotected ECC implementations exploiting classical multivariate Gaussian templates [12], and dates back to 2016. Hence, it does not include the Poussier et al. attack [46] published in the proceedings of the IACR CHES 2017 conference. However, the Poussier et al. attack is only one example among recently published attacks that should complement this document and additionally be considered by evaluators as a worst-case evaluation approach, since it allows possibly exploiting the leakage of all intermediate values (and can be applied even when some capabilities are relaxed, as described in Section 5.2.2). If this kind of worst-case attack is considered, it also annihilates the need and the effort required to mount a plethora of less powerful attacks requiring the same level of capabilities.

Regarding our second ECC case study of the point randomization countermeasure, while the BSI document lists the attack literature targeting the scalar blinding countermeasure for instance, the new attack vector of the point randomization countermeasure is not described. In general, this highlights the fact that despite evaluators spending effort to maintain a comprehensive list of attacks, some attack vectors might still remain unexplored, if worst-case adversaries exploiting all possible sources of information leakage are not considered.

8 Guidance and metrics

The general guidance we recommend for a sound evaluation strategy is to start from the worst-case adversary as defined in Sec. 2.1 and then relax his capabilities while assessing the impact of this relaxation on the attack complexity. In the current state of the art, the possibility to define more specific methods, as well as the confidence we have in their reliability, is not the same for each of the attacks/evaluation steps. The REASSURE Deliverable D1.5 (`www.reassure.eu`) discusses this issue in more details.

- **Measurement and preprocessing** mostly depends on engineering expertise to filter the noise, resynchronize the traces , etc., but their effectiveness is typically application-dependent. Quality metrics here are the SNR for univariate evaluations and information theoretic metrics for multivariate evaluations.
- **Leakage detection and mapping** can rely on a well-established theory in statistical hypothesis testing. Quality metrics are provided by well-known formulas relating sample size, significance level and risks

[1] `https://www.sogis.eu/documents/cc/domains/sc/JIL-Application-of-Attack-Potential-to-Smartcards-v3-0.pdf`.

of (Type I and Type II) errors. When moving from generic detection to specific mapping, other metrics like the Signal-to-Noise Ratio and correlation metrics can also be considered.

- **Leakage modelling** is probably the most complex step, since the 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. In case of probabilistic modeling, information theoretic metrics are usually the most convenient. For other types of modeling tools (e.g., based on machine learning), the only option is to directly evaluate the attack success (possibly on subparts of the key).
- **Information extraction** is in fact quite straightforward once the other steps have been defined. No specific metric is needed here, as it will be encompassed when extracted information will be fed to the exploitation phase.
- **Information processing** is a step which can be quite straightforward or require a deep level of expertise, depending on the required attack approach. 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, evaluation is based on maximum likelihood manner, which is optimal [12]. Advanced approaches include the algebraic (resp., analytical) attacks that target the secret key at once. 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) [24, 23]. Strictly, the backwards evaluation approach is calling for the advanced approaches. Yet, since they imply some heuristic parameters, our current suggestion (which may evolve with the state of the art) is to analyze both types of approaches in order to estimate their respective effectiveness in practice. Evaluating both approaches can also highlight risks of security overstatements in the cases where advanced attacks provide a significant gain over the simpler ones (that can be more formally analyzed and bounded). The quality metric in this respect must take into account the computing power at the disposal of the adversary to exploit the result of the side-channel attack. In this sense, the key rank (in the symmetric case) or key ranking stategies combined with classical cryptanalysis techniques [32] and success bounds on lattice methods [40] (in the asymmetric case) are the most sensible approaches.

9 Conclusion: mind the gap

The proposed backwards approach for evaluating cryptographic products suggests that a gap may exist between the worst-case security of an implementation and its security as assessed by current evaluation practices. Such a gap typically happens when the cost for identifying an attack is significantly higher than the cost for performing the attack. The evaluation of the masked AES implementation in Section 5.1 is an example of this situation. Concretely, this gap is amplified whenever the worst-case attacks taking advantage of both multivariate and higher-order distributions are significantly more powerful than attacks that are either multivariate only or higher-order only. In the context of masked block ciphers, this gap therefore increases with the number of shares and security level [25]. In some other cases, where the optimal attack does not require both multivariate and higher-order approaches, like in our ECC case studies, this gap may remain small.

In our investigations, the most important adversarial capabilities are the implementation knowledge and the access to the countermeasures' randomness. As summarized in Table 1, removing such capabilities in the AES case is critical. While a lack of implementation knowledge can sometimes be compensated by some high-level reverse engineering based on detection and mapping (which however at least require a good leakage model), profiling a masked implementation in a black box manner is a hard task, as discussed in Section 5.1. By contrast, removing these capabilities is less critical for the investigated ECC case studies. This is because in this context, most POIs are expected to be detectable with standard profiling methods, and the few missing points are not expected to lead to a significant change of the attack complexity.

We note that other adversarial capabilities could lead to important gaps. For example, if precise electromagnetic profiling requires a difficult depackaging, it may be that access to samples becomes critical.

Overall, these observations confirm the variety of the challenges raised by side-channel security evaluations, and that the backwards approach is a promising candidate for more representative security evaluations.

Table 1: REASSURE evaluation strategy: examples of security gaps.

Target	Capabilities	Gap
Masked & shuffled AES	Implementation knowledge	🟠
	Countermeasures' randomness	🔴
ECC scalar multiplication	Implementation knowledge	🟡
	Countermeasures' randomness	🟡
ECC point randomization	Implementation knowledge	🟡
	Countermeasures' randomness	🟡

References

[1] Masayuki 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*. Springer, 2010.

[2] Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template attacks in principal subspaces. In Goubin and Matsui [22], pages 1–14.

[3] Melissa Azouaoui, François Durvaux, Kostas Papagiannopoulos, Romain Poussier, François-Xavier Standaert, and Vincent Verneuil. On the Worst-Case Side-Channel Security of ECC Point Randomization in Embedded Devices. *under submission*, 2020.

[4] Melissa Azouaoui, Romain Poussier, and François-Xavier Standaert. Fast side-channel security evaluation of ECC implementations - shortcut formulas for horizontal side-channel attacks against ECSM with the montgomery ladder. In Polian and Stöttinger [44], pages 25–42.

[5] Ryad Benadjila, Louiza Khati, Emmanuel Prouff, and Adrian Thillard. `https://github.com/ANSSI-FR/SecAESSTM32`.

[6] Begül Bilgin and Jean-Bernard 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.

[7] Alex Biryukov, Guang Gong, and Douglas 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*. Springer, 2011.

[8] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Joye and Quisquater [28], pages 16–29.

[9] Olivier Bronchain and François-Xavier Standaert. Side-channel countermeasures' dissection and the limits of closed source security evaluations. *IACR Cryptology ePrint Archive*, 2019:1008, 2019.

[10] Bundesamt für Sicherheit in der Informationstechnik. Minimum Requirements for Evaluating Side-Channel Attack Resistance of Elliptic Curve Implementations. `https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_46_ECCGuide_e_pdf`, 2016.

[11] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Kernel discriminant analysis for information extraction in the presence of masking. In Lemke-Rust and Tunstall [34], pages 1–22.

[12] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Jr. et al. [29], pages 13–28.

[13] François Durvaux and François-Xavier Standaert. From improved leakage detection to the detection of points of interests in leakage traces. In Fischlin and Coron [17], pages 240–262.

[14] Thomas Eisenbarth, Christof Paar, and Björn Weghenkel. Building a side channel based disassembler. In *Trans. Comput. Sci.* [19], pages 78–99.

[15] FIPS 1403. Security requirements for cryptographic modules, 2015. http://csrc.nist.gov/groups/ST/FIPS140_3/.

[16] Wieland Fischer and Naofumi 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.

[17] Marc Fischlin and Jean-Sébastien 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*. Springer, 2016.

[18] Guillaume Fumaroli, Ange Martinelli, Emmanuel Prouff, and Matthieu Rivain. Affine masking against higher-order side channel analysis. In Biryukov et al. [7], pages 262–280.

[19] Marina L. Gavrilova, Chih Jeng Kenneth Tan, and Edward D. Moreno, editors. *Transactions on Computational Science X - Special Issue on Security in Computing, Part I*, volume 6340 of *Lecture Notes in Computer Science*. Springer, 2010.

[20] Benedikt Gierlichs and Axel 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*. Springer, 2016.

[21] Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. A testing methodology for side-channel resistance validation. NIST non-invasive attack testing workshop, 2011. http://csrc.nist.gov/news_events/non-invasive-attack-testing-workshop/papers/08_Goodwill.pdf.

[22] Louis Goubin and Mitsuru Matsui, editors. *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, volume 4249 of *Lecture Notes in Computer Science*. Springer, 2006.

[23] Joey Green, Arnab Roy, and Elisabeth Oswald. A systematic study of the impact of graphical models on inference-based attacks on AES. In Begül Bilgin and Jean-Bernard 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*, pages 18–34. Springer, 2018.

[24] Vincent Grosso and François-Xavier Standaert. Asca, SASCA and DPA with enumeration: Which one beats the other and when? In Tetsu Iwata and Jung Hee 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*, pages 291–312. Springer, 2015.

[25] Vincent Grosso and François-Xavier Standaert. Masking proofs are tight and how to exploit it in security evaluations. In Nielsen and Rijmen [41], pages 385–412.

[26] Christoph Herbst, Elisabeth Oswald, and Stefan Mangard. An AES smart card implementation resistant to power analysis attacks. In Zhou et al. [60], pages 239–252.

[27] International Association for Cryptologic Research. https://www.iacr.org/.

[28] Marc Joye and Jean-Jacques Quisquater, editors. *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*. Springer, 2004.

[29] Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors. *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*. Springer, 2003.

[30] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007.

[31] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Wiener [59], pages 388–397.

[32] Tanja Lange, Christine van Vredendaal, and Marnix Wakker. Kangaroos in side-channel attacks. In Marc Joye and Amir 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.

[33] Kerstin Lemke-Rust and Christof Paar. Gaussian mixture models for higher-order side channel analysis. In Paillier and Verbauwhede [43], pages 14–27.

[34] Kerstin Lemke-Rust and Michael 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*. Springer, 2017.

[35] Liran Lerman and Olivier Markowitch. Efficient profiled attacks on masking schemes. *IEEE Trans. Information Forensics and Security*, 14(6):1445–1454, 2019.

[36] Stefan Mangard. Hardware countermeasures against DPA ? A statistical analysis of their effectiveness. In Okamoto [42], pages 222–235.

[37] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

[38] Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Information Security*, 5(2):100–110, 2011.

[39] Erick Nascimento and Lukasz Chmielewski. Applying horizontal clustering side-channel attacks on embedded ECC implementations. In Thomas Eisenbarth and Yannick Teglia, editors, *Smart Card Research and Advanced Applications - 16th International Conference, CARDIS 2017, Lugano, Switzerland, November 13-15, 2017, Revised Selected Papers*, volume 10728 of *Lecture Notes in Computer Science*, pages 213–231. Springer, 2017.

[40] Phong Q. Nguyen and Igor E. Shparlinski. The insecurity of the elliptic curve digital signature algorithm with partially known nonces. *Des. Codes Cryptogr.*, 30(2):201–217, 2003.

[41] Jesper Buus Nielsen and Vincent Rijmen, editors. *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*, volume 10821 of *Lecture Notes in Computer Science*. Springer, 2018.

[42] Tatsuaki Okamoto, editor. *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, volume 2964 of *Lecture Notes in Computer Science*. Springer, 2004.

[43] Pascal Paillier and Ingrid 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*. Springer, 2007.

[44] Ilia Polian and Marc 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*. Springer, 2019.

[45] Romain Poussier, François-Xavier Standaert, and Vincent Grosso. Simple key enumeration (and rank estimation) using histograms: An integrated approach. In Gierlichs and Poschmann [20], pages 61–81.

[46] Romain Poussier, Yuanyuan Zhou, and François-Xavier Standaert. A systematic approach to the side-channel analysis of ECC implementations with worst-case horizontal attacks. In Fischer and Homma [16], pages 534–554.

[47] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical analysis of second order differential power analysis. *IACR Cryptology ePrint Archive*, 2010:646, 2010.

[48] Josyula R. Rao and Berk 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*. Springer, 2005.

[49] REASSURE PROJECT. Towards principled side-channel security evaluations, 2017.

[50] Werner Schindler and Sorin 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*. Springer, 2012.

[51] Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In Rao and Sunar [48], pages 30–46.

[52] François-Xavier Standaert. How (not) to use welch's t-test in side-channel security evaluations. In Bilgin and Fischer [6], pages 65–79.

[53] François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The world is not enough: Another look on second-order DPA. In Abe [1], pages 112–129.

[54] The Common Criteria. `https://www.commoncriteriaportal.org/`.

[55] Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Soft analytical side-channel attacks. In Palash Sarkar and Tetsu 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.

[56] Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. Shuffling against side-channel attacks: A comprehensive study with cautionary note. In Wang and Sako [58], pages 740–757.

[57] Mathias Wagner. 700+ attacks published on smart cards: The need for a systematic counter strategy. In Schindler and Huss [50], pages 33–38.

[58] Xiaoyun Wang and Kazue 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*. Springer, 2012.

[59] Michael 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*. Springer, 1999.

[60] Jianying Zhou, Moti Yung, and Feng 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*, 2006.

