Fast Side-Channel Security Evaluation of ECC Implementations

Shortcut Formulas for Horizontal Side-channel Attacks against ECSM with the Montgomery ladder

Melissa Azouaoui[1,2], Romain Poussier[3],
François-Xavier Standaert[1]

[1] Université Catholique de Louvain
[2] NXP Semiconductors
[3] Temasek Laboratories, Nanyang Technological University, Singapore

Abstract. Horizontal attacks are a suitable tool to evaluate the (nearly) worst-case side-channel security level of ECC implementations, due to the fact that they allow extracting a large amount of information from physical observations. Motivated by the difficulty of mounting such attacks and inspired by evaluation strategies for the security of symmetric cryptography implementations, we derive shortcut formulas to estimate the success rate of horizontal differential power analysis attacks against ECSM implementations, for efficient side-channel security evaluations. We then discuss the additional leakage assumptions that we exploit for this purpose, and provide experimental confirmation that the proposed tools lead to good predictions of the attacks' success.

Keywords: Elliptic Curve Cryptography (ECC), side-channel attacks, side-channel security evaluations, Horizontal Differential Power Analysis (HDPA).

1 Introduction

Elliptic curve cryptography (ECC) relies on the intractability of the elliptic curve discrete logarithm problem. Due to the efficiency of elliptic curve based cryptosystems in comparison to other public-key cryptosystems such as RSA, they have been widely deployed in modern information systems, and thus are targeted by Side-Channel Attacks (SCAs). One of the most important ingredients of ECC protocols is the Elliptic Curve Scalar Multiplication (ECSM). As a result, various types of SCAs have been introduced against their implementations.

First, Simple Power Analysis (SPA) [17] exploits the fact that the sequence of operations depends on the secret scalar. A regular execution can thwart these attacks [15]. Next, Differential Power Analysis (DPA) [16, 8] recovers the secret scalar from multiple side-channel traces, and thus can be prevented by scalar randomization [8]. Template Attacks (TA) [6] have also been used to break ECC implementations [21]. They rely on the knowledge of the input point and can be thwarted by point randomization [8, 14]. Attacks against ECSM algorithms

additionally include Horizontal Collision Attacks (HCA) [5] which take advantage of the observation that for a certain scalar bit value, identical operands are manipulated at different instants of the execution. These attacks can be hindered by the shuffling countermeasure [18] or randomization techniques [14]. Finally, Horizontal DPA (HDPA) [7] exploits multiple time samples of one side-channel trace, as opposed to the classical vertical DPA described above. Besides, attacks against ECSM algorithms usually follow one out of two standard strategies: *divide-and-conquer* or *extend-and-prune*. In a divide-and-conquer attack, the bits of the scalar are recovered independently, while in an extend-and-prune attack they are recovered recursively.

We are particularly interested in HDPA following an extend-and-prune strategy. Attacks such as in [22] are powerful and suitable for (nearly) worst-case side-channel security assessments, since their horizontal nature allows extracting most of the information from a leakage trace. However, they are intricate to mount, due to the fact that they rely on the knowledge of the implementation, and the exploitation of many time samples from one single noisy side-channel trace. As a result, and inspired by evaluation strategies considered for implementation security in symmetric cryptography (e.g., [24, 11, 13]), we propose shortcut formulas and derive an efficient approximation of the Success Rate (SR) of an HDPA as function of the number of leaking registers exploited and the noise level of the implementation. For this purpose, we first describe our method and its underlying assumptions, and then confirm its practical relevance based on an experimental case study.

The rest of the paper is organized as follows. Section 2 introduces our notations and background on ECC and the extend-and-prune HDPA by Poussier et al. [22]. Section 3 explains the rationale behind our approach and the goal of our research. Section 4 details the efficient derivation of the success rate. Section 5 reports results from simulated experiments and Section 6 shows the relevance of the proposed approach on a real target.

2 Background

2.1 Notations

We use capital letters for random variables and small caps for their realizations. We use sans serif font for functions (e.g., F). We denote the conditional probability of a random variable A given B with $\Pr[A|B]$. We use $\mathcal{U}(\mathbb{F})$ to denote the uniform distribution over a field $\mathbb{F}$ and $\mathcal{N}(\mu, \sigma^2)$ to denote the Gaussian distribution with mean μ and variance σ^2. We use $\sim$ to denote that a random variable follows a given distribution (e.g., $A \sim \mathcal{N}(\mu, \sigma^2)$). We also denote by Φ the Cumulative Distribution Function (CDF) of the normal distribution.

2.2 Elliptic curve scalar multiplication

We denote by $\mathbb{F}_p$ a finite field of characteristic $p > 3$ and $\mathcal{E}(\mathbb{F}_p)$ the set of points $(x, y) \in \mathbb{F}_p^2$ that satisfy the elliptic curve with the Weierstrass equation:

$y^2 = x^3 + ax + b$ along with the point at infinity O. For a scalar $k \in \mathbb{F}_p$ we denote by $(k_0, k_1, ..., k_{n-1})$ its binary representation where k_0 is the most significant bit. For $P, Q \in \mathcal{E}(\mathbb{F}_p)$, $P + Q$ denotes the point addition, and kP the k-repeated addition $P + ... + P$, i.e., the ECSM. Elliptic curve cryptosystems (such as ECDH and ECDSA [23]) require to perform a scalar multiplication kP where k is a secret and P a public curve point. A popular method to implement ECSM securely consists in using the Montgomery ladder [15], shown in Algorithm 1. Its regular operation flow makes it naturally resistant against SPA [15].

Algorithm 1 Montgomery ladder

Input P , $k = (k_0, ..., k_{n-1})$
Output kP

1: $R_0 \leftarrow O$
2: $R_1 \leftarrow P$
3: **for** $i = 0$ to $n - 1$ **do**
4: $\quad R_{1-k_i} \leftarrow R_{1-k_i} + R_{k_i}$
5: $\quad R_{k_i} \leftarrow 2R_{k_i}$
6: **end for**
7: **return** R_0

2.3 Horizontal differential power analysis

As shown by Algorithm 1, the Montgomery ladder ECSM processes the bits of k iteratively, updating the internal state (R_0 and R_1) of the algorithm accordingly. At bit position i of k, the internal state depends on bits $\{0, ..., i\}$ of k. As a result, attacks against Montgomery ladder ECSM implementations are naturally performed by using an extend-and-prune strategy, and recovering the key bits in a recursive manner: the recovery of the i-th bit relies on the correct recovery of the previous bits $\{0, ..., i-1\}$ in order to make a hypothesis on the state [4, 7]. Following this strategy, the HDPA attack presented in [22] divides a constant time ECSM execution into a sequence of predictable operations at each abstraction level of the ECSM as shown in Figure 1. The last layer consists of $n \times N$ register multiplications, where N is the number of register multiplications required to process one scalar bit[1]. We consider the same reference Montgomery ladder ECSM as in [22] on a 32-bit device, using Jacobian coordinates and the point addition and doubling routines given in Appendix A on the NIST P-256 curve [23]. This implementation requires 25 field multiplications per scalar bit and they are performed with a Long Integer Multiplication (LIM) followed by a reduction. Knowing the i previous bits of k and the input point P, the attacker succeeds if he can infer the correct sequence of register values $(r_j)_{0 \leq j < N}$ out of the two possibilities for k_i. For efficiency reasons, the attack assumes the Independence of the Operations' Leakages (IOL) [13]. Besides, a leakage of the

[1] While we only consider register multiplications, the framework can be applied to any operation

form $l_j(r_j) = \delta_j(r_j) + b_j$, where δ_j is the leakage function of r_j. The term b_j represents the noise and is distributed according to $\mathcal{N}(0, \sigma^2)$ is usually assumed for simplicity [20] (yet, any noise distribution could theoretcially be analyzed in a maximum likelihood manner). The correct bit value is then recovered as the one maximizing the product of the probabilities of the register leakages:

$$\prod_j \Pr[l_j \mid (r_j|k_i, P)] = \prod_j \mathcal{N}(l_j \mid (r_j|k_i, P), \sigma^2).$$

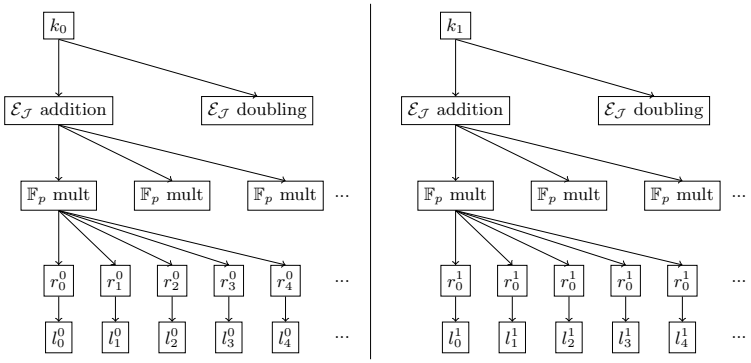

Fig. 1: Leveled view of a regular scalar multiplication. First level (top): scalar bit handling. Second: elliptic curve arithmetic. Third: Field arithmetic. Fourth: register operations. Fifth: leakages on register operations. (Taken from [22]).

3 Problem statement and challenges

One recurrent problem of side-channel security evaluations is the large amount of different state-of-the-art attacks [26] which makes it prohibitive to try all of them: ECC implementations are no exception to this issue. In this context, the HDPA described above is an interesting one to investigate, since it belongs to the most powerful type of attacks against ECC implementations. However, it comes at the cost of a complicated instantiation. First, it requires a precise knowledge of the implementation under attack. Second, it requires to profile every single operation of the ECSM. This step is computationally intensive and requires several manual optimizations in order to process the large ECSM traces in a reasonable time. In the following, we describe how this generic framework can be used for systematizing security evaluations and reducing their cost, by providing shortcut formulas to estimate the success rate of HDPA without performing it.

For this purpose, we draw our inspiration from the associated literature on symmetric cryptography. Indeed, shortcut formulas for success rate estimation

in the case of block ciphers are already a deeply investigated topic. In the simpler case of unprotected (more precisely, unmasked) implementations, efficient approximations of the success rate can typically rely on easy-to-compute metrics such as the Signal-to-Noise Ratio (SNR) [24, 11]. By contrast, for masked implementations, additional assumptions and/or metrics (e.g., the mutual information) are needed [19, 9, 10]. Since considering unprotected ECC implementations, we will next be in the former case and additionally exploit some of the ideas used in [13] for the analysis of multivariate/horizontal attacks. Precisely, we will adapt the SNR metric to the context of ECSM implementations and exploit it for the estimation of the success rate as a function of the number of targeted register leakages and the noise level.

3.1 SNR definitions

In general, the SNR of a device depends on the size of the bus (which defines the maximum signal), the adversary's guessing power (which defines the part of the bus that generates exploitable signal and the part that generates algorithmic noise) and the physical noise. It is defined as the variance of the (exploitable) signal divided by the noise variance. In this paper, we consider a 32-bit device and therefore assume that all the bits of the bus can be predicted (so no algorithmic noise). In the context of a standard DPA attack where the full bus is targeted [20], this would lead to an SNR32 defined for a register indexed j as:

$$\mathrm{SNR32}_j = \frac{\underset{r_j \in \mathbb{F}_{2^{32}}}{\mathsf{var}} \delta_j(r_j)}{\sigma^2}, \tag{1}$$

where δ_j is the deterministic (noise-free) part of the leakage function for a register r_j, as introduced in Section 2.3, and σ^2 is the noise variance. Further assuming an Hamming-weight leakage function for illustration, this leads to $\mathrm{SNR32} = \frac{8}{\sigma^2}$ (with 8=32/4 the variance of a random 32-bit Hamming weight).

When considering a 32-bit implementation of the Montgomery ladder ECSM, the situation slightly differs from this standard DPA context. Indeed, in this case the register content typically depends on a single key bit (rather than 32 ones in the standard DPA case). Therefore, each target register can only take two values instead of every value of $\mathbb{F}_{2^{32}}$. A vertical DPA against an ECSM therefore boils down to distinguishing the leakage of two 32-bit values, whereas a HDPA tries to exploit multiple registers. Concretely, this means that certain registers lead to easier-to-distinguish leakages. Yet, in order to improve the efficiency of the security evaluations, we will also use an average metric to estimate the success rate (and track the distance between this estimate and the success rate of concrete attacks). For this purpose, a first natural idea would be to consider a modified $\mathrm{SNR2}_j$ that captures the difference between the (noise-free) leakages of a register r_j for two scalar bit values:

$$\mathrm{SNR2}_j = \frac{\underset{P \in \mathcal{E}(\mathbb{F}_p)}{\mathbb{E}} \left(\delta_j(r_j | k_i = 0, P) - \delta_j(r_j | k_i = 1, P)\right)^2}{\sigma^2}. \tag{2}$$

3.2 Preliminary observations and caveats

HDPA aims at exploiting the leakages of a large number of leaking registers for a single key bit. In the symmetric case, this can be viewed as targeting several leakage samples for a single subkey (e.g., both the input and the output of a S-box). As a result, in this case we have that $\delta_i = \delta_j$ trivially implies $\text{SNR32}_i = \text{SNR32}_j$. This basically means that under the assumption $\delta_i = \delta_j$, the estimation of the SNR32 is only required for a single register. Grosso and Standaert use this same assumption (of similar leakage functions for all their target intermediate computations) to speed up the computation of a multivariate mutual information to an univariate one [13].

Following this approach, a tempting strategy for ECSM evaluation would thus be to also assume that the leakage functions are similar for all the registers, leading to a constant SNR2. To evaluate the soundness of this approach, Figure 2 illustrates the SNR2_j for each register r_j corresponding to the high 32-bit words of multiplication results. Our reference implementation introduced in Section 2 requires $N = 1600$ register multiplications to process one key bit. The SNR2 is evaluated for a Hamming weight leakage model, so exactly fulfilling the assumption of identical leakage functions, and averaged over 10,000 randomly sampled elliptic curve points. We observe that SNR2_j is not constant even when the leakage function is the same for all registers. These differences can be explained by the algebraic relations between the values computed when the scalar bit equals 0 and the values when this bit equals 1. For example, the regions of high SNR2 on Figure 2 observed in the register index intervals [512,576] and [704,768] respectively correspond to the 9th and 12th field multiplication in the point addition algorithm described in appendix A. For a bit value, it performs the operations E^2 and H^2 or the operations $(-E)^2$ and $(-H)^2$ when the bit is flipped. This leads to bigger differences in the side-channel leakage, as the bits of the opposite of a field element modulus the NIST P-256 prime [23] are almost all flipped. The peaks of zero SNR2 in register index interval [896,960] correspond to the 15th field multiplication in the point addition algorithm. It performs the operation Z_1Z_2 or Z_2Z_1 when the bit is flipped. Since the same elements are multiplied in both cases, 8 equal cross products appear during the computation of the Long Integer Multiplication, thus leading to no information ($\text{SNR2} = 0$).

These observations imply that, as opposed to the symmetric case, a single register cannot be used to evaluate the security of ECSM implementations with respect to vertical DPA, even if the leakage function is the same for all the registers. This variation of the SNR2 highlights the fact that when performing these attacks, some leakage points are more interesting than others. So strictly speaking, such a simple evaluation is not possible for HDPA either.

3.3 The single trace attack scenario

Besides the previous caveat, another difficulty arises from the contradiction between HDPA that are essentially designed to succeed in a single-trace attack context (e.g., against a randomized key or an unknown scalar nonce) and the

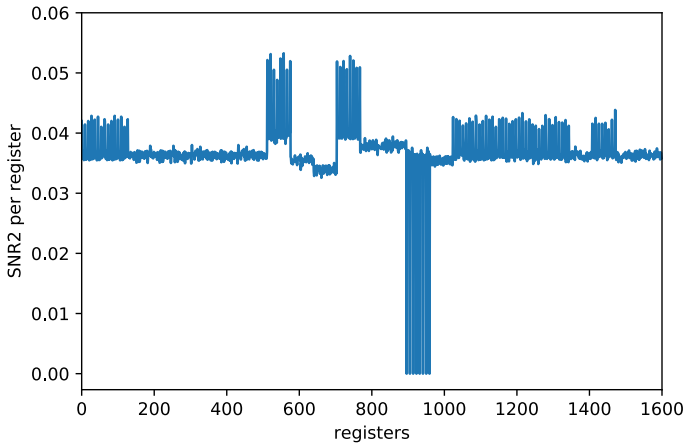

Fig. 2: Average SNR2 for the 1600 targeted registers of field multiplications in the Montgomery ladder.

SNR2 metric which corresponds to an average amount of information collected over multiple points (and is therefore more in line with a vertical DPA).

In order to deal with this issue, we therefore start by defining the register-specific amount of information that corresponds to the distance between a register value when the scalar bit equals 0 and its value when the scalar bit equals 1, for a fixed EC point. For register r_j and a point P, we denote this distance by $d_j(P)$, whose definition is given by Equation 3:

$$d_j(P) = (\delta_j(r_j|k_i = 0, P) - \delta_j(r_j|k_i = 1, P))^2. \quad (3)$$

Perfectly characterizing the security of an ECSM against HDPA naturally requires characterizing all the distances $d_j(P)$. Yet, and interestingly, we will next show that the two challenges described in this section (i.e., the fact that the SNR2 metric is register-dependent and that HDPA is primarily designed for single-trace attacks) can be mitigated concurrently. For this purpose, the main observation is that in view of the number of registers in ECSM implementations, it seems reasonable that the success of an attack targeting all the registers at once actually gets close to the average one. We next formalize this idea and describe the additional assumptions it requires.

4 Efficient success rate approximation

In this section, we show how to derive an approximation of the success rate with respect to the HDPA framework for one scalar bit recovery. Since we consider two equally likely Gaussian hypotheses, the SR when targeting a single register r_j for a given point P is computed as in [6] and given by Equation 4:

$$\mathrm{SR} = \Phi\left(\frac{\sqrt{(\delta_j(r_j|k_i=0,P) - \delta_j(r_j|k_i=1,P))^2}}{2\sigma}\right). \tag{4}$$

We recall that the HDPA described in the previous section assumes IOL for computational efficiency. It was also noted by Poussier et al. [22] that fully characterizing the traces' covariance does not improve the attack results in case of profiling with bounded number of measurements. So we next leverage this assumption and recall that it is a conservative one (deviations can only reduce the attack effectiveness). It allows us to easily extend the previous formula to the case where an attacker would exploit N registers at once. Interestingly, it also re-enforces the analogy between vertical and horizontal DPA. Indeed, the IOL assumption divides the side-channel trace into N univariate samples, which roughly corresponds to a vertical DPA using N traces. As a result, similarly to the vertical DPA case [9], the SR of the horizontal DPA exploiting N samples, denoted by SR^N, is given by Equation 6.

$$\mathrm{SR}^N = \Phi\left(\frac{\sqrt{N \cdot \mathop{\mathbb{E}}_j \ (\delta_j(r_j|k_i=0,P) - \delta_j(r_j|k_i=1,P))^2}}{2\sigma}\right), \tag{5}$$

$$= \Phi\left(\frac{\sqrt{N \cdot \mathop{\mathbb{E}}_j \ d_j(P)}}{2\sigma}\right). \tag{6}$$

As shown by Figure 2, the vertical signal SNR2 is not constant across all registers as opposed to the symmetric case. This is also true for the horizontal signal $d_j(P)$, which will inevitably vary depending on the point and the targeted register. As a result, a strict approximation of the SR^N using Equation 6 would require to compute $d_j(P)$ for every single register. This requires a first step of leakage characterization [6, 25]. This step is quite intensive and the most time and data consuming in HDPA. Using Equation 6, the SR approximation is just as complex and tedious as performing HDPA. This observation shows the need of additional assumptions in order to simplify the security evaluation.

4.1 Additional assumptions

Identical Leakage Functions assumption (ILF): We first assume that the leakage function is identical across all registers: $\delta_i = \delta_j = \delta$, for $i, j \in [0, N-1]$. Note that this is a common assumption that is also used in the security evaluation of masked implementations of block ciphers [13].

Asymptotic Uniformity assumption (AU): We define the notion of *ideal distance* d_{id} as the square difference between the noise-free leakages of two uniformly distributed values $V_1, V_2 \sim \mathcal{U}(\mathbb{F}_{2^{|r|}})$. More formally, given a leakage function δ, the ideal distance is given by Equation 7:

$$d_{\mathrm{id}} = \mathop{\mathbb{E}}_{v_1,v_2} \ (\delta(v_1) - \delta(v_2))^2. \tag{7}$$

Naturally, d_{id} can be seen as the *vertical* information provided by a register whose values are uniformly distributed when the input point varies. Our main assumption, the AU, states that the mean of the distances $d_j(P)$ over a large number of registers tends towards d_{id}, as given by Equation 8. Informally, it means that the average horizontal information $d_j(P)$ for a fixed point P of a big enough number of registers is equal to the vertical information of a single uniformly distributed register:

$$\mathop{\mathbb{E}}_{j=0}^{N-1} d_j(P) \xrightarrow[N\to+\infty]{} d_{\text{id}}. \tag{8}$$

4.2 Efficient success rate approximation

Using the two assumptions introduced in the previous subsection, we can efficiently estimate the SR of HDPA against an ECSM implementation. For the AU assumption, we further assume that the number N of exploited registers is big enough so that $\mathop{\mathbb{E}}_{j} d_j(P)$ is close to d_{id}. As a result, the SR^N approximation is boiled down to the computation of d_{id} and the estimation of the noise level σ. The success rate formula of Equation 6 is then trivially adapted to d_{id} as:

$$\text{SR}^N = \Phi\left(\frac{\sqrt{N \cdot d_{\text{id}}}}{2\sigma}\right). \tag{9}$$

Hamming weight leakage example: We illustrate this formula with an example based on a Hamming weight leakage function HW. The HW of a uniform random variable on $\mathbb{F}_{2^{|r|}}$ is approximately distributed as $\mathcal{N}(\frac{|r|}{2}, \frac{|r|}{4})$. For $U_1, U_2 \sim \mathcal{U}(\mathbb{F}_{2^{|r|}})$, we have $\mathsf{HW}(U_1) - \mathsf{HW}(U_2) \sim \mathcal{N}(0, \frac{|r|}{2})$ and $(\mathsf{HW}(U_1) - \mathsf{HW}(U_2))^2 \sim \Gamma(\frac{1}{2}, |r|)$, where Γ denotes the Gamma distribution, here with shape parameter $\frac{1}{2}$ and scale parameter $|r|$. If the AU assumption holds, then the ideal distance d_{id} is equal to $\frac{|r|}{2}$. The corresponding success rate is given by Equation 10:

$$\text{SR}^N = \Phi\left(\frac{\sqrt{N \cdot |r|}}{2\sqrt{2}\sigma}\right). \tag{10}$$

4.3 Potential invalidation of the assumptions

The previous equations express the SR of a HDPA as a function of its main parameters, which allows gaining intuition about how the complexity of such attacks scales. Yet, the concrete correctness of this proposal depends on the ILF and AU assumptions. In this subsection, we discuss how realistic these assumptions are, and identify issues that may contradict them in practice.

Algorithmic issue. Even if the leakage model is the same for all targeted registers, the SNR2 is not identical for all registers, as seen on Figure 2. The SNR2

and the distances $d_j(P)$ depend on the distribution of the intermediate values, the ECSM algorithm, the curve representation and the finite field arithmetic.

Physical issue. While commonly used in SCAs, the ILF assumption is never fully verified in practice [12]. We might observe $\delta_i \neq \delta_j$ when $i \neq j$. This can introduce additional discrepancies among the registers' distances. It can impact the convergence of the mean distance to the ideal distance and thus the accuracy of the SR approximation using the AU assumption.

In the next sections, we show the validity of our approximations with respect to both issues. First, in section 5, simulations are used to show that the AU assumption provides a valid approximation of the behavior of ECSM intermediate values. Next, in section 6, we use real measurements to show that the physical errors are not too problematic for the accuracy of the SR approximation.

5 Simulated experiment: the algorithmic issue

In this section, we tackle the algorithmic issue due to the distribution of the intermediate values of the ECSM and to what extent it deviates from a uniform distribution. For that purpose we use a perfect setting with a HW leakage function using simulated traces, to fulfill the ILF assumption so that conclusions are not affected by any physical aspect of a real device's leakages. We consider our reference implementation of the Montgomery ladder described in Section 2. We consider an attacker targeting all the $N = 1600$ multiplication results. For each execution with a random point P and a random scalar bit k_i, we are provided with the register values $(r_j^i)_{0 \leq j < 1600}$ along with their simulated leakages $l_j^i = \mathsf{HW}(r_j^i) + b_j$ where $b_j \sim \mathcal{N}(0, \sigma^2)$. The noise level is chosen to replicate the target device in Section 6: $\sigma^2 = 440$.

Convergence towards the ideal distance: On Figure 3 we plot the evolution of the average distances of 1000 elliptic curve points. The y axis corresponds to the mean distance computed over the registers indexed by the x axis. Each colored curve corresponds to one randomly chosen elliptic curve point. The horizontal black line represents the ideal distance $d_{\mathrm{id}} = 16$. We can observe that even though we only consider 1600 registers over the large number of leaking registers of an ECSM execution, for different points P of the elliptic curve $\mathcal{E}(\mathbb{F}_p)$, the average distances $\frac{1}{N}\mathop{\mathbb{E}}_{j=1}^{N-1} d_j(P) = \frac{1}{N}\mathop{\mathbb{E}}_{j=1}^{N-1}(\delta(r_j|k_i = 0, P) - \delta(r_j|k_i = 1, P))^2$ tends towards the ideal distance d_{id} when N gets larger[2]. This result shows that the AU assumption can roughly describe the behavior of the ECSM intermediate values' leakages, and the approximation is more and more accurate as the number of registers required for the success of the attack increases.

Success rate approximation: The perfect simulated setting allows us to investigate the impact of the AU assumption on the SR approximation. Namely

[2] Note that attacking several scalar bits at the same time would also result in increasing the number of registers, thus positively impacting the convergence.

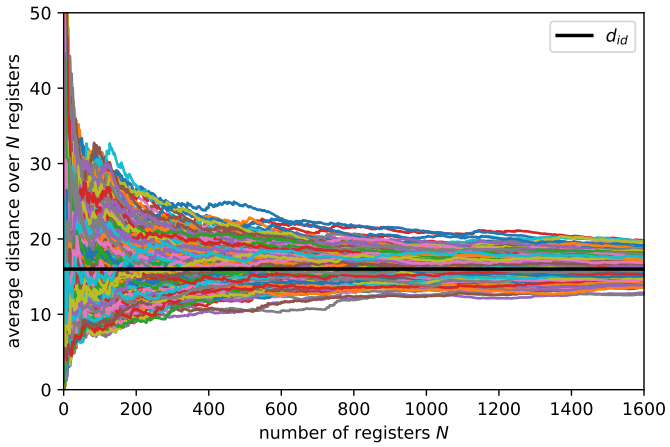

Fig. 3: Convergence of the average HW distances towards the ideal distance.

the real success rate is only biased by the distances of register values. The results of the simulated experiments are illustrated in Figure 4. Each of the 20 colored curves corresponds to the SR of HDPA evaluated for one elliptic curve point (repeated 100 times), and the orange curve is the SR approximation using d_{id}. Knowing the noise variance σ^2, the later is computed using Equation 10. The figure suggests that the approximation predicts well enough the real SR, showing that the algorithmic issue is not problematic for this particular implementation.

Impact of the noise: We performed the attack multiple times for different SNR32 values. First, the SNR32 of the target device in Section 6 (SNR32 = 0.0182), and additionally for SNR32 $\in \{0.1, 0.5, 1\}$. The results are depicted in Figure 5. The solid curves represent the SR of HDPA as function of the number of registers, and the dashed curves the corresponding approximations using the AU assumption. We draw attention to the gap between the real SR (solid line) and the SR approximation (dashed line) computed using Equation 10, which gets tighter as the SNR32 decreases. The bias introduced by the intermediate values of the ECSM makes the average distance over the small number of registers required for the success of HDPA for high SNR32 slightly deviate from the ideal distance. As the SNR32 decreases, this bias becomes smaller compared to the variance of the noise. Moreover, HDPA requires more registers to succeed, and thus requires to sum the distances over multiple registers which would tend towards the ideal distance as shown by Figure 3. This is an interesting result as we are mainly interested in the low SNR32 case, as it corresponds to high security devices that require worst-case analysis.

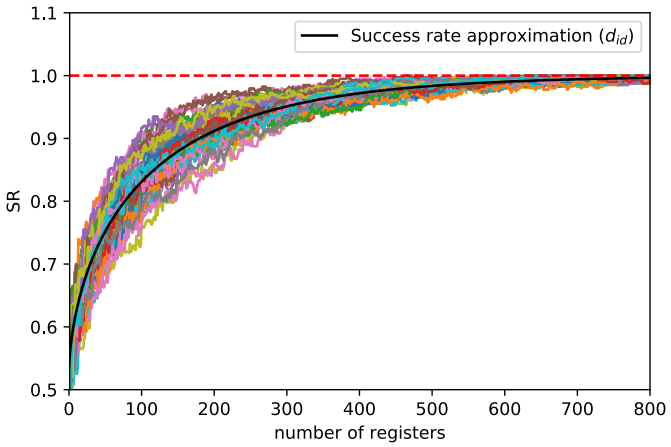

Fig. 4: Comparison of the SR and its approximation on simulated HW leakages.

6 Real experiment: the physical issue

In this section, we investigate the impact of the ILF assumption on the accuracy of the SR approximation using the AU assumption. We use real measurements, where a different leakage function δ_j is expected for each leaking register r_j. Our experiments target our reference implementation similarly to the simulated case. The target device is a 32-bit ARM Cortex-M4 micro-controller from the Atmel SAM4C-EK evaluation kit [2, 1] running at 100 MHz. We monitored the voltage variation using a 4.7 Ω resistor inserted in the power supply circuit of the chip. We performed the trace acquisition using a Lecroy WaveRunner HRO 66 ZI oscilloscope running at 200 megasamples per second. We recorded the execution of 10,000 scalar multiplications. For each of them, we triggered the measurement at the beginning of the execution and recorded the processing of one scalar bit. We performed HDPA such as described in Section 2 but assuming two different leakage models. First, a linear regression taking as a basis the Hamming Weight of the leaking registers, similarly to the simulated experiment in the previous section. This yields for every register r_j, a leakage function of the form $\delta_j(r_j) = a_j + b_j \cdot \mathsf{HW}(r_j)$. Additionally, we performed HDPA for a linear regression based leakage model using a 32-bit basis, such as described by the original attack by Poussier et al [22].

Hamming weight linear regression: We study the influence of the physical issue on the convergence of the average distance across multiple registers towards the ideal distance. We start by evaluating the distance $d_j(P)$ for each register: $d_j(P) = b_j^2(\mathsf{HW}(r_j|k_i = 0, P) - \mathsf{HW}(r_j|k_i = 1, P))^2$. We aim to compare the ideal distance to the mean distance for multiple different elliptic curve points.

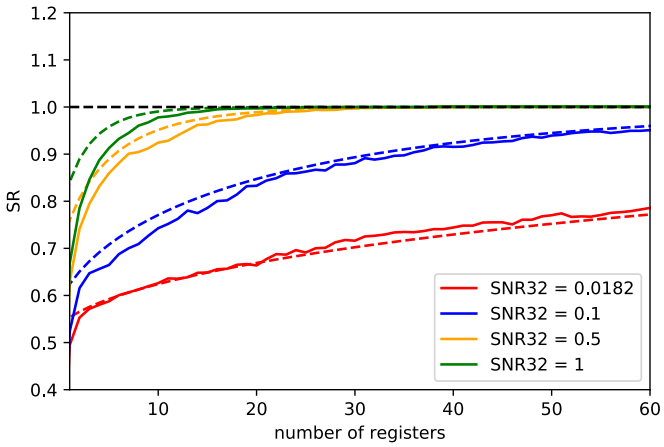

Fig. 5: Impact of the noise on the SR approximation.

We evaluated the leakage model for the ideal distance using 20 random register leakages. Figure 6 depicts the convergence of the average distance over the leaking registers for 1000 random elliptic curve points towards the ideal distance d_{id}. We observe that the distances indeed converge towards the ideal distance similarly to the simulated case in Figure 3 despite different leakage models for each individual register. Additionally, Figure 7 shows the comparison between the real SR in blue of HDPA on real traces acquired from the target device previously described and its approximation in orange given by Equation 9. We averaged the SR over multiple points, so that conclusions are not affected by the algorithmic issue. We note that the approximation is still satisfactory but less accurate than in the simulated case. This is expected as the attack is performed on real side-channel measurements and the HW is not the most accurate leakage modeling strategy for this device, while the SR approximation assumes that the leakage model has been perfectly characterized.

32-bit linear regession: We plot the convergence towards the ideal distance on Figure 8 for 1000 random elliptic curve points. We evaluated again the leakage model for the ideal distance using 20 random register leakages. We notice that despite having different leakage coefficients for each individual bit of the 1600 registers, the average distances still tend towards the ideal distance. This additional result further highlights the soundness of the AU assumption. Figure 9 shows the comparison between the SR of the HDPA in blue and its approximation in orange evaluated using Equation 9. First, we notice that the SR approximation for a full basis linear regression attack is more accurate compared to the HW model case. This is due to the fact that the leakage of the considered device is best estimated by the second model.

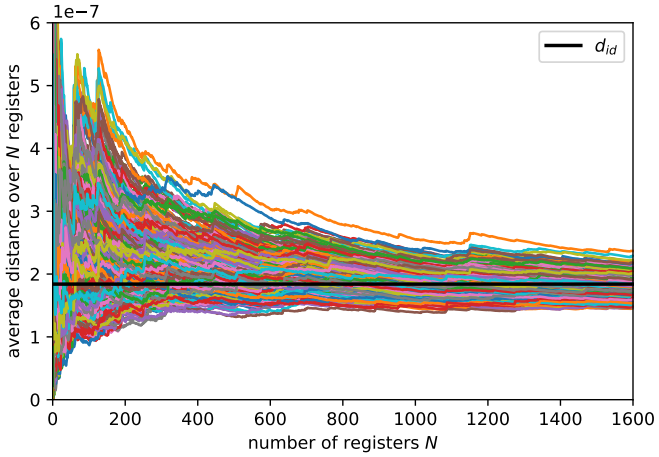

Fig. 6: Convergence towards the ideal distance of the average distances with different HW based real leakage models for each register.

7 Conclusion

Assessing the SCA security of an implementation is a tedious task. This is particularly true for complex cryptosystems for which numerous attack paths are possible. In this paper, we described a first methodology for analyzing the security of ECSM implementations against (close to) worst-case HDPA It allows us to express the success rate of such attacks based on an easy-to-estimate (ideal distance) metric, in function the number of leakage samples exploited by the adversary (which depends on the register size, the field size and the number of field operations) and the noise level. This shortcut formula trades a bit of accuracy in the success rate estimation for considerable efficiency gains. It could be easily extended to windowed algorithms and to the SR approximation of a full scalar recovery. Future works might investigate the application of this methodology to other implementations of public-key cryptosystems.

Acknowledgement. François-Xavier Standaert is a senior research associate of the Belgian Fund for Scientific Research. This work has been funded in part by the European Commission through the H2020 project 731591 (acronym REASSURE) and by the ERC Consolidator Grant 724725 (acronym SWORD). The authors acknowledge the support from the 'National Integrated Centre of Evaluation' (NICE), a facility of Cyber Security Agency, Singapore (CSA). The authors would like to thank Vincent Verneuil for the valuable comments and the fruitful discussions.

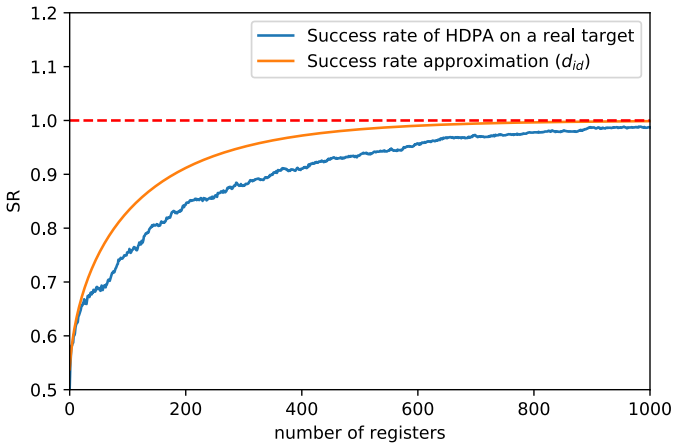

Fig. 7: Comparison of the SR of HDPA and its approximation assuming a HW based linear leakage model.

References

1. Atsam4c-ek user guide : `http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel_11251_SmartEnergy_ATSAM4C-EK-User_Guide_SAM4C8-SAM4C16_User-Guide.pdf`.
2. Cortex-m4 technical reference manual : `http://infocenter.arm.com/help/topic/com.arm.doc.ddi0439b/DDI0439B_cortex_m4_r0p0_trm.pdf`.
3. Lejla Batina and Matthew Robshaw, editors. *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*. Springer, 2014.
4. Aurélie Bauer, Éliane Jaulmes, Emmanuel Prouff, and Justine Wild. Horizontal and vertical side-channel attacks against secure RSA implementations. In *Topics in Cryptology - CT-RSA 2013 - The Cryptographers' Track at the RSA Conference 2013, San Francisco,CA, USA, February 25-March 1, 2013. Proceedings*, pages 1–17, 2013.
5. Aurélie Bauer, Eliane Jaulmes, Emmanuel Prouff, and Justine Wild. Horizontal collision correlation attack on elliptic curves. In Tanja Lange, Kristin Lauter, and Petr Lisoněk, editors, *Selected Areas in Cryptography – SAC 2013*, pages 553–570, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
6. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, pages 13–28, 2002.
7. Christophe Clavier, Benoit Feix, Georges Gagnerot, Mylène Roussellet, and Vincent Verneuil. Horizontal correlation analysis on exponentiation. In *Informa-*

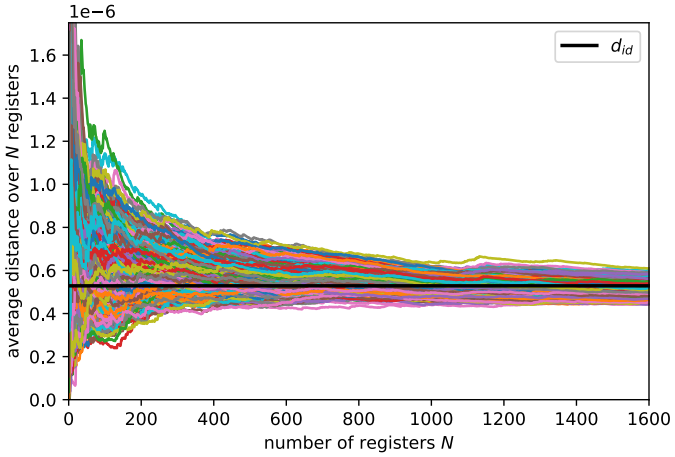

Fig. 8: Convergence towards the ideal distance of the average distances evaluated with a different linear regression function for each register.

tion and Communications Security - 12th International Conference, ICICS 2010, Barcelona, Spain, December 15-17, 2010. Proceedings, pages 46–61, 2010.

8. Jean-Sébastien Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In *Proceedings of the First International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '99, pages 292–302, London, UK, UK, 1999. Springer-Verlag.
9. A. Adam Ding, Liwei Zhang, Yunsi Fei, and Pei Luo. A statistical model for higher order DPA on masked devices. In Batina and Robshaw [3], pages 147–169.
10. Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete - or how to evaluate the security of any leaking device. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 401–429, 2015.
11. Yunsi Fei, Qiasi Luo, and A. Adam Ding. A statistical model for dpa with novel algorithmic confusion analysis. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems – CHES 2012*, pages 233–250, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
12. Benoît Gérard and François-Xavier Standaert. Unified and optimized linear collision attacks and their application in a non-profiled setting: extended version. *J. Cryptographic Engineering*, 3(1):45–58, 2013.
13. Vincent Grosso and François-Xavier Standaert. Masking proofs are tight and how to exploit it in security evaluations. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, pages 385–412, 2018.
14. Marc Joye and Christophe Tymen. Protections against differential analysis for elliptic curve cryptography — an algebraic approach —. In Çetin K. Koç, David

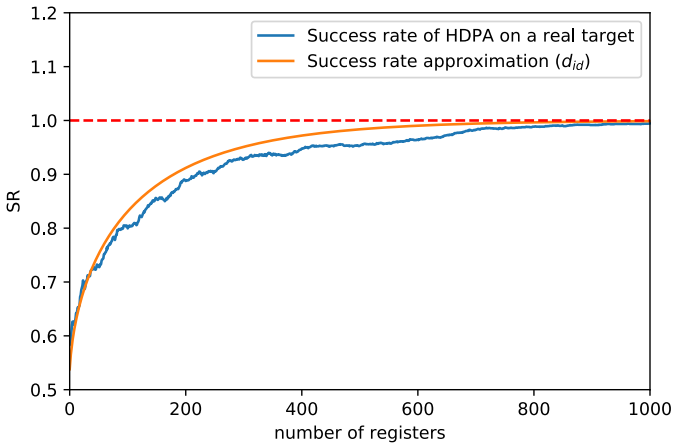

Fig. 9: Comparison of the SR of HDPA and its approximation assuming a linear regression leakage model.

Naccache, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2001*, pages 377–390, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

15. Marc Joye and Sung-Ming Yen. The montgomery powering ladder. In Burton S. Kaliski, çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, pages 291–302, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
16. Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO' 99*, pages 388–397, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.
17. Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '96, pages 104–113, London, UK, UK, 1996. Springer-Verlag.
18. Duc-Phong Le, Chik How Tan, and Michael Tunstall. Randomizing the montgomery powering ladder. In Raja Naeem Akram and Sushil Jajodia, editors, *Information Security Theory and Practice*, pages 169–184, Cham, 2015. Springer International Publishing.
19. Victor Lomné, Emmanuel Prouff, Matthieu Rivain, Thomas Roche, and Adrian Thillard. How to estimate the success rate of higher-order side-channel attacks. In Batina and Robshaw [3], pages 35–54.
20. 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.
21. Marcel Medwed and Elisabeth Oswald. Template attacks on ecdsa. In Kyo-Il Chung, Kiwook Sohn, and Moti Yung, editors, *Information Security Applications*, pages 14–27, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

22. 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 *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 534–554, 2017.
23. NIST FIPS PUB. 186-2: Digital signature standard (dss). *National Institute for Standards and Technology*, 2000.
24. Matthieu Rivain. On the exact success rate of side channel analysis in the gaussian model. In *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, pages 165–183, 2008.
25. Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In *CHES 2005*, pages 30–46, 2005.
26. Mathias Wagner. 700+ attacks published on smart cards: The need for a systematic counter strategy. In Werner Schindler and Sorin A. Huss, editors, *Constructive Side-Channel Analysis and Secure Design*, pages 33–38, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

A Addition and doubling formulas

Algorithm 2 Point addition using Jacobian coordinates

Input $P = (X_1, Y_1, Z_1)$, $Q = (X_2, Y_2, Z_2)$
Output $P + Q = (X_3, Y_3, Z_3)$
$A \leftarrow Z_1^2$, $B \leftarrow Z_2^2$, $C \leftarrow X_1 B$, $D \leftarrow X_2 A$, $E \leftarrow C - D$, $F \leftarrow Y_1 B Z_2$, $G \leftarrow Y_2 A Z_1$, $H \leftarrow F - G$, $I \leftarrow E^2$, $J \leftarrow IE$, $K \leftarrow CI$
$X_3 \leftarrow H^2 + J - 2K$
$Y_3 = H(K - X_3) - FJ$
$Z_3 = Z_1 Z_2 E$
return (X_3, Y_3, Z_3)

Algorithm 3 Point doubling using Jacobian coordinates

Input $P = (X_1, Y_1, Z_1)$
Output $P + P = (X_2, Y_2, Z_2)$
$A \leftarrow X_1^2$, $B \leftarrow Y_1^2$, $C \leftarrow Z_1^2$, $D \leftarrow 3A + aC^2$, $E \leftarrow B^2$, $F \leftarrow 4X_1 B$
$X_2 \leftarrow D^2 - 2F$
$Y_2 \leftarrow D(F - X_2) - 8E$
$Z_2 = 2Y_1 Z_1$
return (X_2, Y_2, Z_2)

In the point doubling algorithm described above, the multiplication by $a = -3$ is done using field subtraction, leading to one less field multiplication.