Noname manuscript No.
(will be inserted by the editor)

Optimal Side-Channel Attacks for Multivariate Leakages and Multiple Models

Nicolas Bruneau[1,2] · **Sylvain Guilley**[3,1] · **Annelie Heuser**[1] · **Damien Marion**[3,1] · **Olivier Rioul**[1,4]

Received: date / Accepted: date

Abstract Side-channel attacks allow to extract secret keys from embedded systems like smartcards or smartphones. In practice, the side-channel signal is measured as a trace consisting of several samples. Also, several sensitive bits are manipulated in parallel, each leaking differently. Therefore, the informed attacker needs to devise side-channel distinguishers that can handle both *multivariate* leakages and *multiple* models. In the state-of-the-art, these two issues have two independent solutions: on the one hand, dimensionality reduction can cope with multivariate leakage; on the other hand, online stochastic approach can cope with multiple models.

In this paper, we combine both solutions to derive closed-form expressions of the resulting *optimal* distinguisher in terms of matrix operations, in all situations where the model can be either profiled offline or regressed online. Optimality here means that the success rate is maximized for a given number of traces. We recover known results for uni- and bi-variate models (including correlation power analysis), and investigate novel distinguishers for multiple models with more than two parameters. In addition, following ideas from the AsiaCrypt'2013 paper "*Behind the Scene of Side-Channel Attacks*", we provide fast computation algorithms in which the traces are accumulated prior to computing the distinguisher values.

Keywords Side-channel analysis · Optimal distinguishers · Multivariate leakage · Stochastic attacks

[1] LTCI, CNRS, Télécom ParisTech, Université Paris-Saclay, 75 013 Paris, France
[2] STMicroelectronics, AST Division, 13 790 Rousset, France
[3] Secure-IC S.A.S., Threat Analysis Business Line, 35 510 Cesson-Sévigné, France
[4] CMAP, École polytechnique, Université Paris-Saclay, 91 128 Palaiseau, France
E-mail: firstname.lastname@telecom-paristech.fr

1 Introduction

Side-channel attacks allow to extract secret keys from cryptographic devices. Template attacks [**?**] have been introduced as the strongest analysis method. They consist in two phases: (*i*) a profiling offline phase were the leakage model of the device under attack is characterized; (*ii*) an attack online phase in which the secret key is extracted using fresh measurements along with the pre-characterized model. Such attacks are known to use a maximum likelihood principle to ensure the highest possible success probability (see, eg., [**?**]).

In this paper we study *optimal* attacks with the best possible success probability when extracting the secret key[1]. We leverage on such optimal distinguishers to answer the following question: how to attack with the best probability of success when the *leakage* is *multivariate* and the *model* are *multiple*? An initial empirical[2] work has already been carried out in [**?**] which confirmed that this type of approach can be very fruitful[3].

[1] The success probability in key recovery is chosen as a figure of merit for optimization. Such an objective is typical of "pure" side-channel attacks. Other approaches [**?**,**?**,**?**] relax the condition that the key found by the side-channel analysis be ranked first and complements it with a key enumeration stage. This yields a data vs. complexity tradeoff that is not explored in this paper.

[2] The work in [**?**] does not detail the *modus operandi* result for the regression neither plugs it into the distinguisher, which is incidentally not chosen to be the optimal one.

[3] *Multi-target attacks* [**?**, **?**] have a somewhat different goal, namely the best aggregation of information about several subparts of a key, possibly leaking at different times with different models, in order to recover the full key efficiently. Here we consider only *one* multivariate leakage model and focus on recovering *one* subpart of the key. However, our derivation is capable of handling multivariate leakages and models and may still be combined with the multi-target approaches.

Contributions. We derive closed-form expressions for the optimal distinguishers in all situations where the model is known (e.g., using profiling) or regressed online. In the case of a known univariate model, we recover the results in [**?**], However, our "fully matrix" formalism makes equations simpler and proofs shorter. Moreover, compared to [**?**] we extend the leakage model to the case where the traces are not necessarily centered, thereby allowing a more natural application on real traces. In the realistic "(on-line) stochastic attack" situation where the model is *parametric*, i.e. where the coefficients of the model are unknown, we express the optimal distinguisher by maximizing success over the whole set of possible coefficients. Finally, we provide fast computation algorithms for our novel distinguishers, which happen to be remarkably simple and efficient.

Outline. The remainder of this paper is organized as follows. Sec. 2 provides a modelization of a side-channel attack that is generic[4] enough to capture many different multivariate scenarios The main results of this paper are outlined in Sec. 3. Sec. 4 presents experimental results on simulated traces and real-world acquisition campaigns. Conclusions and perspectives are in Sec. 5.

2 Notations and Leakage Model

2.1 Notations

We let X denote the leakage measurements, Y the model, N the measurement noise, and α the link between the model and the measurements[5]. The model Y depends on a key guess k, an n-bit vector, and on some known text T (usually also an n-bit vector) e.g., through a function ϕ such that $Y = \phi(T, k)$. A well-known example is $Y = w_H(T \oplus k)$, where w_H is the Hamming weight function. However, in general, some parameters of the model are unknown. To remain generic, we do not detail further the link between Y and the pair (T, k). As it is customary in side-channel analysis, the correct key is denoted by $k^\star$. The corresponding model using the correct key $Y(k^\star)$ is denoted by $Y^\star$.

Let Q be the number of queries (number of measurements), D be the data dimensionality (number of time samples per measurement trace) and S be the model dimensionality ($\phi : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{R}^S$ is a vectorial function, with S components). Roman letters in **bold** indicate vectors or matrices that have a dimension in Q, i.e., which are different for each trace $q = 1, 2, \ldots, Q$. More precisely, $\mathbf{X}$ represents the full attack campaign, a matrix of $D \times Q$ measurement samples. The q-th trace is denoted X_q which is a $D \times 1$ column vector. Similarly, for the q-th trace, the $S \times 1$ column vector Y_q represents the deterministic part of the model while the $D \times 1$ column vector N_q is the corresponding measurement noise with $D \times D$ correlation matrix Σ.

We denote by $\mathrm{tr}\,(\cdot)$ the trace of a square matrix, that is the sum of its diagonal terms. Note that $\mathrm{tr}\,(AB) = \mathrm{tr}\,(BA)$ for compatible matrix dimensions. Let $\|\cdot\|_2$ denote the Euclidean norm of a $1 \times Q$ row vector. Thus $\|\mathbf{X}\|_2^2 = \mathbf{X}\mathbf{X}^\mathsf{T} = \mathrm{tr}\left(\mathbf{X}^\mathsf{T}\mathbf{X}\right)$, where $(\cdot)^\mathsf{T}$ is the transposition operator. Finally let $\|\cdot\|_F$ denote the Frobenius norm of a matrix (square root of the sum of its squared elements), such that $\|M\|_F^2 = \mathrm{tr}\left(MM^\mathsf{T}\right)$.

2.2 General Model

We make the following simplifying assumptions. First, the (environmental) noise is steady, e.g., chip temperature and supply voltage do not vary during the side-channel measurements. Thus $N_1, N_2, \ldots, N_Q$ are independent and identically distributed (i.i.d.) (denoted by N with index q dropped). Second, the attacker does not inject partial information gathered from the leakage analysis into a possible choice of plaintexts/ciphertexts (nonadaptive attack)[6]. Thus $Y_1, Y_2, \ldots, Y_Q$ are assumed i.i.d. (denoted by Y). Under the adopted leakage model it follows that the leakage measurements $X_1, X_2, \ldots, X_Q$ are also i.i.d. (denoted by X).

A distinguisher $\mathcal{D}$ maps a collection of leakages $\mathbf{x}$ and texts $\mathbf{t}$ to an estimation of the secret key $k^\star$. Let us recall that $\mathbf{x}$ and $\mathbf{t}$ are realizations of random variables $\mathbf{X}$ and $\mathbf{T}$: $\mathbf{x}$ is a $D \times Q$ matrix of real numbers (the acquisition campaign) and $\mathbf{t}$ is a $1 \times Q$ vector of n-bit words (bytes when $n = 8$) which are the publicly known plaintext or ciphertext bytes. An *optimal* distinguisher maximizes the probability of success $\mathcal{D}(\mathbf{x}, \mathbf{t}) = k^\star$.

The simplest situation occurs when X consists in a modulation of Y plus noise, in which case we let α be the signal envelope. In real traces, however, we face the more general situation where the model can be offset by some quantity the general case being an S-dimensional *parametric* model with $S \geq 2$ components. For this reason, we consider α as a $D \times S$ matrix and we set in

[4] By *generic*, we qualify a leakage model more complex than the classical Hamming weight or distance, where each bit of the sensitive variable has different strengths of leakage (situation we will show to happen in practice).

[5] Notations X, Y are consistent with the usual convention in machine learning, where X is for the collected data and Y for the classification labels.

[6] In fact, our results tolerate chosen texts, but consider them as observed inputs to the attack. We do not optimize the attack according to chosen inputs.

matrix notation

$$\mathbf{X} = \alpha \mathbf{Y}^\star + \mathbf{N} \tag{1}$$

where $\mathbf{X}$ is $D \times Q$, α is $D \times S$, $\mathbf{Y}^\star$ is $S \times Q$, and $\mathbf{N}$ is $D \times Q$. Notice that our convention to consider traces as lines and dimensions as rows allows us to write the deterministic part of the leakage as $\alpha\mathbf{Y}^\star$ which writes more naturally than the opposite order where traces would be viewed as a vertical time series[7].

We notice that in the state-of-the-art, monovariate leakage models are assumed vectorial in the context where they are considered unknown pseudo-Boolean functions of the pair (T, k). In this paper, we highlight that these coefficients extend to waveforms in the context of multivariate leakage, and thus take on a physical interpretation of leakage models as a sum of waveforms resulting from the leakage of individual resources. This means that seeing α as D lines, each representing the series of S weights, is awkward, since not related to the way the multivariate leakage is built from the processed data. Instead, it is natural to see α as S columns, each representing the waveform which is generated by one coordinate of model Y.

For each trace $q = 1, 2, \ldots, Q$, we assume that the vector $N = N_q$ follows that same multivariate normal distribution $\mathcal{N}(0, \Sigma)$, where the $D \times D$ correlation matrix $\Sigma = \mathbb{E}(NN^\mathsf{T})$ is assumed known to the attacker[8]. Since Σ is assumed symmetric positive definite, there exists a matrix $\Sigma^{1/2}$, which is such that $\Sigma^{1/2}\Sigma^{1/2} = \Sigma$. We refer to $\Sigma^{1/2}$ as the *standard deviation noise* matrix.

The model (1) used throughout the paper is quite generic and has multiple facets depending on the choice of S and the respective values given to α and Y. This is discussed next.

2.3 Examples with $S = 2$ and $S = 9$

For $S = 1$, the traces consist only in a *modulation* of the model plus noise as in [**?**, **?**]. When considering traces that are not only modulated but also have an *offset* term we have $S = 2$. We then write the 2-dimensional model as $\binom{\mathbf{Y}}{\mathbf{1}}$, where $\mathbf{Y}$ and $\mathbf{1}$ are $1 \times Q$ matrices $(Y_1, Y_2, \ldots, Y_Q)$ and $(1, 1, \ldots, 1)$. The $D \times 2$ matrix α in (1) actually takes the special form $(\alpha\ \beta)$ where β is the offset value.

An illustration is provided in Fig. 1 where the parameter $\beta \in \mathbb{R}^D$ is the waveform when there is no signal, whereas $\alpha \in \mathbb{R}^D$ is the signal envelope. The complete model is the sum $\alpha Y + \beta$, where Y is the Hamming weight of some intermediate variable (such as the XOR operation $T \oplus k$) on $n = 4$ bits. While the leakage signal may be represented as a continuous curve as illustrated in Fig. 1, the practical acquisition consists in a temporal series of D "discrete samples", typically within one clock period. For $S = 2$, we thus write (1) as

$$\mathbf{X} = \alpha \mathbf{Y}^\star + \beta \mathbf{1} + \mathbf{N} \tag{2}$$

where $\mathbf{X}$ is $D \times Q$, α and β are $D \times 1$, $\mathbf{Y}^\star$ and $\mathbf{1} = (1, \ldots, 1)$ are $1 \times Q$, and $\mathbf{N}$ is $D \times Q$. Here $\mathbf{Y}$ is assumed centered: $\mathbb{E}(\mathbf{Y}) = \mathbf{0} = (0, \ldots, 0)$ (since the non-centered part is captured by the $\beta\mathbf{1}$ term) and of unit variance for every q: $\mathrm{Var}(Y_q) = \mathbb{E}(Y_q^2) = 1$.

For $S \geq 2$, the actual value of S reflects the complexity of the model. For example, in the *weighted sum of bits* model, the model for each trace can be written as $\sum_{s=1}^{n} \alpha_s Y_s + \beta$, where Y_s is the sth bit of the n-bit sensitive variable Y. Accordingly, we have

$$S = n + 1, \qquad \text{and thus:}$$
$$\alpha = \Big(\alpha_1 \ldots \alpha_n\ \beta\Big), \qquad \mathbf{Y} = (\mathbf{Y}_1 \ldots \mathbf{Y}_n\ \mathbf{1})^\mathsf{T}. \tag{3}$$

This leakage model is more complex than before but may arise in practice. For example, Fig. 2 plots the coefficients $\alpha_1, \ldots, \alpha_8$ estimated of the traces taken from an ATMega smartcard—the datasets are available from the DPA contest V4 team [**?**]. In particular one can observe that samples around $[50, 80]$ are ordered by Hamming weight: this part of the trace resembles the upper left part of Fig. 1 for $S = 2$. By analysing the $(n+1)$-variate model of (3), one can indeed see that around $[50, 80]$, the vectors $\alpha_1, \ldots, \alpha_8$ are almost identical. However, samples in intervals $[170, 250]$ or $[330, 400]$ have a more complex model. These times, the eight vectors $\alpha_1, \ldots, \alpha_8$ are clearly different, so the leakage is 9-variate.

In the sequel, we consider both types of attacks: those with *offline profiling* where α for each component of the model is precharacterized like in Fig. 2 and also those where the model is *learned online* like in a Linear Regression Attack [**?**].

3 Theoretical Results and Implementation

3.1 General Mathematical Expressions

In this section we derive the mathematical expression of the optimal distinguisher $\mathcal{D}$ in the general case of multivariate leakage ($D \geq 1$), and multiple models ($S \geq 1$).

[7] We underline that $\mathbf{Y}^\star$ denotes the model for the correct key; we use $\mathbf{Y}(k)$ for a model assuming a guessed key k. Sometimes, in a view to make notations more legible, the dependence in k is tacit and $\mathbf{Y}(k)$ simply writes as $\mathbf{Y}$.

[8] We may simplify (2) by incorporating $\beta\mathbf{1}$ into the noise expectation, but the noise is intrinsically zero-mean and it is clearer to exhibit a specific offset term.

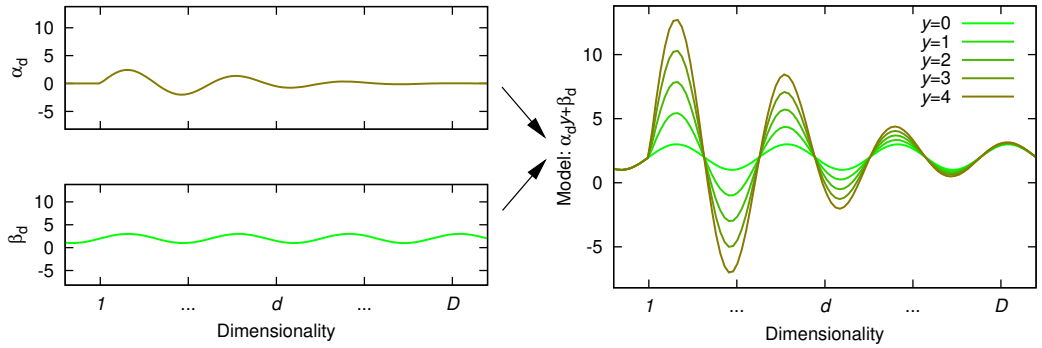

Fig. 1 Example of leakage model with $S = 2$ and a model in Hamming weight, with $n = 4$ values (no noise is added)

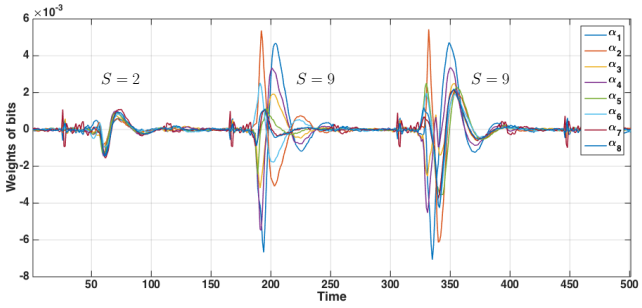

Fig. 2 Leakage evaluation of traces from DPA contest V4 (knowing the mask)

An illustration of our results is given in Fig. 3 for the case when the leakage is completely known (or profiled as in the template attack) and when the leakage is unknown and estimated online.

Definition 1 (Optimal Distinguisher Knowing or Ignoring α)

$$\mathcal{D}_{\mathrm{ML}}(\mathbf{x},\mathbf{t}) = \operatorname*{argmax}_{k\in\mathbb{F}_2} p(\mathbf{x}|\mathbf{t}) \qquad \text{and}$$
$$\mathcal{D}_{\mathrm{ML,sto}}(\mathbf{x},\mathbf{t}) = \operatorname*{argmax}_{k\in\mathbb{F}_2} \max_{\alpha\in\mathbb{R}^{D\times S}} p(\mathbf{x}|\mathbf{t},\alpha).$$

In both cases (Theorems 1 and 2 below) the result is a distinguisher which is computed using simple matrix operations. While $\mathcal{D}_{\mathrm{ML}}$ resembles a template attack with Gaussian templates [**?**], $\mathcal{D}_{\mathrm{ML,sto}}$ is a novel expression that results from a non-trivial maximization over the matrix α and may be interpreted as a generalization of a multivariate correlation power attack [**?**].

Theorem 1 *The optimal* maximum likelihood *(ML) distinguisher [?] for Gaussian noise writes*

$$\mathcal{D}_{\mathrm{ML}}(\mathbf{x},\mathbf{t}) = \operatorname*{argmin}_{k} \ \operatorname{tr}\Big((\mathbf{x}-\alpha\mathbf{y})^{\mathsf{T}}\Sigma^{-1}(\mathbf{x}-\alpha\mathbf{y})\Big). \quad (4)$$

Proof From [**?**] we have $\mathcal{D}_{\mathrm{ML}}(\mathbf{x},\mathbf{t}) = \operatorname{argmax}_k \ p(\mathbf{x}|\mathbf{y})$ while from (1) we see that $p(\mathbf{x}|\mathbf{y}) = p_{\mathbf{N}}(\mathbf{x}-\alpha\mathbf{y})$. From the i.i.d. assumption the noise density $p_{\mathbf{N}}(\mathbf{n})$ is given by

$$\begin{aligned} p_{\mathbf{N}}(\mathbf{n}) &= \prod_{q=1}^{Q} \frac{1}{\sqrt{(2\pi)^D|\det\Sigma|}} \exp -\frac{1}{2} n_q{}^{\mathsf{T}}\Sigma^{-1}n_q \\ &= \frac{1}{(2\pi)^{DQ/2}} \frac{1}{(\det\Sigma)^{Q/2}} \exp -\frac{1}{2}\Big(\sum_{q=1}^{Q} n_q{}^{\mathsf{T}}\Sigma^{-1}n_q\Big) \\ &= \frac{1}{(2\pi)^{DQ/2}(\det\Sigma)^{Q/2}} \exp -\frac{1}{2}\operatorname{tr}\left(\mathbf{n}^{\mathsf{T}}\Sigma^{-1}\mathbf{n}\right). \end{aligned}$$

Thus $p_{\mathbf{N}}(\mathbf{x}-\alpha\mathbf{y})$ is maximum when the expression $\operatorname{tr}\left(\mathbf{n}^{\mathsf{T}}\Sigma^{-1}\mathbf{n}\right)$ for $\mathbf{n} = \mathbf{x}-\alpha\mathbf{y}$ is minimum. □

In Eqn. (4) of Theorem 1, the trace

$$\operatorname{tr}\Big((\mathbf{x}-\alpha\mathbf{y})^{\mathsf{T}}\Sigma^{-1}(\mathbf{x}-\alpha\mathbf{y})\Big)$$

consists in:

- the sum of Q Mahalanobis [**?**] distances (see also Eqn. (22) of [**?**]),
- the sum of D elements (which is useful when $D \ll Q$), as attested by rewriting

$$\operatorname{tr}\Bigg(\underbrace{(\mathbf{x}-\alpha\mathbf{y})^{\mathsf{T}}\Sigma^{-1}(\mathbf{x}-\alpha\mathbf{y})}_{Q\times Q \text{ matrix}}\Bigg) = \operatorname{tr}\Bigg(\underbrace{\Sigma^{-1}(\mathbf{x}-\alpha\mathbf{y})(\mathbf{x}-\alpha\mathbf{y})^{\mathsf{T}}}_{D\times D \text{ matrix}}\Bigg).$$

Theorem 2 *The optimal stochastic multivariate attack is given by*

$$\mathcal{D}_{\mathrm{ML,sto}}(\mathbf{x},\mathbf{t}) = \operatorname*{argmax}_{k\in\mathbb{F}_2^n} \ \operatorname{tr}\left(\mathbf{y}^{\mathsf{T}}(\mathbf{y}\mathbf{y}^{\mathsf{T}})^{-1}\mathbf{y} \ \ \mathbf{x}^{\mathsf{T}}\Sigma^{-1}\mathbf{x}\right) \quad (5)$$

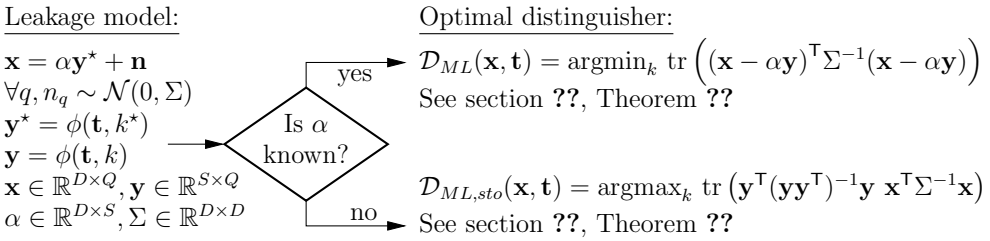

Fig. 3 Mathematical expression for multivariate ($D \geq 1$) optimal attacks with a linear combination of models ($S \geq 1$)

for which the optimal value of α is given by

$$\alpha^{opt} = (\mathbf{x}\mathbf{y}^\mathsf{T})(\mathbf{y}\mathbf{y}^\mathsf{T})^{-1}. \tag{6}$$

For the proof, we need some known results of linear algebra (Lemma 1) and linear regression (Lemma 2).

Lemma 1 *Let $\mathbf{b}$ an $S \times Q$ matrix, with $S < Q$. The $S \times S$ matrix $\mathbf{b}\mathbf{b}^\mathsf{T}$ is invertible if and only if $\mathbf{b}$ has full rank S, i.e., if and only if the S lines of $\mathbf{b}$ are independent.*

Proof Let x be a $S \times 1$ column vector. We have that $x^\mathsf{T}\mathbf{b}\mathbf{b}^\mathsf{T}x = \|\mathbf{b}^\mathsf{T}x\|^2 = 0$ implies $\mathbf{b}^\mathsf{T}x = 0$ hence $x = 0$. Hence the matrix $\mathbf{b}\mathbf{b}^\mathsf{T}$ is positive definite. □

Lemma 2 *Let $\mathbf{a}$, $\mathbf{b}$ and α be respectively $1 \times Q$, $S \times Q$ and $1 \times S$ with $S < Q$, where $\mathbf{b}$ has full rank S. Then $\|\mathbf{a} - \alpha\mathbf{b}\|_2$ reaches its minimum for $\alpha = \mathbf{a}\mathbf{b}^\mathsf{T}(\mathbf{b}\mathbf{b}^\mathsf{T})^{-1}$.*

Proof Expanding the squared norm gives $\|\mathbf{a} - \alpha\mathbf{b}\|_2^2 = (\mathbf{a} - \alpha\mathbf{b})(\mathbf{a} - \alpha\mathbf{b})^\mathsf{T} = \mathbf{a}\mathbf{a}^\mathsf{T} - 2\alpha\mathbf{b}\mathbf{a}^\mathsf{T} + \alpha\mathbf{b}\mathbf{b}^\mathsf{T}\alpha^\mathsf{T}$. Therefore, the gradient $\frac{\partial}{\partial\alpha}\|\mathbf{a} - \alpha\mathbf{b}\|_2^2 = -2\mathbf{b}\mathbf{a}^\mathsf{T} + 2\mathbf{b}\mathbf{b}^\mathsf{T}\alpha^\mathsf{T}$ vanishes if and only if $\alpha^\mathsf{T} = (\mathbf{b}\mathbf{b}^\mathsf{T})^{-1}\mathbf{b}\mathbf{a}^\mathsf{T}$, i.e., $\alpha = \mathbf{a}\mathbf{b}^\mathsf{T}(\mathbf{b}\mathbf{b}^\mathsf{T})^{-1}$ where we have used the fact that $\mathbf{b}\mathbf{b}^\mathsf{T}$ is invertible by Lemma 1. □

*Proof (***Proof of Theorem 2)** Let $\mathbf{x}' = \Sigma^{-1/2}\ \mathbf{x}$ and $\mathbf{y}' = (\mathbf{y}\mathbf{y}^\mathsf{T})^{-1/2}\ \mathbf{y}$. The optimal distinguisher minimizes the following expression over $\alpha \in \mathbb{R}^{D\times S}$:

$$\begin{aligned}&\text{tr}\left((\mathbf{x} - \alpha\mathbf{y})^\mathsf{T}\Sigma^{-1}(\mathbf{x} - \alpha\mathbf{y})\right)\\ &= \text{tr}\left((\mathbf{x}' - \alpha'\mathbf{y})(\mathbf{x}' - \alpha'\mathbf{y})^\mathsf{T}\right) = \sum_{d=1}^{D}\|\mathbf{x}'_d - \alpha'_d\mathbf{y}\|^2.\end{aligned}$$

By Lemma 2 the minimization over α'_d yields $\alpha'_d = (\mathbf{x}'_d\mathbf{y}^\mathsf{T})(\mathbf{y}\mathbf{y}^\mathsf{T})^{-1}$ for all $d = 1, \ldots, D$. This gives $\alpha' = (\mathbf{x}'\mathbf{y}^\mathsf{T})(\mathbf{y}\mathbf{y}^\mathsf{T})^{-1}$ hence $\alpha = (\mathbf{x}\mathbf{y}^\mathsf{T})(\mathbf{y}\mathbf{y}^\mathsf{T})^{-1}$, which remarkably does *not* depend on Σ.

The minimized value of the distinguisher is thus

$$\begin{aligned}&\min_\alpha\ \text{tr}\left((\mathbf{x} - \alpha\mathbf{y})^\mathsf{T}\Sigma^{-1}(\mathbf{x} - \alpha\mathbf{y})\right)\\ &= \text{tr}\left((\mathbf{x} - \alpha^{\text{opt}}\mathbf{y})^\mathsf{T}\Sigma^{-1}(\mathbf{x} - \alpha^{\text{opt}}\mathbf{y})\right)\\ &= \text{tr}\left((\mathsf{Id} - \mathbf{y}^\mathsf{T}(\mathbf{y}\mathbf{y}^\mathsf{T})^{-1})^2\mathbf{x}^\mathsf{T}\Sigma^{-1}\mathbf{x}\right)\\ &= \text{tr}\left(\mathbf{x}^\mathsf{T}\Sigma^{-1}\mathbf{x}\right) - \text{tr}\left(\mathbf{y}^\mathsf{T}(\mathbf{y}\mathbf{y}^\mathsf{T})^{-1}\ \mathbf{x}^\mathsf{T}\Sigma^{-1}\mathbf{x}\right)\end{aligned}$$

where Id denotes the $D \times D$ identity matrix and where $\text{tr}\left(\mathbf{x}^\mathsf{T}\Sigma^{-1}\mathbf{x}\right)$ is a constant independent of k. This proves Theorem 2. □

The expression of $\mathcal{D}_{\text{ML,sto}}(\mathbf{x}, \mathbf{t})$ given in Theorem 2 consists in the trace of a $Q \times Q$ matrix, which can be admittedly very large. It can be, however, rewritten in a way that is easier to compute when Q is much greater than S and D:

Corollary 1 (Alternative Expression of $\mathcal{D}_{\text{ML,sto}}$) *Letting $\mathbf{x}' = \Sigma^{-1/2}\ \mathbf{x}$, and $\mathbf{y}' = (\mathbf{y}\mathbf{y}^\mathsf{T})^{-1/2}\ \mathbf{y}$ as in the proof of Theorem 2, we have*

$$\mathcal{D}_{\text{ML,sto}}(\mathbf{x}, \mathbf{t}) = \underset{k\in\mathbb{F}_2^n}{\text{argmax}}\ \|\mathbf{x}'\mathbf{y}'^\mathsf{T}\|_F. \tag{7}$$

Here the Frobenius norm is of a $D \times S$ matrix.

Proof Let us write $(\mathbf{y}\mathbf{y}^\mathsf{T})^{-1} = (\mathbf{y}\mathbf{y}^\mathsf{T})^{-1/2}(\mathbf{y}\mathbf{y}^\mathsf{T})^{-1/2}$ in (5). By the properties of the trace,

$$\begin{aligned}&\text{tr}\left(\mathbf{y}^\mathsf{T}(\mathbf{y}\mathbf{y}^\mathsf{T})^{-1}\mathbf{y}\ \ \mathbf{x}^\mathsf{T}\Sigma^{-1}\mathbf{x}\right)\\ &= \text{tr}\left(\underbrace{\left((\mathbf{y}\mathbf{y}^\mathsf{T})^{-\frac{1}{2}}\mathbf{y}(\Sigma^{-\frac{1}{2}}\mathbf{x})^\mathsf{T}\right)}_{S\times D}\underbrace{\left((\mathbf{y}\mathbf{y}^\mathsf{T})^{-\frac{1}{2}}\mathbf{y}(\Sigma^{-\frac{1}{2}}\mathbf{x})^\mathsf{T}\right)^\mathsf{T}}_{D\times S}\right)\\ &= \text{tr}\left((\mathbf{y}'\mathbf{x}'^\mathsf{T})(\mathbf{y}'\mathbf{x}'^\mathsf{T})^\mathsf{T}\right) = \|\mathbf{x}'\mathbf{y}'^\mathsf{T}\|_F^2.\end{aligned}$$ □

Remark 1 Notice that in corollary 1, $\mathbf{y}'$ is a vector of empirical covariance equal to the identity matrix. Indeed, $\mathbf{y}'\mathbf{y}'^\mathsf{T} = (\mathbf{y}\mathbf{y}^\mathsf{T})^{-1/2}\mathbf{y}\mathbf{y}^\mathsf{T}(\mathbf{y}\mathbf{y}^\mathsf{T})^{-1/2} = \mathsf{Id}$.

3.2 Mathematical Expressions for $S = 2$

In order to provide interpretations for the optimal distinguisher expressions, we detail how an optimal attack unfolds when the leakage consists in a sum of a modulated scalar model and an offset ($S = 2$). The cases for profiled attacks (denoted $\mathcal{D}_{\mathrm{ML}}^{S=2}$) and non-profiled attacks (denoted $\mathcal{D}_{\mathrm{ML,sto}}^{S=2}$) are presented in Fig. 4.

Interestingly, when $S = 2$, the template attack can decompose in two steps (affine projection followed by a Euclidean distance to the model). Remarkably, the projection vector is the same for all key guesses. This extends similar results [**?**] where only the linear relationship between leakage and model is explored. As for the online attack, $\mathcal{D}_{\mathrm{ML,sto}}^{S=2}$ consists in a sum of square of CPA attacks on transformed data, aiming at orthogonalizing the noise.

3.3 Efficient Implementation

Both $\mathcal{D}_{\mathrm{ML}}$ and $\mathcal{D}_{\mathrm{ML,sto}}$ can be optimized using the idea presented in [**?**]. This article applies a precomputation step in the case the number of traces is larger than the number of possible plaintexts ($Q > \#\mathcal{T} = 2^n$). In this case, all summations $\sum_q$ can be advantageously replaced by $\sum_t \sum_{t_q=t}$. In most cases, the sum $\sum_{t_q=t}$ can be achieved on the fly, and does not involve an hypothesis on the key. Therefore, a speed gain of 2^n (the cardinality of the key space) is expected.

Such optimization strategy can be applied to $\mathcal{D}_{\mathrm{ML}}$. Indeed, let us define $\mathbf{x}' = \Sigma^{-1/2}\mathbf{x}$ and $\alpha' = \Sigma^{-1/2}\alpha$. Then,

$$\mathcal{D}_{\mathrm{ML}}(\mathbf{x},\mathbf{t}) = \operatorname*{argmin}_k \sum_{d=1}^{D} \|\mathbf{x}'_d - \alpha'_d \mathbf{y}\|_2^2 \qquad \text{(see Corollary 1)}$$

$$= \operatorname*{argmin}_k \sum_{d=1}^{D} \sum_{t\in\mathbb{F}_2^n} \left(\sum_{q/t_q=t} {x'_{d,q}}^2 - 2 \sum_{q/t_q=t} x'_{d,q}\alpha'_d y(t,k) + \Big(\sum_{q/t_q=t} 1\Big)(\alpha'_d y(t,k))^2 \right)$$

$$= \operatorname*{argmin}_k \sum_{d=1}^{D} \sum_{t\in\mathbb{F}_2^n} -2 \underbrace{\Big(\sum_{q/t_q=t} x'_{d,q} \Big)}_{\text{denoted as } x'_{d,t}} \alpha'_d y(t,k) + \underbrace{\Big(\sum_{q/t_q=t} 1 \Big)}_{\text{denoted as } n_t} (\alpha'_d y(t,k))^2 \tag{8}$$

$$= \operatorname*{argmax}_k \ \mathrm{tr}\left(\mathbf{x}' (\alpha' \mathbf{y}(k))^{\mathsf{T}} \right) - \frac{1}{2} \sum_{t\in\mathbb{F}_2^n} n_t \ \|\alpha' y(t,k)\|_2^2 \,. \tag{9}$$

Notice that at line (8), the term $\sum_{q/t_q=t} {x'_{d,q}}^2$ which does not depend on the key, is simplified. The fast version of this computation is given in Alg. 1.

The same optimization applies to $\mathcal{D}_{\mathrm{ML,sto}}$. Indeed, in expression (7) of $\mathcal{D}_{\mathrm{ML,sto}}(\mathbf{x},\mathbf{t}) = \operatorname{argmax}_k \ \|\mathbf{x}'\mathbf{y}'^{\mathsf{T}}\|_F^2$,

input : $\mathbf{x}$, $\mathbf{t}$
output: $\mathcal{D}_{\mathrm{ML}}(\mathbf{x},\mathbf{t})$

// Initialize to zero a matrix $x'_{d,t}$ of size $D \times 2^n$
// Initialize to zero a vector n_t of length 2^n
1 **for** $q \in \{1,\dots,Q\}$ **do** // On-the-fly accumulation
2 $\quad x'_{t_q} \leftarrow x'_{t_q} + \Sigma^{-1/2} x_q$
3 $\quad n_{t_q} \leftarrow n_{t_q} + 1$
4 **return** // Single evaluation, as in (9)
$\operatorname{argmax}_{k\in\mathcal{K}} \ \mathrm{tr}\left(\mathbf{x}'(\alpha'\mathbf{y}(k))^{\mathsf{T}}\right) - \frac{1}{2}\sum_t n_t \ \|\alpha' y(t,k)\|_2^2$

Algorithm 1: Fast computation algorithm for $\mathcal{D}_{\mathrm{ML}}$

one can write

$$\|\mathbf{x}'\mathbf{y}'^{\mathsf{T}}\|_F^2 = \sum_{s,d} \Big(\sum_{q=1}^{Q} x'_{d,q} y'_{s,q} \Big)^2$$

$$= \sum_{s,d} \Big(\sum_{t\in\mathbb{F}_2^n} \underbrace{\Big(\sum_{q/t_q=t} x'_{d,t} \Big)}_{\text{denoted as } x'_{d,t}} \underbrace{\Big(y'_s(t,k) \Big)}_{\text{denoted as } y'_{s,t}} \Big)^2 . \tag{10}$$

This means that $\mathbf{x}'$ can be obtained by simple accumulation, exactly as in line 2 of Alg. 1. The term $y'_s(t,k)$ requires the computation of $\mathbf{y}\mathbf{y}^{\mathsf{T}}$. In the case $Q \gg 1$, it can be assumed that the texts $\mathbf{t}$ are uniformly distributed. Hence, when $Q \to +\infty$, by the law of large numbers,

$$\frac{1}{Q}\mathbf{y}\mathbf{y}^{\mathsf{T}} = \frac{1}{Q}\sum_{q=1}^{Q} y_q y_q^{\mathsf{T}} = \sum_{t\in\mathbb{F}_2^n} \frac{\sum_{q/t_q=t} 1}{Q} y(t,k) y(t,k)^{\mathsf{T}}$$

$$\xrightarrow[Q\to+\infty]{} \frac{1}{2^n} \sum_{t\in\mathbb{F}_2^n} y(t,k) y(t,k)^{\mathsf{T}} .$$

Therefore, in (10), $y'_s(t)$ can also be precomputed. To the best of our knowledge, this optimization has never been discussed previously. The resulting distinguishing procedure is given in Alg. 2. At line 3, the argument of the Frobenius norm can be computed by a fast matrix multiplication. Also, we notice that the matrix inversion at line 0 is actually a precomputation which involves only the model. Besides, if the EIS (*Equal Images under all Sub-keys*) assumption holds [**?**, Def. 2], e.g., $y(t,k)$ only depends on $t \oplus k$, then $\sum_t y(t,k)y(t,k)^{\mathsf{T}}$ does not depend on k, hence only one single matrix inversion to compute. Eventually, the computational complexity of the optimal stochastic attack simply consists in traces accumulation per class, and as many matrix products and Frobenius norms as keys to be guessed.

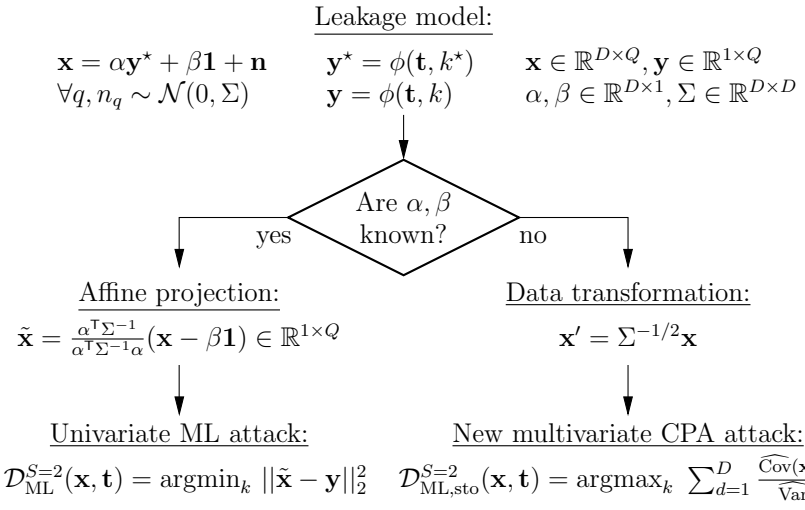

Fig. 4 Modus operandi for multivariate ($D \geq 1$) optimal attacks with one model $\mathbf{Y}$ associated to envelope $\alpha \in \mathbb{R}^{D\times 1}$ and a constant offset $\beta \in \mathbb{R}^{D\times 1}$ ($S = 2$)

input : $\mathbf{x}$, $\mathbf{t}$
output: $\mathcal{D}_{\mathrm{ML,sto}}(\mathbf{x}, \mathbf{t})$

0 // Precompute $\#\mathcal{K} = 2^n$ matrices $y'(k)$ of size $S \times 2^n$, s.t. $y'(k) = (\frac{1}{2^n}\sum_t y(t,k)y(t,k)^\mathsf{T})^{-1/2}y(k)$.
// Initialize to zero a matrix $x'_{d,t}$ of size $D \times 2^n$
1 **for** $q \in \{1, \ldots, Q\}$ **do**
2 $\quad x'_{t_q} \leftarrow x'_{t_q} + \Sigma^{-1/2}x_q$ // In-place accumulation of a column in matrix $\mathbf{x}'$
3 **return** $\mathrm{argmax}_{k\in\mathcal{K}} \ \|\mathbf{x}'\mathbf{y}'(k)^\mathsf{T}\|_F$ // As in (10)

Algorithm 2: Fast computation algorithm for $\mathcal{D}_{\mathrm{ML,sto}}$ when $\mathbf{t}$ is balanced

4 Practical Results

4.1 Characterization of Σ

In this article, we assume that the attacker knows the noise covariance matrix. We give a straightforward procedure for the estimation.

1. collect Q traces (i.e., a matrix $\mathbf{x} \in \mathbb{R}^{D\times Q}$) where the plaintext is fixed to a given value,
2. estimate Σ as $\hat{\Sigma} = \frac{1}{Q-1}\big(\mathbf{x} - \frac{1}{Q}\mathbf{x}\mathbf{1}^\mathsf{T}\mathbf{1}\big)\big(\mathbf{x} - \frac{1}{Q}\mathbf{x}\mathbf{1}^\mathsf{T}\mathbf{1}\big)^\mathsf{T}$, where $\mathbf{1} = (1, \ldots, 1) \in \mathbb{R}^{1\times Q}$. This estimator is *sample covariance matrix*, which is unbiased.

Remark 2 Notice that Σ cannot be obtained by a direct profiling on the same traces to be used for the attack. Indeed, in those traces, the plaintext is varying, hence the attacker would use for $\hat{\Sigma}$ the covariance matrix of $\mathbf{x} - \alpha^{\mathrm{opt}}\mathbf{y}$, where α^{opt} is equal to $\alpha^{\mathrm{opt}} = (\mathbf{x}\mathbf{y}^\mathsf{T})(\mathbf{y}\mathbf{y}^\mathsf{T})^{-1}$ (recall (6)). Hence, $\hat{\Sigma} = \frac{1}{Q-1}(\mathbf{x} - \alpha^{\mathrm{opt}}\mathbf{y})(\mathbf{x} - \alpha^{\mathrm{opt}}\mathbf{y})^\mathsf{T}$. But the distinguisher $\mathcal{D}_{\mathrm{ML,sto}}$ is

$$
\begin{aligned}
&\mathcal{D}_{\mathrm{ML,sto}}(\mathbf{x}, \mathbf{t}) \\
&= \mathrm{argmin}_{k\in\mathbb{F}_2^n} \min_{\alpha\in\mathbb{R}^{D\times S}} \mathrm{tr}\left((\mathbf{x} - \alpha\mathbf{y})^\mathsf{T}\hat{\Sigma}^{-1}(\mathbf{x} - \alpha\mathbf{y})\right) \\
&= \mathrm{argmin}_{k\in\mathbb{F}_2^n} \min_{\alpha\in\mathbb{R}^{D\times S}} \mathrm{tr}\left(\hat{\Sigma}^{-1}(\mathbf{x} - \alpha\mathbf{y})(\mathbf{x} - \alpha\mathbf{y})^\mathsf{T}\right) \\
&= \mathrm{argmin}_{k\in\mathbb{F}_2^n} \mathrm{tr}\left(\hat{\Sigma}^{-1}(\mathbf{x} - \alpha^{\mathrm{opt}}\mathbf{y})(\mathbf{x} - \alpha^{\mathrm{opt}}\mathbf{y})^\mathsf{T}\right) \qquad (11) \\
&= \mathrm{argmin}_{k\in\mathbb{F}_2^n} \mathrm{tr}\left((Q-1)\hat{\Sigma}^{-1}\hat{\Sigma}\right) = \mathrm{argmin}_{k\in\mathbb{F}_2^n} D(Q-1). \qquad (12)
\end{aligned}
$$

Indeed, at line (11), we demonstrated in the proof of Theorem 2 in that the minimal value (6) of α is independent on Σ. Eventually, it can be seen at line (12) that the distinguisher with $\hat{\Sigma}$ instead of Σ does not depend on the key[9].

4.2 Attacks on Synthetic (i.e., Simulated) Traces

In this subsection we present simulations when α is known exactly or regressed online. We consider an attack of PRESENT, where the SBox is $n = 4 \to n = 4$. For the sake of the simulations, we choose two kinds of α:

[9] Indeed, $\mathrm{argmin}_k \, \mathrm{cst} = \mathbb{F}_2^n$, meaning that all keys are equiprobable. Intuitively, when both the noise and the model parameters are regressed at the same time, any key manages to achieve the same match between parametric model and side-channel observations.

– "identical": all the $n = 4$ bits leak the same waveform, like in the Hamming weight model,
– "proportional": the waveform has weight 1 for SBox bit 0, and is multiplied by 2 (resp. 3 and 4) for SBox bit 1 (resp. 2 and 3).

The waveform for each bit is that represented in Fig. 1 (upper left graph). Specifically, for all $1 \leq d \leq D$ and $1 \leq s \leq S$, the envelope consists in damped oscillations:

$$\alpha_{d,s} = e^{-\frac{2d}{D}} \cos\big(2\pi \frac{d}{D}\big) \text{ for the "identical" case,} \tag{13}$$

$$\alpha_{d,s} = s \cdot e^{-\frac{2d}{D}} \cos\big(2\pi \frac{d}{D}\big) \text{ for the "proportional" case.} \tag{14}$$

The noise is chosen normal, using two distributions:

– "isotropic": the covariance matrix is σ^2 times the $D \times D$ identity,
– "auto-regressive" (of "AR" for short): the covariance matrix element at position (d, d'), for $1 \leq d, d' \leq D$, is $\sigma^2 \rho^{|d-d'|}$. This noise is not independent from sample to its neighbours, but the correlation ρ decreases exponentially as samples get further apart.

Proposition 1 *The success probability of $\mathcal{D}_{\mathrm{ML}}$ is greater than that of $\mathcal{D}_{\mathrm{ML,sto}}$.*

Proof Indeed, according to [**?**], $\mathcal{D}_{\mathrm{ML}}$ maximizes the success probability. Thus, the distinguisher $\mathcal{D}_{\mathrm{ML,sto}}$ has a smaller success probability. The success probability is the same if the minimization over α in the proof of Theorem 2 yields the exact matrix α used in the model (1). □

Simulations allow to estimate the loss in terms of efficiency of not knowing the model (Proposition 1), by comparing distinguishers $\mathcal{D}_{\mathrm{ML}}$ ((4)) and $\mathcal{D}_{\mathrm{ML,sto}}$ ((5)). The success rate of the optimal distinguisher $\mathcal{D}_{\mathrm{ML}}$ is drawn in order to materialize the limit between feasible (below) and unfeasible (above) attacks.

Results for low noise ($\sigma = 1$) are represented in Fig. 5. We can see that the Hamming weight model is clearly harder to attack, because the leakage of one bit cannot be distinguished from that of the other bits. Besides, we notice that the stochastic attack is performing much worse than the optimal attack: about 10 times more traces are required for an equivalent success probability in key extraction.

Results for high noise ($\sigma = 4$) are represented in Fig. 6. Again, the "proportional" model is easier to attack than the "identical" model (for each bit). Now, we also see that the gap between the optimal ML attack and the stochastic attack narrows: only about 5 times more traces are needed for the stochastic attack to perform as well as the optimal attack in terms of success probability. Besides, we notice that the AR noise is favorable to the attacker. It is therefore important in practice for the attacker to characterize precisely the noise distribution (recall the methodology presented in Sec. 4.1).

Clearly, these conclusions are in line with the *template* versus *stochastic* (offline) study carried out in [**?**]: for high noise, the (online) learning of the model requires more traces, hence is more accurate. Therefore, the performance of $\mathcal{D}_{\mathrm{ML,sto}}$ gets closer to that of $\mathcal{D}_{\mathrm{ML}}$ than for high noise.

4.3 Attacks on Real-World Traces

We now compare CPA with $\mathcal{D}_{\mathrm{ML}}$ and $\mathcal{D}_{\mathrm{ML,sto}}$ on measurements provided by the DPA contest V4. These traces have have been acquired from an 8-bit processor, hence have a signal-to-noise ratio greater than one, reaching 7 at some points in time. The interval for our case-study is $[170, 250]$ from Fig. 2, hence $D = 80$. Regarding ML, two learning strategies have been implemented:

1. the model is learned from a disjoint set of 5k traces, which is the operational scenario for a profiled attack;
2. the model is learned from the traces being attacked (denoted **self** in Fig. 7). This case does not represent a realistic attack, but is interesting in that it highlights the best possible attacker.

The attack success rates are plotted in Fig. 7. One can see that both variants of $\mathcal{D}_{\mathrm{ML}}$ and $\mathcal{D}_{\mathrm{ML,sto}}$ achieve better with $S = 9$ than with $S = 2$. This is consistent with the analysis carried out in Sec. 2.3. Actually, the CPA has a very poor performance because the model is actually very far from a Hamming weight: as can be seen in Fig. 2.3(a), some parameters α_i (e.g., for $i = 2$ and 6) are positive in region $[180, 200]$ whereas others α_j (e.g., for $j = 1$, 3, 4 and 5) are negative. The compensating signs account why the Hamming weight model is inappropriate. The ML with model pre-characterization on the traces under attack show that very strong attacks are possible (using a few traces only). Interestingly, when the model used by ML is characterized on 5k traces distinct from the traces being attacked, the performance is almost similar. Eventually, the online stochastic attack derived in this paper ($\mathcal{D}_{\mathrm{ML,sto}}$) performs better than CPA (the distinguisher being the maximum value of the Pearson correlation over the $D = 80$ samples).

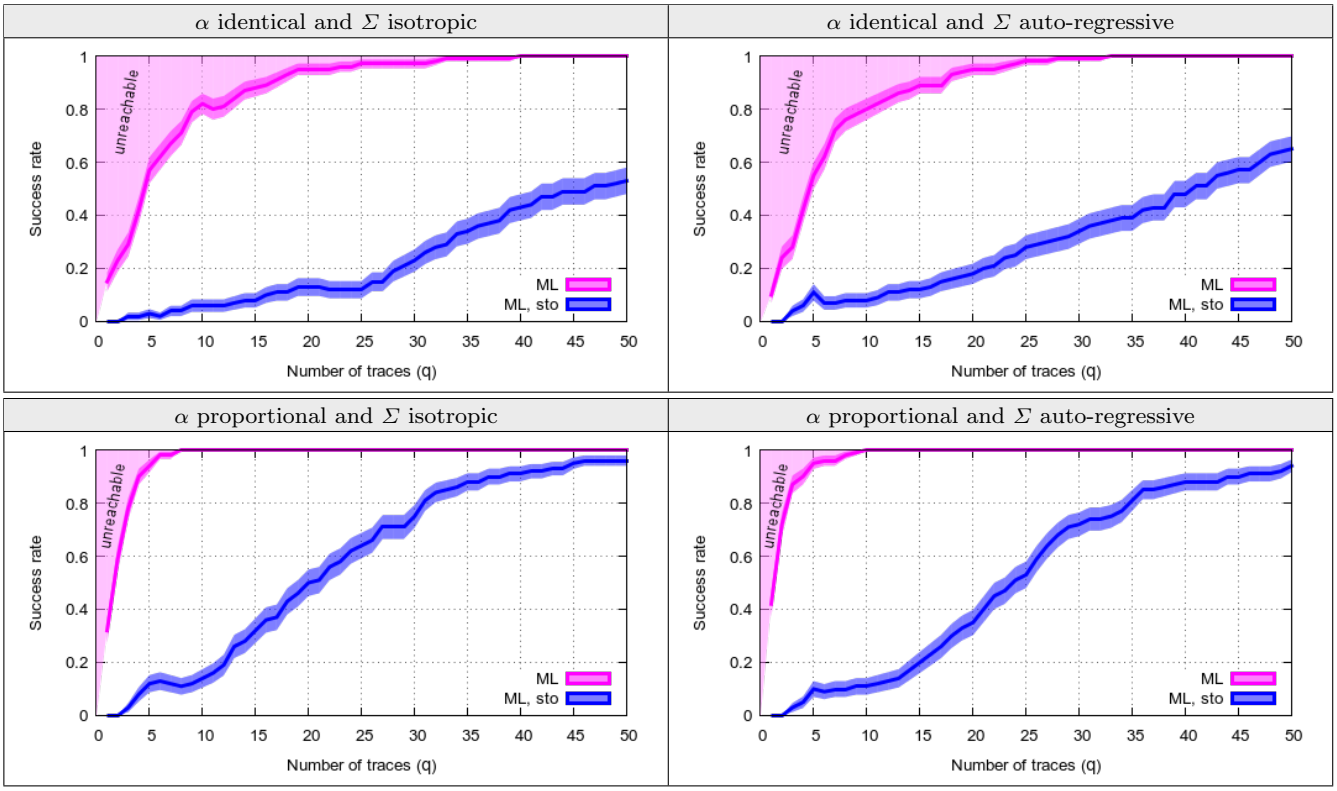

Fig. 5 Simulations for $D = 3$, $S = 5$, $n = 4$, $\sigma = 1$ (AR noise with $\rho = 0.5$).

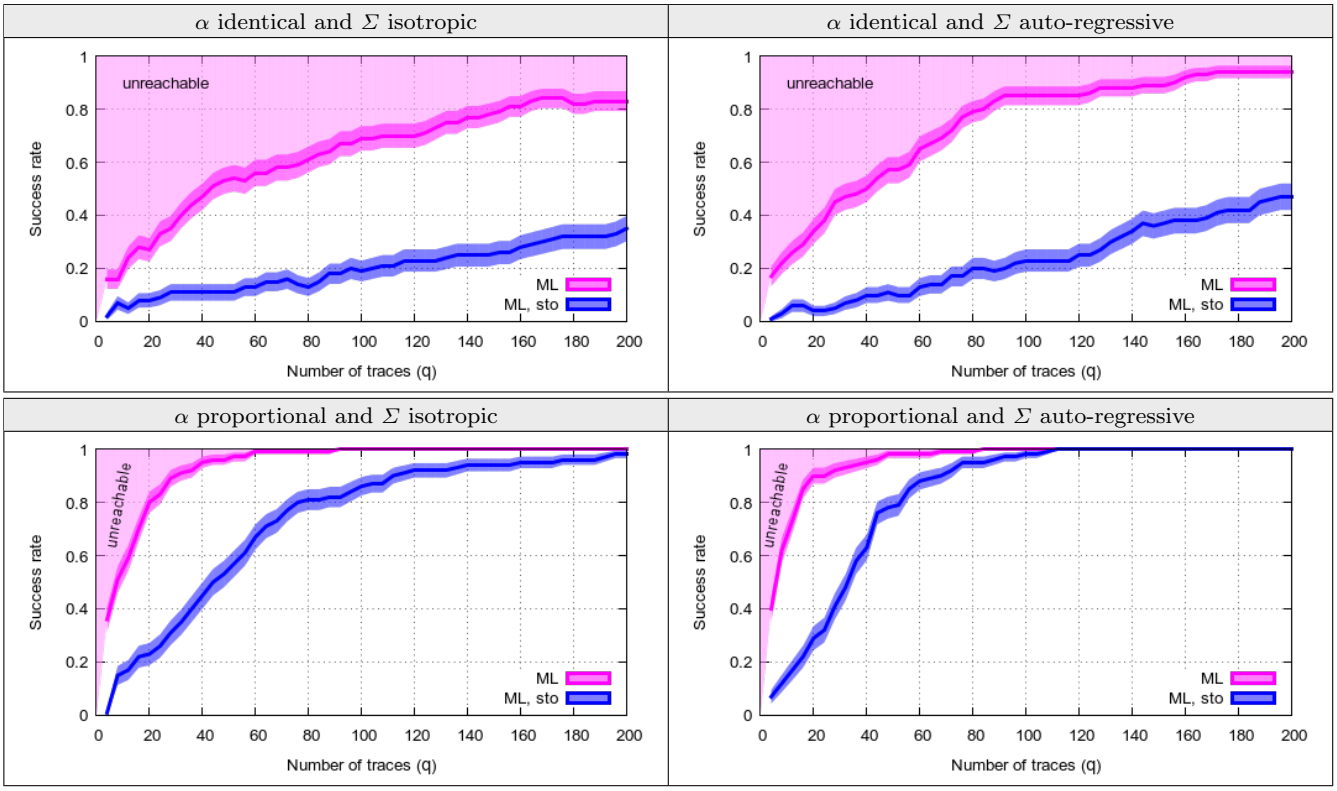

Fig. 6 Simulations for $D = 3$, $S = 5$, $n = 4$, $\sigma = 4$ (AR noise with $\rho = 0.5$).

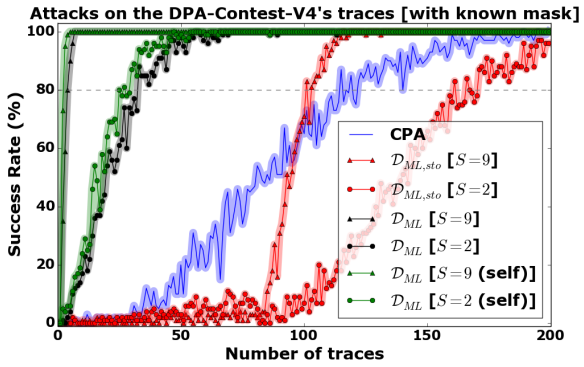

Fig. 7 Comparison of success rate of CPA, $\mathcal{D}_{\mathrm{ML,sto}}$ for $S \in \{9, 2\}$, and $\mathcal{D}_{\mathrm{ML}}$ for $S \in \{9, 2\}$ (with two distinct learning methods)

5 Conclusions and Perspectives

Distinguishing a key from both multivariate leakage samples and multiple models can be done in one step as shown in this paper. A compact expression of the distinguisher is provided, using matrix operations. The strategy is applied to real-world traces in profiled and non-profiled scenarios. The resulting attack is more efficient than the traditional approach "dimensionality reduction then stochastic (linear regression) attack)". The new multivariate distinguisher outperforms the other state-of-the-art attacks. The presented methodology allows for leakage agnostic attacks on vectorial leakage measurements and complex models. In addition, the matrix-based expression of the distinguisher benefits from matrix-oriented software that implements computational optimizations for large dimensions.

A companion future work would consist in determining the optimal model dimensionality and basis from any acquisition campaign. Another perspective is to adapt the methodology to masked implementations, as already done for monovariate leakage in [**?**], yet for this case the distinguishers will certainly not exhibit simple closed-form expressions. However, we believe that the approach could be fruitful in practice backed with suitable optimization software.

Acknowledgements The authors wish to thank Liran Lerman for interesting discussions about stochastic attacks.

Part of this work has been funded by the ANR CHIST-ERA project SECODE (*Secure Codes to thwart Cyber-physical Attacks*).