Side-Channel Analysis of Arithmetic Encodings for Post-Quantum Cryptography: Cautionary Notes with Application to Kyber

Duyen Pay[1] and François-Xavier Standaert[1]

[1] UCLouvain, ICTEAM, Crypto Group, Louvain-la-Neuve, Belgium

Abstract. The unprotected implementations of `Kyber` and `Dilithium` have recently been shown to offer a variety of side-channel attack paths. These attacks have in turn triggered the investigation of secure and efficient masked implementations. In this paper, we observe that the design and evaluation of such masked implementations come with new challenges, due to the manipulation of small and non-uniform secrets that is common in post-quantum encryption algorithms, which may hinder their good understanding. On the one hand, we show that using the Signal-to-Noise Ratio (SNR) per share to select Points-of-Interest (POIs) in leakage traces, as it is common in symmetric cryptography, can lead to confusing outcomes where leakage samples that correspond to the manipulation of another share than the targeted one are detected. On the other hand, we show that the arithmetic encoding of small and non-uniform secrets leads to representation-dependencies so that summing or subtracting shares leads to different amounts of information leakage. We apply these observations to `Kyber` and show that they essentially vanish when increasing the number of shares. Incidentally, we also discuss the attack strategies to recover small and non-uniform secrets with side-channel attacks efficiently. We hope these observations can help implementers and evaluators to better interpret their security claims.

1 Introduction

The implementation of post-quantum cryptographic algorithms with security guarantees against side-channel attacks is known to be challenging. Focusing on recently selected standards, powerful attacks against `Crystals-Kyber` [56] have been put forward in an already long sequence of works, for example including [53,47,60,62,51,59], which then motivated the investigations of protected implementations [9,2,10,14]; similar efforts exist for `Crystals-Dilithium` [38], both on the attack side [52,37,43,7] and on the protection side [46,1,15].

The main countermeasure used in these protected implementations is masking [27,13]. It allows building on a broad literature primarily developed for symmetric cryptography, which clarified the theoretical guarantees that masking offers [33,50,20,21] and the various challenges for these guarantees to be observed in practice [42,48,16,3,23]. Yet, and despite conceptual similarity, masking post-quantum cryptographic algorithms also comes with specificities. One of them, already covered in the aforementioned references, is the requirement to mix Boolean encodings and prime encodings. Negatively, this implies expensive conversion algorithms, a topic that was itself the focus of a long sequence of

works (e.g., see [17,26] for early results and [18,8,6,19] for more recent ones). Positively, arithmetic masking in prime fields has recently been shown to offer a better tolerance to low-noise leakages, due to its reduced "algebraic compatibility" with the typical (linear) leakage models observed in practice [45].

In this paper, we are concerned with another difference, namely the fact that post-quantum algorithms require the manipulation of small and non-uniform secrets. Despite looking innocuous at first sight (e.g., it does not affect the security order of the countermeasure), this implies that a number of convenient intuitions that hold when masking symmetric algorithms like the `AES` do not apply to post-quantum algorithms, which we summarize in two cautionary notes.

The first observation relates to the selection of Points-of-Interest (POIs) in the leakage traces, which is an important step toward mounting powerful profiled attacks [25,22]. The Signal-to-Noise ratio is among the most popular tools for this purpose [39], since it allows spotting all the (bijectively connected) POIs that can be characterized with a single template. However, we show that the natural approach of estimating the SNR per share of a masked implementation can lead to confusion in the case of a small non-uniform secret. Namely, if such a small secret is shared in two pieces, it inevitably implies that the leakage of the shares is (mathematically) correlated. As a result, the SNR estimated for the first share will also lead to detect samples that depend on the second share, which may degrade the quality of the templates built for each share.

The second observation relates to the increased representation-dependency of post-quantum arithmetic encodings compared to Boolean masking. In the case of Boolean masking which is most frequently used in symmetric cryptography, there is a single way to write the additive encoding. But for arithmetic encodings, one can choose to sum or subtract shares, which creates a representation-dependency of the leakage informativeness computed with the mutual information [58]. This representation-dependency is then amplified by the small size of the secret, which implies that only selected distributions can be observed by the adversary.

As part of our investigations we also discuss the (e.g., maximum likelihood and maximum a posteriori) attack strategies that can be used to efficiently recover small and non-uniform secrets with profiled side-channel attacks.

We illustrate these notes using both simulated leakages and actual measurements, show that they hold for Simple Power Analysis (SPA) and Differential Power Analysis (DPA), and discuss their application to `Kyber`.[1] We also relativise their impact by showing that they essentially vanish when the number of shares used in the encodings increases. Overall, we nevertheless believe these observations are important to highlight the specificities of post-quantum arithmetic encodings. They have a direct impact on first-order masking that remains popular due to the reduced overheads it leads to. For example, [31,34] are specialized to first-order and [9,5,24] are only evaluated for first-order. They also convey the message that post-quantum cryptography comes with side-channel evaluation challenges that differ from the ones observed in symmetric cryptography.

[1] In the SPA case, the leakage informativeness depends on the few inputs that the adversary can observe, so the representation-dependency is less unique [41].

2 Background

2.1 Kyber algebraic structure

`Kyber` is an IND-CCA2-secure key-encapsulation mechanism that allows the establishing of a shared secret key between two communicating parties. Its security is based on the hardness of solving the Learning-With-Errors problem in Module lattices (MLWE problem). In short, the MLWE problem is to distinguish between the uniform samples $(\mathbf{a}_i, b_i)$ from $R_q^k \times R_q$ and samples $(\mathbf{a}_i, b_i)$, where $\mathbf{a}_i$ is uniformly distributed from R_q^k and $b_i = \mathbf{a}_i^T \mathbf{s} + e_i$, and where the secret $\mathbf{s}$ and the noise e_i follow special distributions. The polynomial ring is defined to be $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ consisting of polynomials of the form

$$f = f_0 + f_1 X + \cdots + f_n X^n, \tag{1}$$

where $f_i \in \mathbb{Z}_q$ for all j. The noise polynomials in `Kyber` are sampled from the Centered Binomial Distribution (CBD). The CBD is parameterized by an integer $\eta \in \{2, 3\}$. To sample a polynomial e, from CBD (B_η) requires to sample each of its coefficients e_j independently from B_η, with B_η defined as

$$\begin{aligned} &(a_1, a_2, \ldots, a_\eta, b_1, b_2, \ldots, b_\eta) \xleftarrow{\$} \{0,1\}^{2\eta}, \\ &e_j = \sum_{i=1}^{\eta} (a_i - b_i). \end{aligned} \tag{2}$$

`Kyber` comes with different sets of parameters, which depend on the target security level (see Table 1 in [56]). For the sake of simplicity, we fixed the parameters in this note to `Kyber`512, where $n = 256, q = 3329, \eta = 2$, and $k = 1$.

Notations For the rest of the note we use, calligraphic letters (e.g., $\mathcal{X}$) for sets, capital letters (e.g., X) for random variables, small letters (e.g., x) for realizations of the random variable. Bold capital and bold small letters (e.g., $\mathbf{X}$, $\mathbf{x}$) further denote random vectors and their realizations, respectively

We use the notation $X \xleftarrow{\$} \mathcal{X}$ for X being sampled uniformly at random from the set $\mathcal{X}$ and $X \leftarrow B_\eta$ if it follows the CBD distribution with parameter η. Due to our choice of parameters, the set of secrets is fixed to

$$\mathcal{S} = \{0, 1, -1, 2, -2\}, \tag{3}$$

of which the corresponding prior distribution is given by

$$\mathsf{p}_{\mathcal{S}} = [0.325, 0.25, 0.25, 0.0625, 0.0625]. \tag{4}$$

2.2 Boolean and arithmetic masking

In a d^{th}-order masked implementation, each intermediate variable is split into $d + 1$ shares, leading to so-called encodings that we define next [50].

Definition 1 (d-share encoding). *Let $\mathcal{X}$ be a set in a group $(\mathcal{G}, *)$ where $*$ is some group operation, and let d be a positive integer. The d-share encoding of $X \in \mathcal{X}$ is a mapping*

$$\begin{aligned}\mathsf{Enc}_d^{\mathsf{g},*} : \mathcal{X} &\to \mathcal{G}^d : \\ X &\mapsto (X_1, \ldots, X_d)\end{aligned}$$

such that $(X_i)_{i=1}^{d-1} \xleftarrow{\$} \mathcal{G}$, $X = \mathsf{g}(X_1, X_2, \ldots, X_d)$ and g acts on X_i through $*$.

In this definition, the g function dictates how shares are combined at the beginning (resp., unmasked at the end) of a sensitive operation. For example, in a symmetric cipher like the AES Rijndael, where the underlying group is $\mathbb{Z}_{2^8}$, the Boolean additive 2-share encoding is defined as

$$\mathsf{Enc}_d^{\mathsf{g},\oplus}(X) = (X_1, X_2),$$

where $X_1 \xleftarrow{\$} \mathbb{Z}_{2^8}$ and $X = \mathsf{g}(X_1, X_2) = X_1 \oplus X_2$ (i.e., $X_2 = X \oplus X_1$).

The arithmetic 2-share encoding in an additive group $(\mathcal{G}, +)$ can be defined similarly as $\mathsf{Enc}_d^{g_i,+}(X) = (X_1, X_2)$, with as only difference that it can be expressed in two different ways, with

$$X = \mathsf{g}_1(X_1, X_2) = X_1 + X_2 \qquad \text{or} \qquad X = \mathsf{g}_2(X_1, X_2) = X_2 - X_1.$$

For the rest of the note, we focus on such additive masking, omit the group operation on the superscript, and use the simplified notations

$$\mathsf{Enc}_d^{\text{sum}} \text{ for } g(X_1, X_2, \ldots, X_d) = \sum_{i=1}^{d} X_i, \qquad \text{i.e., } X_d^{\text{sum}} = X - \sum_{i=1}^{d-1} X_i,$$

$$\mathsf{Enc}_d^{\text{diff}} \text{ for } g(X_1, X_2, \ldots, X_d) = X_d - \sum_{i=1}^{d-1} X_i, \qquad \text{i.e., } X_d^{\text{diff}} = X + \sum_{i=1}^{d-1} X_i.$$

Lastly, the noise polynomials in `Kyber` consist of $n = 256$ coefficients that are independently sampled from B_η and are masked independently. Without losing generality, we consider the masking of one coefficient of such polynomials.

2.3 POI detection with the SNR

Side-channel attacks exploit leakage traces $\mathbf{L} = \{\mathbf{l}_i\}_{i=1}^q$ that correspond to data $X = \{x_i\}_{i=1}^q$. Each trace may contain hundreds of thousands of samples, i.e., $\mathbf{l} = \{l_t\}_{t=0}^N$ with large N values, where only a few of them are actually informative for the attack in the sense that they directly depend on the target variable X. As a result, selecting such POIs usually comes as a preliminary step in side-channel attacks. A popular statistical tool for this purpose is the side-channel SNR [40].

Assuming standard modeling of the leakage traces such that every sample is the sum of a deterministic $\delta_t(x)$ part and a noise part n_t [55], namely

$$l_t^x = \delta_t(x) + n_t, \tag{5}$$

the side-channel SNR can be directly estimated as

$$\mathrm{SNR}_t = \frac{\hat{\mathrm{var}}_x[\hat{\mathrm{E}}[\mathbf{l}_t^x]]}{\hat{\mathrm{E}}_x[\hat{\mathrm{var}}[\mathbf{l}_t^x]]}. \tag{6}$$

Next, the adversary can work on a subtrace made of samples with sufficient SNR rather than working on the full trace, leading to better efficiency.

2.4 Profiled attacks

From profiling samples $(\mathbf{L}, X)$, a profiled distinguisher estimates a model of the conditional Probability Mass Function (PMF) $\hat{\mathsf{p}}(x|\mathbf{l})$, from which a maximum a posteriori attack can be launched, with the most likely secret chosen as

$$x^* = \arg\max_{x \in \mathcal{X}} \hat{\mathsf{p}}(x|\mathbf{l}).$$

In the following, we estimate such a model using Fisher's Linear Discriminant Analysis (LDA), which can be viewed as an improvement of Chari et all.'s seminal template attacks [13,57]. We then exploit the information extracted from individual shares using a Soft-Analytical Side-channel Attack (SASCA), which allows us to efficiently recover information on the target secret [61,29]. We will denote such a combination as LDAxSASCA for the rest of the note.

Linear Discriminant Analysis The task of modeling $\hat{\mathsf{p}}(x|\mathbf{l})$ is well-known to suffer the *curse of dimensionality* [36]. So to further reduce the number of dimensions after POI selection, LDA projects the original data to a subspace of lower dimension which maximizes the inter-class variance and minimizes the intra-class variance. LDA is known to be optimal in terms of minimizing the Bayes error for binary classification under normality and homoscedasticity assumptions [30]. The LDA directions $\mathbf{w}$ are the solution of the maximization problem of the objective $\frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}}$, where $\mathbf{S}_B$ and $\mathbf{S}_W$ are the inter-class scatter and intra-class scatter matrices, respectively. They can be estimated as

$$\hat{\mathbf{S}}_B = \sum_{c=1}^{n_c} N_c(\hat{\boldsymbol{\mu}}_c - \hat{\boldsymbol{\mu}})(\hat{\boldsymbol{\mu}}_c - \hat{\boldsymbol{\mu}})^T,$$

$$\hat{\mathbf{S}}_W = \sum_{c=1}^{n_c}\sum_{i=1}^{N}(\mathbf{l}_i^c - \hat{\boldsymbol{\mu}}_c)(\mathbf{l}_i^c - \hat{\boldsymbol{\mu}}_c)^T,$$

where $\hat{\boldsymbol{\mu}}_c = \frac{1}{N_c}\sum_{i=1}^{N_c}\mathbf{l}_i^c$ is the empirical mean of the traces corresponding to x in class c, and, $\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{c=1}^{n_c}\hat{\boldsymbol{\mu}}_c N_c$ is the total mean of all classes.

Finding $\mathbf{w}$ is usually reduced to the problem of finding the eigenvectors of the matrix $\hat{\mathbf{S}}_W^{-1}\hat{\mathbf{S}}_B$ and several eigenvectors that correspond to the highest eigenvalues are composed into a projection matrix $\mathbf{W}$. The original data is then transformed to lower dimension space (i.e., $\mathbf{l}_{lda} = \mathbf{W}\mathbf{l}$). The leakage traces after LDA projection are finally used to model the leakage Probability Density Function (PDF) as multivariate Gaussian templates, leading to the conditional PDF

$$\hat{\mathsf{f}}(\mathbf{l}|x) = \frac{1}{\sqrt{(2\pi)^k \det \hat{\mathbf{\Sigma}}}} \exp\left(-\frac{1}{2}(\mathbf{l}-\hat{\boldsymbol{\nu}}_c)^T \hat{\mathbf{\Sigma}}^{-1}(\mathbf{l}-\hat{\boldsymbol{\nu}}_c)\right), \tag{7}$$

where $\hat{\boldsymbol{\nu}}_c$ is the empirical mean of the projected traces corresponding to x in class c and the covariance matrix $\hat{\mathbf{\Sigma}}$ (also estimated from projected traces) is pooled from the covariance matrices of all classes $\hat{\mathbf{\Sigma}}_c$ as

$$\hat{\mathbf{\Sigma}} = \frac{1}{N}\sum_{c=1}^{n_c} N_c \hat{\mathbf{\Sigma}}_c.$$

Soft-Analytical Side-channel Attacks. SASCAs were introduced in [61] and have recently gained popularity in analyzing masked implementations of symmetric ciphers [12], to perform single-trace attacks against Keccak implementations [35] or to target the Number Theoretic Transform (NTT) used in lattice-based cryptosystems [49]. In general, a SASCA combines a description of a leaking implementation thanks to a factor graph with a decoding, for example using the Belief Propagation (BP) algorithm. While initially introduced as a way to exploit the deeper leakage samples of block cipher implementations (i.e., where the intermediate computations depend on too many key bits to be targeted via a divide-and-conquer approach), it also turns out to be very handy to analyze the leakage of masked implementations at limited computational cost [28].

Precisely, in the context of this paper, we want to estimate the leakage PDF $\hat{\mathsf{f}}(\mathbf{l}|s)$ of a d-share encoding $\mathsf{Enc}_d(S) = (X_1, \ldots, X_d)$, which corresponds to the following (Gaussian) mixture distribution

$$\hat{\mathsf{f}}(\mathbf{l}|s) = \sum_{x_1,\ldots,x_{d-1}\in\mathbb{Z}_Q} \hat{\mathsf{f}}(\mathbf{l}|x_1)\cdot\hat{\mathsf{f}}(\mathbf{l}|x_2)\cdot\ldots\cdot\hat{\mathsf{f}}(\mathbf{l}|x_d)\cdot\mathsf{p}(x_1)\cdot\mathsf{p}(x_2)\cdot\ldots\cdot\mathsf{p}(x_{d-1}),$$

without exhaustively summing over all the shares. The latter can be done efficiently by using Proposition 1 in [44] and computing

$$\hat{\mathsf{f}}(\mathbf{l}|s) = \hat{\mathsf{f}}(\mathbf{l}|x_1) \circ \hat{\mathsf{f}}(\mathbf{l}|x_2) \circ \ldots \hat{\mathsf{f}}(\mathbf{l}|x_d), \tag{8}$$

where $\circ$ denotes the convolution operation. Performing these convolutions can be seen as a SASCA on a tree-like graph, and the BP algorithm is known to provide an exact solution in this case. We use the optimized library `SCALib` for this purpose, adjusted to fit with the special distribution of the secret.[2] Based on the obtained PDF, we finally compute the PMF $\hat{\mathsf{p}}(s|\mathbf{l})$ thanks to Bayes.

[2] https://scalib.readthedocs.io/en/stable/

2.5 Evaluation metrics

We will use information theoretic metrics (namely, the mutual & perceived information) in order to assess security against DPA (since they provide a tight quantification of such attacks' data complexities) and security metrics (namely the guessing entropy) in order to assess security against SPA.

Mutual Information and Perceived Information Information theoretic metrics are common tools to evaluate the worst-case security against DPA [58]. The most popular such metric is the Mutual Information (MI), defined as

$$\mathrm{MI}(S;\mathbf{L}) = \mathrm{H}(S) + \sum_{s\in\mathcal{S}} \mathsf{p}(s) \int_{\mathbf{l}\in\mathcal{L}^d} \mathsf{f}(\mathbf{l}|s) \cdot \log_2 \mathsf{p}(s|\mathbf{l}).$$

The MI value can be used to bound the minimum number of measurements N_a that an adversary must obtain in order to recover X via DPA [4]. In practice, the MI is usually estimated by sampling to avoid the intractable cost of the integration when the dimension of $\mathbf{L}$ grows as

$$\widehat{\mathrm{MI}}(S;\mathbf{L}) = \mathrm{H}(S) + \sum_{s\in\mathcal{S}} \mathsf{p}(s) \sum_{i=1}^{N_x} \frac{1}{N_x} \cdot \log_2 \mathsf{p}(s|\mathbf{l}^s(i)), \tag{9}$$

where $\mathbf{l}^s(i)$ and N_x are ith leakage trace generated with the secret $S = s$ and the total number of traces corresponds to this secret, respectively. This estimation is known to converge to the correct MI value as N_s grows [11].

The MI can however only be computed in case the adversary has access to the true leakage distribution. In concrete settings, this leakage distribution is usually unknown, leading to the need to estimate either the model $\hat{\mathsf{p}}(.|.)$ or the metric. The Perceived Information (PI) captures the first approach and allows evaluating the amount of information that can be extracted from an estimated model, possibly biased by estimation or assumption errors [54]. It can be computed by sampling as

$$\widehat{\mathrm{PI}}(S;\mathbf{L}) = \mathrm{H}(S) + \sum_{s\in\mathcal{S}} \mathsf{p}(s) \sum_{i=1}^{N_s} \frac{1}{N_s} \cdot \log_2 \hat{\mathsf{p}}(s|\mathbf{l}^s(i)). \tag{10}$$

The sampling process to estimate $\widehat{\mathrm{PI}}(S;\mathbf{L})$ needs to be carried out on a separate set than used to estimate $\hat{\mathsf{p}}(.|.)$, to ensure it is unbiased. It is shown in [11] that the PI upper bounds the MI, and the equality holds if the model is perfect.

Guessing Entropy While information theoretic metrics offer an efficient way to predict the data complexity of side-channel attacks, they ignore their time complexity and therefore, are usually paired with security metrics that give a more direct view of an implementation's concrete security level. A popular option

for this purpose is to estimate the Guessing Entropy (GE), which measures the average amount of keys an adversary must enumerate to perform a side-channel key recovery. Typically, after performing an attack, the adversary has a guess vector $\mathbf{g} = [g_1, \ldots, g_{|\mathcal{S}|}]$, where target secret candidates g_i's are sorted by decreasing likelihood. Then GE of such the attack is

$$\widehat{\mathbf{GE}} = \underset{\text{attacks}}{\hat{\mathrm{E}}}[i | g_i \text{ is the correct key}]. \tag{11}$$

In the context of SPA, one can directly estimate such a metric (without information theoretic ones), since the attack complexity is fixed by the context.

3 Leakage simulation and real measurement setup

We illustrate our cautionary notes with both simulations and actual measurements. The first ones aim to enable easier interpretation, since they correspond to a more controlled environment where the leakage function is known. The second ones aim to confirm the practical relevance of our observations.

3.1 Simulated leakages

The secrets are generated from the set of Equation 3 and follow the distribution of Equation 4. The leakages for d-share encodings are generated as follows:

1. First, generate $\mathsf{Enc}_d(S)$ where the first $d-1$ shares are drawn uniformly at random from $\mathbb{Z}_q$, and the last share X_d is computed to ensure correctness.
2. Next, the leakage of each share L_i is computed with the Hamming Weight (HW) model and additive Gaussian noise: $L_i = \mathrm{HW}(X_i) + B_i$, where $B_i \leftarrow \mathcal{N}(0, \sigma^2)$. As a result, the share's leakage PDF has the form

$$\mathsf{f}(l_i | x_i) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{l_i - \mathrm{HW}(x_i)}{\sigma}\right)^2}. \tag{12}$$

3. Finally, the leakage vector corresponding to the processing of S, $\mathbf{L}$ is the concatenation of the shares' leakages, i.e., $\mathbf{L} = [L_1, \ldots, L_d]$.

Integrating such leakage into Equation 6, the SNR for each share can be computed as a function of the noise variance σ^2, i.e., $\mathrm{SNR} = \frac{2{,}67}{\sigma^2} \approx \frac{11/4}{\sigma^2}$, where 11 is the number of bits used to represent the moduli and $11/4$ is the variance of the Hamming weights corresponding to random 11-bit values (and the $\approx$ sign reflects the fact that shares are uniform in $\mathbb{Z}_q$ for q prime rather than $\mathbb{Z}_{2^{11}}$).

3.2 Measurement setup

We measured an implementation similar to the public one from [10], running on an ARM Cortex-M4 STM32F415. This implementation uses $\mathsf{Enc}_2^{\mathrm{sum}}$ and we

tweaked it in order to produce traces for $\mathsf{Enc}_2^{\mathsf{diff}}$ as well. The noise coefficients are generated following Equation 2 with $\{a_i, b_i\}$ produced by the AES128.

The MCU was mounted on the CW308 UFO board, with an external 8 MHz crystal oscillator to fix the system clock. The leakages were measured with the CT1 current probe and the signal was sampled by a PicoScope 5244D at 500 MSamples/s with 12-bit resolution with no signal pre-processing nor averaging. We collected two million traces for each target and focused our analysis on the encoding loaded before the execution of the NTT in `Kyber`'s re-encryption.

3.3 Evaluation Methodology

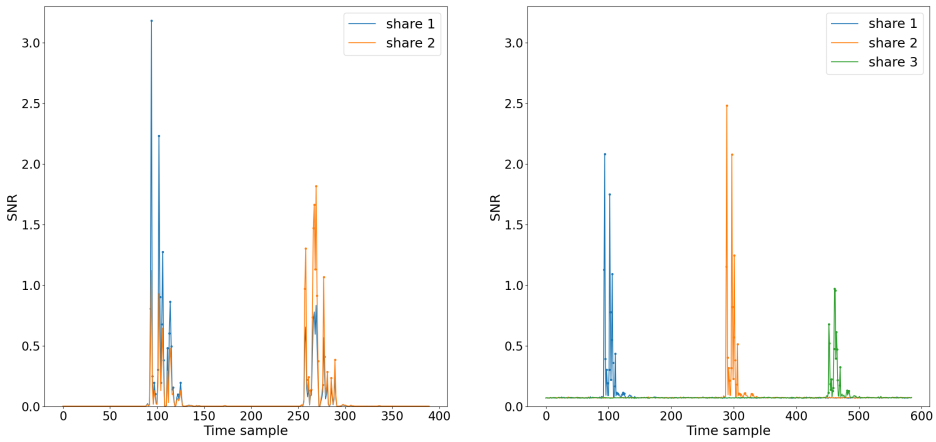
Based on the previous background, our evaluations (both with simulated leakages and actual measurements) are based on the following steps:

1. Divide the dataset into a profiling dataset and an attack dataset.
2. **On measurements** Use the SNR in order to select POIs (i.e., pick the points with highest SNR for evaluation). Then estimate (for the POIs) the leakage PDF given the shares $\hat{\mathsf{f}}(\mathbf{l}|x_i)$ for each share using LDA.
   In simulations Compute $\mathsf{f}(\mathbf{l}|x_i)$ directly as given by Equation 12.
3. Estimate the secret PDF $\hat{\mathsf{f}}(\mathbf{l}|s)$ from the shares PDF using SASCA.
4. Compute the MI/PI/GE using the secret PDF on the attack dataset.

4 Cautionary note on POI detection

As mentioned in the introduction, the SNR is a popular tool for selecting POIs in leakage traces. In this section, we show that its application to the shares of an arithmetic encoding can create confusion when the encoded value is small. For this purpose, we first report the SNR computed from the measurements of Section 3.2 for the two shares of an arithmetic encoding in Figure 1a, where the first share is manipulated around time sample 100 and the second share is manipulated around time sample 275. One can see that the POIs suggested by the SNR are not perfectly isolated: the SNR computed for the first (resp., second) share can pop up at the position of the second (resp., first) share. Such *ghost peaks* give the incorrect impression that there is useful information about a share beyond the points in time where it is manipulated. As a result, blindly applying this POI selection can disturb the performances of a profiled attack.

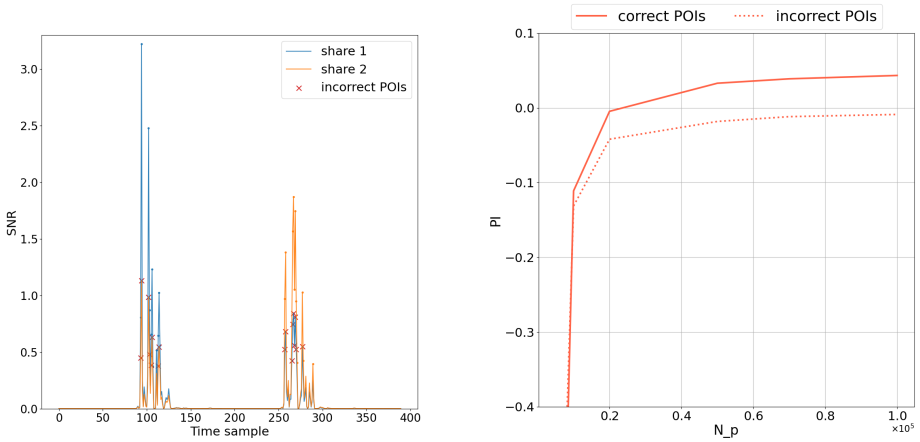
One important remark in this respect is that since conditioned on the secret, the shares are not independent, it also implies that wrongly selecting POIs for the first share (resp., second share) in the time samples corresponding to the second share (resp., first share) does not only increase the profiling data (and time) complexity, as would be expected if they were independent [36]. We report the PI estimated from the LDAxSASCA profiled with a blind application of the SNR-based POI detection vs. an informed one where we only keep the samples that match the actual manipulation of the shares in Figure 2b. It shows that the ghost peaks perturb the model persistently, as reflected by a negative PI



(a) 2 shares. (b) 3 shares.

Fig. 1: Shares' SNR of an arithmetic encoding.

when incorrect POIs are used.[3] This is because errors are not averaged by using more profiling data in this case, which is in contrast with the selection of non-informative points that do not correspond to any of the shares.



(a) SNR per share. (b) PI of $\mathsf{Enc}_2^{\mathrm{sum}}$ from LDA-SASCA.

Fig. 2: Impact of wrong POI selection on LDAxSASCA: 2-share case.

These ghost peaks exist due to the fact that the secret is not uniform and has small support. That is, for each value of the first share (e.g., $X_1 = 0$), the second share only takes some values corresponding to all possible values of the secret

[3] Here, the model is computed with a non-uniform prior. A similar observation holds with uniform prior. We discuss the impact of these priors in Section 6.4.

S (i.e., $X_2 \in \{0, 1, -1, 2, -2\}$), and also follows the secret's distribution. As a result, instead of being uniformly distributed over $\mathbb{Z}_q \times \mathbb{Z}_q$ as classically observed for encodings used in symmetric cryptography, the pairs of shares (X_1, X_2) lie in a specific/restricted set and therefore carry information about each other.

More precisely, in the 2-share case, the conditional entropy of one share given the other, $\mathrm{H}(X_1|X_2)$, exactly equals the entropy of the secret $\mathrm{H}(S)$. This fact holds for all distributions of S and is an unchanged relationship between the two shares' values. Combined with the fact that $\mathrm{H}(X_1|X_2)$ spreads on $\mathbb{Z}_q$ while $H(S)$ spreads only on $\mathcal{S}$ where $|\mathcal{S}| \ll q$, each share mathematically correlates with the other. This explains our observations of Figure 1a and shows that they are not specific to one detection tool: any tool relying on the estimation of statistical moment (e.g., Pearson's correlation) would suffer from the same problem.

This phenomenon however disappears when the number of shares is more than two, as illustrated in Figure 1b, which we will explain based on an example. Say we consider the pair (X_1, X_3). Since the secret's distribution is now *absorbed* by the uniform distribution of X_2, the pair (X_1, X_3) is uniform over $\mathbb{Z}_q \times \mathbb{Z}_q$ and the correlation between them vanishes. As a result, the SNR per share rightfully spots leakage samples that correspond to the shares' manipulation only.

Based on this first cautionary note, and when considering a 2-share case, our following experiments will therefore all be based on an informed POI selection, where we manually isolate POIs that correspond to the target share.

5 Interlude on attack strategies

A natural next step after identifying POIs is to perform a profiled attack. In the case of `Kyber`, we can for example target the encoding manipulated just before the NTT computation in the re-encryption step with an SPA (since it is an ephemeral secret), and therefore estimate the resulting guessing entropy.

Yet, since the secret we target is then non-uniform, the maximum likelihood and maximum a posteriori attack strategies are not equivalent anymore. This is again in contrast with the situation in symmetric cryptography, where the target secrets always have a uniform prior. We next detail these different strategies.

The Maximum Likelihood (ML) approach selects the secret as

$$\tilde{s} = \underset{s}{\operatorname{argmax}}\ \mathsf{f}(\mathbf{l}|s),$$

while the Maximum A Posteriori (MAP) approach selects it as:

$$\begin{aligned}\tilde{s} &= \underset{s}{\operatorname{argmax}}\ \mathsf{p}(s|\mathbf{l}),\\ &= \underset{s}{\operatorname{argmax}}\ \frac{\mathsf{f}(\mathbf{l}|s) \cdot \mathsf{p}(s)}{\sum_{s^*} \mathsf{f}(\mathbf{l}|s^*) \cdot \mathsf{p}(s^*)},\\ &= \underset{s}{\operatorname{argmax}}\ \mathsf{f}(\mathbf{l}|s) \cdot \mathsf{p}(s).\end{aligned}$$

When extended to multi-trace leakage vectors $\mathbf{l}$ it directly gives

$$\tilde{s} = \underset{s}{\operatorname{argmax}} \prod_{i=1}^{q} \mathsf{f}(\mathbf{l}(i)|s),$$

in the ML case, while the generalization of the MAP given in [58,32] is

$$\tilde{s} = \underset{s}{\operatorname{argmax}}\ \mathsf{p}(s) \cdot \prod_{i=1}^{q} \mathsf{f}(\mathbf{l}(i)|s).$$

Those strategies can be equivalently written in logarithmic form as:

$$\tilde{s} = \underset{s}{\operatorname{argmax}} \sum_{i=1}^{q} \log \mathsf{f}(\mathbf{l}(i)|s) \qquad \text{(ML)} \qquad (13)$$

$$\tilde{s} = \underset{s}{\operatorname{argmax}} \left[\sum_{i=1}^{q} \log \mathsf{f}(\mathbf{l}(i)|s) + \log \mathsf{p}(s) \right] \qquad \text{(MAP)} \qquad (14)$$

As a result, the two approaches are equivalent when there is a uniform prior on s and differ otherwise. Interestingly, the latter happens in our `Kyber` case study. Furthermore, the arithmetic encoding we target enables SPA with repetition. That is, the adversary can repeatedly observe the leakage of this encoding for the same (stable) secret s. We analyzed the efficiency of these different strategies with the simulated leakages of Section 3.1. The guessing entropy of attacks exploiting 1, 10, 50, and 100 repetitions in function of the shares' SNR is given in Figures 3 and 4 for the sum and diff. encodings (and 500, 1000, 2000, 10000, 20000 repetitions in Figures 5 and 6), leading to the following observations.

Firstly, when the prior information of the secret is available, MAP consistently performs better for every noise level. More precisely, both the ML and MAP approaches allow accumulating information from multiple traces. Yet, whenever the distinguisher encounters an non-informative leakage (e.g., when $\mathsf{f}(\mathbf{l}(i)|\cdot)$ is equal for correct and incorrect secrets), ML guesses the secret at random while MAP bases its guess on the prior distribution. Hence, for low number of repetitions, MAP leads to better results, as shown in Figures 3 and 4.

Secondly, MAP is essentially ML one-time-weighted by the prior. Thus, both converge towards the same value and correctly guess the secret when the model is sound with enough data, as shown for high SNRs in Figures 5 and 6.

Additionally, we observe that the guessing entropy of Figure 4 sometimes saturates, which is due to the distributions of some secret values that remain hard to distinguish and will be discussed in the next section.

6 Cautionary note on representation-dependency

In this section, we investigate the dependency of arithmetic encodings protecting small and non-uniform secrets to their representation (i.e., whether they

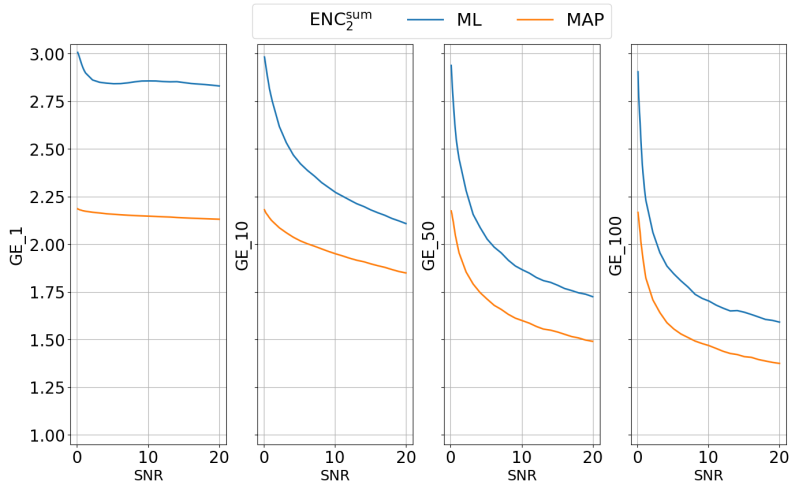


Fig. 3: GE of simulated attacks against the sum encoding with different strategies.

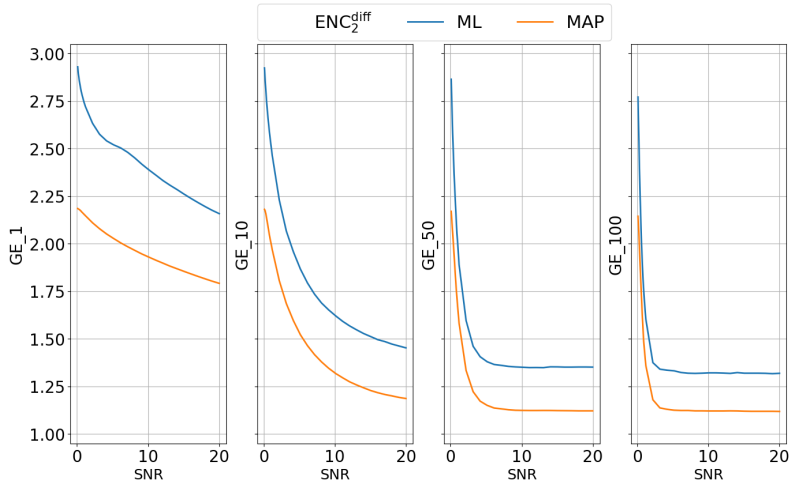


Fig. 4: GE of simulated attacks against the diff. encoding with different strategies.

sum or subtract shares). As already mentioned, a natural application of such encodings is before the NTT in `Kyber`'s re-encryption step, where a SPA with repetition is possible. More precisely, the previous section already hinted towards this representation-dependency and we now aim to discuss it more in depth. For this purpose, we will start with an intuitive discussion based on PDF plots in Section 6.1, follow with a simulated analysis that puts forward this dependency and how it vanishes with a larger number of shares in Section 6.2, confirm these findings with experiments in Section 6.3 and discuss their extension to DPA for completeness (since not motivated by a concrete case study) in Section 6.4.

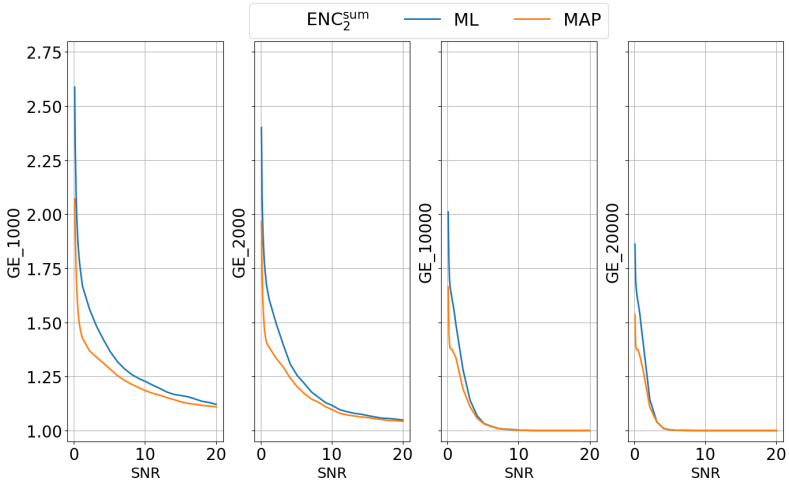


Fig. 5: GE of simulated attacks against the sum encoding with more averaging.

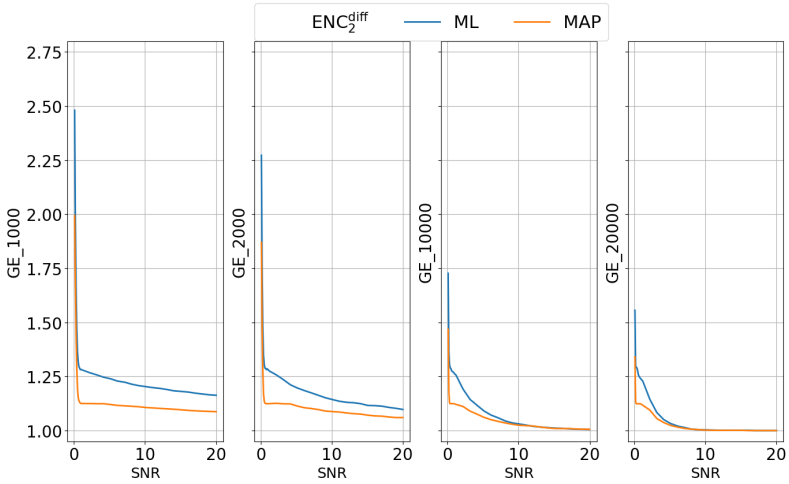


Fig. 6: GE of simulated attacks against the diff encoding with more averaging.

6.1 PDF plots for the two encodings

The plots corresponding to the mixture PDF of the two (sum and diff.) encodings in the noisy Hamming weight leakage model are given in Figures 7 and 8.

They lead to two main observations. First, we see that the diff. encoding seems more informative than the sum one. This is because the distributions in Figure 8 are (visually) more separated than the ones in Figure 7. Second, we also see that some distributions are very hard to distinguish in the diff. encoding case (e.g., those of s =1,2 or s =-1,-2). This suggests that the diff. encoding will lead to easier attacks in the DPA setting and (on average) in the SPA setting, but

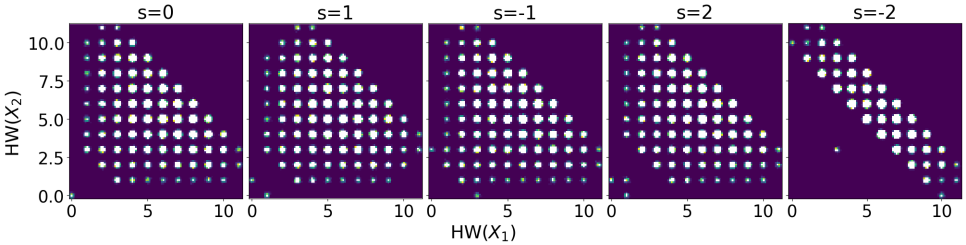


Fig. 7: Bivariate PDF $\mathsf{f}(\mathbf{l}|s)$ for the sum encoding.

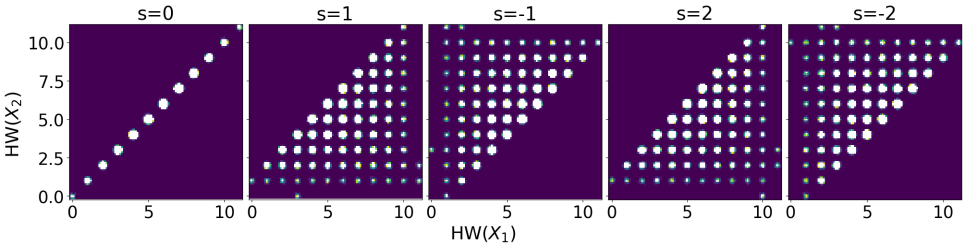


Fig. 8: Bivariate PDF $\mathsf{f}(\mathbf{l}|s)$ for the diff. encoding.

some of these keys may remain hard to distinguish in the SPA setting (which is what we observed with the saturation effect in the previous section).

6.2 Simulated leakages

Moving to a more quantitative analysis, Figure 9 shows the evolution of the guessing entropy for the two encodings, in function of the shares' SNR and the number of shares, for an increasing number of repetitions (when moving from left to right). It confirms the previous intuition that the diff. encoding leads to stronger attacks than the sum one (in similar conditions). It also highlights that the gap between the informativeness of the two encodings decreases when the noise and the number of shares increases. This is presumably explained by the fact that when combining more (noisy) shares, the mixture PDFs tend to be more uniform, which therefore flattens patterns that may appear with a low number of shares. Note that the reduction of this gap is combined with the reduction of informativeness caused by lower SNR and larger number of shares. It will be easier to observed in the information theoretic plots of Section 6.4.

6.3 Actual measurements

Figure 10 provides the guessing entropy in function of the number traces used to profile the leakage model (N_p) of the two encodings, for two shares and the noise level provided by our actual measurements (again for increasing the number of repetitions in the attack). It confirms that the conclusions obtained with simulated leakages are also matched for our software implementation setting.

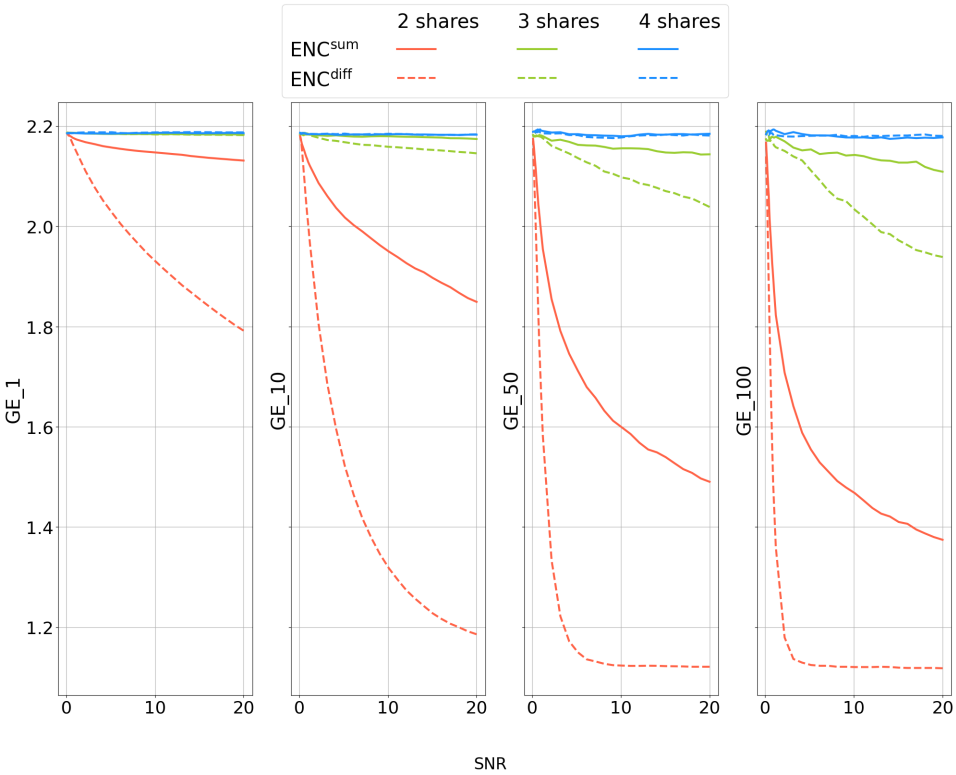


Fig. 9: GE of simulated attacks with different numbers of shares.

6.4 From SPA to DPA

Eventually, and for completeness, we provide results similar to those of the previous section but replacing the guessing entropy (i.e., a security metric that captures SPA) by the MI/PI (i.e., information theoretic metrics that efficiently capture DPA) in Figure 11. The mutual information is used for simulated leakages, the perceived information is used for actual measurements. Our conclusions are again essentially similar (exhibiting even simpler patterns). Namely, the gap between the two encodings is clear and vanishes with more shares. As in Section 4, we used a non-uniform prior to estimate the MI and PI. Results with a uniform prior lead to the same conclusions. The study of how such information theoretic metrics can be formally connected to the different attack strategies outlined in Section 5 is an interesting scope for further investigations.

7 Conclusions

This note highlights some new challenges that the design and evaluation of post-quantum cryptographic implementations against side-channel attacks may lead

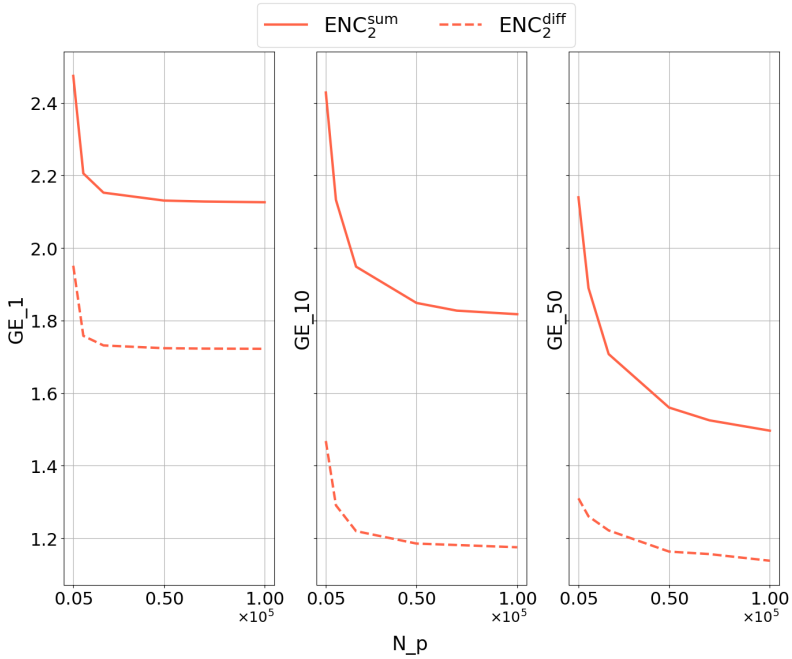
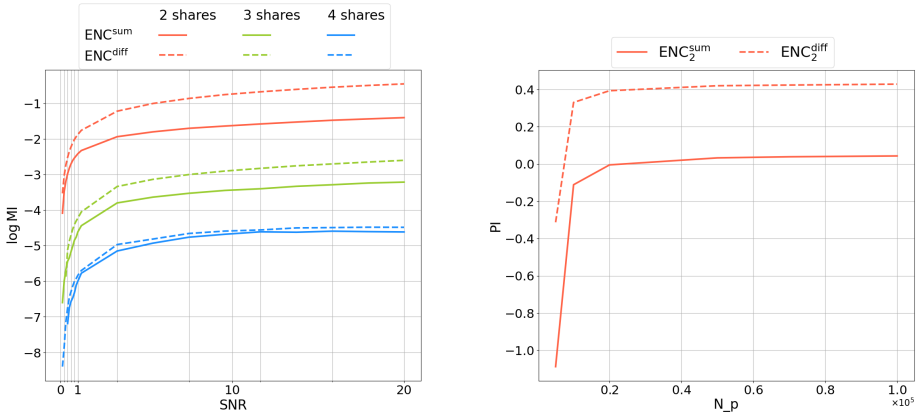


Fig. 10: GE of actual attacks with two shares.



(a) Encoding MI (simulations). (b) Encoding PI (measurements).

Fig. 11: MI and PI of simulated and actual attacks

to. For example, confusion in the detection of POIs, need of different attack strategies and representation-dependencies in arithmetic encodings. It suggests that some of the (now standard) tools and intuitions that emerged from the study of symmetric cryptographic implementations cannot be straightforwardly extended to the post-quantum context without caution. The main reason of

this gap is the manipulation of small and non-uniform secrets. While it raises no fundamental impossibilities (i.e., standard attacks can be mounted and standard countermeasures are still effective), it nevertheless requires slight adaptations for existing tools to be used in this case. We hope the notes in this paper can help implementers and evaluators to gain a good understanding of the physical security provided by masked implementations of post-quantum algorithms.

Acknowledgments. François-Xavier Standaert is a senior research associate of the Belgian Fund for Scientific Research (FNRS-F.R.S.). This work has been funded in part by the Walloon Region through the project CyberExcellence (convention number 2110186) and by the ERC Advanced Grant 101096871 (BRIDGE). Views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Union or the ERC. Neither the European Union nor the granting authority can be held responsible for them.

References

1. Melissa Azouaoui, Olivier Bronchain, Gaëtan Cassiers, Clément Hoffmann, Yulia Kuzovkova, Joost Renes, Tobias Schneider, Markus Schönauer, François-Xavier Standaert, and Christine van Vredendaal. Protecting dilithium against leakage revisited sensitivity analysis and improved implementations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(4):58–79, 2023.
2. Melissa Azouaoui, Olivier Bronchain, Clément Hoffmann, Yulia Kuzovkova, Tobias Schneider, and François-Xavier Standaert. Systematic study of decryption and reencryption leakage: The case of kyber. In *COSADE*, volume 13211 of *Lecture Notes in Computer Science*, pages 236–256. Springer, 2022.
3. Josep Balasch, Benedikt Gierlichs, Vincent Grosso, Oscar Reparaz, and François-Xavier Standaert. On the cost of lazy engineering for masked software implementations. In *CARDIS*, volume 8968 of *Lecture Notes in Computer Science*, pages 64–81. Springer, 2014.
4. Julien Béguinot, Wei Cheng, Sylvain Guilley, Yi Liu, Loïc Masure, Olivier Rioul, and François-Xavier Standaert. Removing the field size loss from duc et al.'s conjectured bound for masked encodings. In *COSADE*, volume 13979 of *Lecture Notes in Computer Science*, pages 86–104. Springer, 2023.
5. Michiel Van Beirendonck, Jan-Pieter D'Anvers, Angshuman Karmakar, Josep Balasch, and Ingrid Verbauwhede. A side-channel-resistant implementation of SABER. *ACM J. Emerg. Technol. Comput. Syst.*, 17(2):10:1–10:26, 2021.
6. Michiel Van Beirendonck, Jan-Pieter D'Anvers, and Ingrid Verbauwhede. Analysis and comparison of table-based arithmetic to boolean masking. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):275–297, 2021.
7. Alexandre Berzati, Andersson Calle Viera, Maya Chartouny, Steven Madec, Damien Vergnaud, and David Vigilant. Exploiting intermediate value leakage in dilithium: A template-based approach. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(4):188–210, 2023.
8. Luk Bettale, Jean-Sébastien Coron, and Rina Zeitoun. Improved high-order conversion from boolean to arithmetic masking. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):22–45, 2018.

9. Joppe W. Bos, Marc Gourjon, Joost Renes, Tobias Schneider, and Christine van Vredendaal. Masking kyber: First- and higher-order implementations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(4):173–214, 2021.
10. Olivier Bronchain and Gaëtan Cassiers. Bitslicing arithmetic/boolean masking conversions for fun and profit with application to lattice-based kems. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(4):553–588, 2022.
11. Olivier Bronchain, Julien M. Hendrickx, Clément Massart, Alex Olshevsky, and François-Xavier Standaert. Leakage certification revisited: Bounding model errors in side-channel security evaluations. *IACR Cryptol. ePrint Arch.*, page 132, 2019.
12. Olivier Bronchain and François-Xavier Standaert. Breaking masked implementations with many shares on 32-bit software platforms or when the security order does not matter. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):202–234, 2021.
13. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.
14. Jean-Sébastien Coron, François Gérard, Simon Montoya, and Rina Zeitoun. High-order polynomial comparison and masking lattice-based encryption. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(1):153–192, 2023.
15. Jean-Sébastien Coron, François Gérard, Matthias Trannoy, and Rina Zeitoun. Improved gadgets for the high-order masking of dilithium. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(4):110–145, 2023.
16. Jean-Sébastien Coron, Christophe Giraud, Emmanuel Prouff, Soline Renner, Matthieu Rivain, and Praveen Kumar Vadnala. Conversion of security proofs from one leakage model to another: A new issue. In *COSADE*, volume 7275 of *Lecture Notes in Computer Science*, pages 69–81. Springer, 2012.
17. Jean-Sébastien Coron and Louis Goubin. On boolean and arithmetic masking against differential power analysis. In *CHES*, volume 1965 of *Lecture Notes in Computer Science*, pages 231–237. Springer, 2000.
18. Jean-Sébastien Coron, Johann Großschädl, Mehdi Tibouchi, and Praveen Kumar Vadnala. Conversion from arithmetic to boolean masking with logarithmic complexity. In *FSE*, volume 9054 of *Lecture Notes in Computer Science*, pages 130–149. Springer, 2015.
19. Jan-Pieter D'Anvers. One-hot conversion: Towards faster table-based A2B conversion. In *EUROCRYPT (4)*, volume 14007 of *Lecture Notes in Computer Science*, pages 628–657. Springer, 2023.
20. Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. In *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 423–440. Springer, 2014.
21. Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete or how to evaluate the security of any leaking device. *IACR Cryptol. ePrint Arch.*, page 119, 2015.
22. François Durvaux and François-Xavier Standaert. From improved leakage detection to the detection of points of interests in leakage traces. In *EUROCRYPT (1)*, volume 9665 of *Lecture Notes in Computer Science*, pages 240–262. Springer, 2016.
23. Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and François-Xavier Standaert. Composable masking schemes in the presence of physical defaults & the robust probing model. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):89–120, 2018.

24. Tim Fritzmann, Michiel Van Beirendonck, Debapriya Basu Roy, Patrick Karl, Thomas Schamberger, Ingrid Verbauwhede, and Georg Sigl. Masked accelerators and instruction set extensions for post-quantum cryptography. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(1):414–460, 2022.
25. Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. stochastic methods. In *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 15–29. Springer, 2006.
26. Louis Goubin. A sound method for switching between boolean and arithmetic masking. In *CHES*, volume 2162 of *Lecture Notes in Computer Science*, pages 3–15. Springer, 2001.
27. Louis Goubin and Jacques Patarin. DES and differential power analysis (the "duplication" method). In *CHES*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer, 1999.
28. Vincent Grosso and François-Xavier Standaert. Masking proofs are tight and how to exploit it in security evaluations. In *EUROCRYPT (2)*, volume 10821 of *Lecture Notes in Computer Science*, pages 385–412. Springer, 2018.
29. Qian Guo, Vincent Grosso, François-Xavier Standaert, and Olivier Bronchain. Modeling soft analytical side-channel attacks from a coding theory viewpoint. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):209–238, 2020.
30. Onur C. Hamsici and Aleix M. Martínez. Bayes optimality in linear discriminant analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(4):647–657, 2008.
31. Daniel Heinz, Matthias J. Kannwischer, Georg Land, Thomas Pöppelmann, Peter Schwabe, and Amber Sprenkels. First-order masked kyber on ARM cortex-m4. *IACR Cryptol. ePrint Arch.*, page 58, 2022.
32. Annelie Heuser, Olivier Rioul, and Sylvain Guilley. Good is not good enough - deriving optimal distinguishers from communication theory. In 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*, pages 55–74. Springer, 2014.
33. Yuval Ishai, Amit Sahai, and David A. Wagner. Private circuits: Securing hardware against probing attacks. In *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.
34. Tendayi Kamucheka, Alexander Nelson, David Andrews, and Miaoqing Huang. A masked pure-hardware implementation of kyber cryptographic algorithm. In *FPT*, page 1. IEEE, 2022.
35. Matthias J. Kannwischer, Peter Pessl, and Robert Primas. Single-trace attacks on keccak. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(3):243–268, 2020.
36. Liran Lerman, Romain Poussier, Olivier Markowitch, and François-Xavier Standaert. Template attacks versus machine learning revisited and the curse of dimensionality in side-channel analysis: extended version. *J. Cryptogr. Eng.*, 8(4):301–313, 2018.
37. Yuejun Liu, Yongbin Zhou, Shuo Sun, Tianyu Wang, Rui Zhang, and Jingdian Ming. On the security of lattice-based fiat-shamir signatures in the presence of randomness leakage. *IEEE Trans. Inf. Forensics Secur.*, 16:1868–1879, 2021.
38. Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. Crystals-dilithium algorithm specifications and supporting documentation. *NIST Post-Quantum Cryptography Standard*, 2022.

39. Stefan Mangard. Hardware countermeasures against DPA ? A statistical analysis of their effectiveness. In *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004.
40. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.
41. Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Inf. Secur.*, 5(2):100–110, 2011.
42. Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-channel leakage of masked CMOS gates. In *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005.
43. Soundes Marzougui, Vincent Ulitzsch, Mehdi Tibouchi, and Jean-Pierre Seifert. Profiling side-channel attacks on dilithium: A small bit-fiddling leak breaks it all. *IACR Cryptol. ePrint Arch.*, page 106, 2022.
44. Loïc Masure, Valence Cristiani, Maxime Lecomte, and François-Xavier Standaert. Don't learn what you already know scheme-aware modeling for profiling side-channel analysis against masking. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(1):32–59, 2023.
45. Loïc Masure, Pierrick Méaux, Thorben Moos, and François-Xavier Standaert. Effective and efficient masking with low noise using small-mersenne-prime ciphers. In *EUROCRYPT (4)*, volume 14007 of *Lecture Notes in Computer Science*, pages 596–627. Springer, 2023.
46. Vincent Migliore, Benoît Gérard, Mehdi Tibouchi, and Pierre-Alain Fouque. Masking dilithium - efficient implementation and side-channel evaluation. In *ACNS*, volume 11464 of *Lecture Notes in Computer Science*, pages 344–362. Springer, 2019.
47. Kalle Ngo, Elena Dubrova, Qian Guo, and Thomas Johansson. A side-channel attack on a masked IND-CCA secure saber KEM implementation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(4):676–707, 2021.
48. Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of nonlinear functions in the presence of glitches. *J. Cryptol.*, 24(2):292–321, 2011.
49. Robert Primas, Peter Pessl, and Stefan Mangard. Single-trace side-channel attacks on masked lattice-based encryption. In *CHES*, volume 10529 of *Lecture Notes in Computer Science*, pages 513–533. Springer, 2017.
50. Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 142–159. Springer, 2013.
51. Gokulnath Rajendran, Prasanna Ravi, Jan-Pieter D'Anvers, Shivam Bhasin, and Anupam Chattopadhyay. Pushing the limits of generic side-channel attacks on lwe-based kems - parallel PC oracle attacks on kyber KEM and beyond. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(2):418–446, 2023.
52. Prasanna Ravi, Mahabir Prasad Jhanwar, James Howe, Anupam Chattopadhyay, and Shivam Bhasin. Side-channel assisted existential forgery attack on dilithium - A NIST PQC candidate. *IACR Cryptol. ePrint Arch.*, page 821, 2018.
53. Prasanna Ravi, Sujoy Sinha Roy, Anupam Chattopadhyay, and Shivam Bhasin. Generic side-channel attacks on cca-secure lattice-based PKE and kems. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(3):307–335, 2020.
54. Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A formal study of power variability issues and side-

channel attacks for nanoscale devices. In *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 109–128. Springer, 2011.
55. Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In *CHES*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.
56. Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Gregor Seiler, Damien Stehlé, and Jintai Ding. Crystals-kyber algorithm specifications and supporting documentation. *NIST Post-Quantum Cryptography Standard*, 2022.
57. François-Xavier Standaert and Cédric Archambeau. Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer, 2008.
58. François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.
59. Yutaro Tanaka, Rei Ueno, Keita Xagawa, Akira Ito, Junko Takahashi, and Naofumi Homma. Multiple-valued plaintext-checking side-channel attacks on post-quantum kems. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(3):473–503, 2023.
60. Rei Ueno, Keita Xagawa, Yutaro Tanaka, Akira Ito, Junko Takahashi, and Naofumi Homma. Curse of re-encryption: A generic power/em analysis on post-quantum kems. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(1):296–322, 2022.
61. Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Soft analytical side-channel attacks. In *ASIACRYPT (1)*, volume 8873 of *Lecture Notes in Computer Science*, pages 282–296. Springer, 2014.
62. Zhuang Xu, Owen Pemberton, Sujoy Sinha Roy, David F. Oswald, Wang Yao, and Zhiming Zheng. Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of kyber. *IEEE Trans. Computers*, 71(9):2163–2176, 2022.