Ondelettes pour le traitement des signaux compromettants

Gabriel Destouet

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : Mathématiques Appliquées

Arrêté ministériel : 25 mai 2016

Présentée par

Gabriel DESTOUET

Thèse dirigée par **Valérie PERRIER**,Professeur des universités, Université Grenoble Alpes

préparée au sein du **Laboratoire Jean Kuntzmann** dans **l'École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

Ondelettes pour le traitement des signaux compromettants

Wavelets for side-channel analysis

Thèse soutenue publiquement le **3 mars 2022**,
devant le jury composé de :

Madame VALERIE PERRIER
Professeur des Universités, GRENOBLE INP, Directrice de thèse
Monsieur OLIVIER MICHEL
Professeur des Universités, GRENOBLE INP, Président
Madame MARIANNE CLAUSEL
Professeur des Universités, UNIVERSITE DE LORRAINE, Rapporteure
Madame ANNELIE HEUSER
Chargé de recherche, CNRS BRETAGNE ET PAYS DE LA LOIRE, Examinatrice
Monsieur OLIVIER RIOUL
Professeur, TELECOM PARIS, Rapporteur
Monsieur GUENAEL RENAULT
Chercheur HDR, ANSSI, Examinateur

Abstract

In the context of the security assessment of electronic devices, evaluators can perform attacks on cryptographic algorithm implemented in secure cryptoprocessor. In our case, we are interested in side-channel attacks which use electrical consumption or electromagnetic signals from the device to recover sensitive information such as cryptographic keys. The leakage model, which depends on the software and the device, is generally not accessible and difficult to derive. Moreover, the evaluators also have to take into account the various conditions in which the attacks could in theory be performed and look for the best attack to anticipate attackers. Last but not least, the manufacturers can add countermeasures which will have an impact either on the hardware itself or the algorithm. Consequently, the evaluators have to continuously improve their attack methods in order to deal with complex, noisy and desynchronized signals. Actual attack methods analyze signals in the temporal domain to recover sensitive information. But the analysis in the time-frequency domain usually presents better capabilities for identifying patterns linked with the algorithm instructions. In particular, wavelet transforms provide detailed time-frequency (time-scale) maps that can be used to recover the different events linked with the cryptographic algorithm. The goal of this thesis is to develop new attack methods based on wavelet transforms, with a focus on handling jitter effects from countermeasures.

We start by presenting some tools of wavelet analysis for the visualization and extraction of patterns linked to algorithmic operations. After resynthesis in the time domain, these patterns are used as adapted filters in a simple realignment method. Next, we study the estimation of wavelet frames adapted to patterns. Without analytical properties in the signals that could help in the choice of a particular wavelet family, we drive the estimation in the superfamily of Generalized Morse Wavelets. The learned frame is then used to carry out side-channel attacks. We also study, in a subsequent work, a more direct approach that does not rely on a prior realignment of signals, for this we use the scattering transform to reduce the effect of jitter countermeasures in side-channel signals. Along with the proposed preprocessing, we study an ensemble method for the approximation of a leakage model. In a last work, we build a general statistical model for side-channel signals. We take into account a model of the algorithm, of the jitter countermeasure and of pattern generation through wavelet frames. It will be used for the generation of artificial side-channels signals and for the estimation of the time of occurrence of algorithmic operations useful for the realignment.

Résumé

Dans le cadre de l'évaluation de la sécurité de systèmes d'informations, des évaluateurs réalisent des attaques sur des algorithmes cryptographiques implémentés sur des composants sécurisés afin d'évaluer leur vulnérabilité face aux fuites d'information sensible. Dans notre cas, nous nous intéressons aux attaques par canaux-cachés, qui consistent à récupérer de l'information (la valeur d'une clé de chiffrement par exemple) en analysant des signaux de consommation électrique ou de rayonnement électromagnétique. Le modèle de fuite est généralement inaccessible et difficile à estimer. De plus, l'évaluateur doit prendre en compte les différentes conditions dans laquelle l'attaque peut être effectuée et chercher la meilleure attaque afin d'anticiper les attaquants. Enfin, le fabricant du composant peut installer des contre-mesures, qui peuvent se traduire par une modification de la structure algorithmique du logiciel ou par une altération physique du composant. Ainsi l'évaluateur doit constamment chercher de nouvelles attaques afin de pouvoir traiter des signaux de plus en plus complexes, bruités et désynchronisés. Actuellement les méthodes pour exploiter ces signaux et extraire des éléments secrets reposent usuellement sur l'analyse et l'exploitation du signal dans le domaine temporel. Or, l'analyse dans un espace temps-fréquence permet en général d'identifier plus facilement les motifs liés aux instructions algorithmiques et l'influence des contre-mesures. En particulier, les transformées en ondelettes permettent d'avoir une analyse fine des signaux et d'identifier dans l'espace temps-fréquence les différents événements liés à l'algorithme de chiffrement. L'objectif de cette thèse est de développer de nouvelles méthodes d'attaques basées sur les transformées en ondelettes. En particulier, nous nous focaliserons sur le traitement de la désynchronisation des signaux.

Nous commencerons avec la présentation d'outils d'analyse en ondelettes permettant la visualization et l'extraction des motifs présents dans les signaux et liés aux opérations algorithmiques. Ces motifs pourront être transformés en filtres adaptés pour une méthode simple de réalignement. Ensuite, nous étudions l'estimation de trames d'ondelettes adaptées aux motifs. En l'absence de propriétés analytiques dans les signaux pouvant motiver un choix particulier de famille d'ondelettes, nous emploierons la superfamille des ondelettes de Morse généralisées. La trame apprise sera ensuite utilisée pour effectuer des attaques. Dans un deuxième temps, nous laisserons momentanément les approches se basant sur des techniques de réalignement pour étudier la transformation en scattering qui permet de réduire directement des bruits de désynchronisation et de déformations dans les signaux. La méthode proposée sera couplée avec une méthode d'ensemble pour l'approximation du modèle de fuite des informations sensibles. Enfin, dans un dernier chaptitre, nous établirons un modèle statistique génératif pour les signaux par canaux cachés, construit de

manière à prendre en compte la partie algorithmique, les phénomènes de désynchronisation et la génération de motifs via une trame d'ondelettes. Il sera utilisé pour la génération de signaux par canaux-cachés et on en déduira une méthode d'estimation des temps d'apparition des opérations pour le réalignement.

Acknowledgements

J'aimerais tout d'abord remercier ma directrice de thèse Valérie Perrier au Laboratoire Jean Kuntzmann, pour le suivi et l'intérêt qu'elle a accordés à mes travaux. Les riches discussions que nous avons pu avoir sur la théorie des ondelettes et en analyse se sont révélées indispensables dans la maitrise de mon sujet de thèse.

Je remercie mes co-encadrantes Anne Frassati et Cécile Dumas au Centre d'Evaluation de la Sécurité des Technologies de l'Information au CEA:

Anne Frassati pour son enthousiasme et dynamisme qui a été un moteur dans le déroulement de ma thèse. En sa qualité de chef de laboratoire, l'attention qu'elle a apportée aux conditions de travail et au moral des doctorants a été primordiale, et cela surtout en pleine crise sanitaire.

Cécile dumas pour son aide dans mon initiation au domaine des attaques par canaux-cachés. L'aspect multidisciplinaire du sujet c'est traduit avec Cécile par des discussions enrichissantes, à l'interface entre le traitement du signal, la modélisation statistique et la cryptographie.

Enfin, je leur sais gré à toutes les trois pour les nombreuses relectures qu'elles ont pu faire de mon mémoire et de mes publications, et les corrections qu'elles ont pu y apporter.

J'aimerais remercier tous les membres du jury dont les rapporteurs Marianne Clausel et Olivier Rioul, et les examinateurs Annelie Heuser, Olivier Michel et Guenael Renault. Je les remercie d'avoir accepté de faire partie de mon jury, d'avoir pris le temps de s'intéresser à mes travaux, et pour les intéressants échanges lors de ma soutenance.

Je suis reconnaissant aux membres du CESTI et du LJK pour leur accueil, et notamment Jessy, Eleonora, Marie-Angela, Laurent, Claire et Vincent pour les fructueuses discussions au sujet des attaques par canaux cachés et de la cybersécurité plus généralement. Je salue bien sur tous les stagiaires et doctorants que j'ai rencontré, Loic, Antoine, Hubert, Vincent, Karim, Emrick, Esther et Pierre-Alain avec qui j'ai partagé de bon moments.

Enfin je remercie ma famille et mes amis qui m'ont soutenu dans cette aventure.

Nomenclature

$\mathbb{N}$	Set of positive integers
$\mathbb{Z}$	Set of positive and negative integers
$\mathbb{R}$	Set of real numbers
$\mathbb{C}$	Set of complex numbers
$\mathbb{K}$	Either $\mathbb{R}$ or $\mathbb{C}$
$\mathcal{C}^n$	Set of vectors in $\mathbb{R}^{+n}$ summing to one, i.e. a n-simplex
$C(\mathbb{K})$	Set of continuous functions on $\mathbb{K}$
$\mathcal{L}^1(\mathbb{R})$	Set of absolute integrable functions on $\mathbb{R}$
$\mathcal{L}^2(\mathbb{R})$	Set of square integrable functions on $\mathbb{R}$
$\ell^2(\mathbb{Z})$	Set of finite energy sequences
$A \times B$	Cartesian product of sets A and B
$\lvert A \rvert$	Cardinality of set A
i	Imaginary number
$\Re(z)$	Real part of z
$\Im(z)$	Imaginary part of z
$\overline{z}$	Conjugate of z
$f * g$	Convolution of functions or vectors f and g
$f \circledast g$	Circular convolution of functions f and g
$\langle u, v \rangle$	Inner product of vectors u and v
$\mathrm{vec}(A)$	Vectorize operator transforming matrix A into a column vector

A^*	Adjoint of linear operator A
A^T	Transpose of matrix or vector A
$A^\dagger$	Conjugate transpose of matrix A, i.e. $\overline{A^T}$
$\nabla_x f$	Jacobian of f according to variable x
$\mathrm{tr}\,(A)$	Trace of matrix A
$\mathcal{F}$	Fourier transform operator
$\widehat{f}$	Fourier transform of f
STFT	Short-Time Fourier Transform operator
δ_τ	Dirac distribution centered at τ
$\mathbf{1}_{b(x)}$	Indicator function evaluated to 1 if the condition $b(x)$ is true, 0 otherwise
L_τ	Translation operator parametrized with translation τ
S_a	Scaling operator parametrized with scale a
f^-	Time reversal and conjugate version of a function f, i.e. $f^-(t) = \overline{f}(-t)$
$\mathbb{E}_X\left[f(x)\right]$	Expectation of f according to the probability measure of X

Contents

List of Figures

List of Tables

Introduction

The invention of silicon based metal-oxide-semiconductor at the beginning of the 20th century by the Bell Labs has been one of the most important breakthroughs in technology and science. It was the first semiconductor to be massively produced and greatly revolutionized the field of electrical engineering. It led to the development and industrialization of modern computers which in turn transformed our life and culture.

The use of computers is now widespread and rules our daily lives. Over time, these devices have grown in complexity and functionality, and today the knowledge required to fully understand the physical theory and computer science behind them is unreachable for a single person. Thus, in our daily use of any computer device, we trust engineers and computer scientists to make sure it meets our expectations.

However, for some applications such as information security, it is important that tiers verify that the device meets its specifications. Indeed, the development of information technology also led to new scam and fraud techniques that threatens the use we can make of these devices and the trust we placed in them. Nowadays, government agencies and independent information security laboratories act as trusted third parties who can verify the security of the devices handling our personal and secret information.

This thesis was held in an Information Technology Security Evaluation Facility (ITSEF) located at Grenoble in France at the CEA, specialized in the evaluation of the security of information processing devices. In order to assess the security of a device, information security evaluators ensure first that the product meets its specifications and that the system has no design flaws, next he may perform high-level attacks, including physical attacks, to test its security. In consequence, and with the development of new countermeasures by manufacturers and developers to meet security standards, their attack methods have to be continuously improved to anticipate attackers.

In this context, many types of attacks may pose a threat on a cryptographic device responsible for storing or ciphering sensible information. As an important type of attack, side-channel analysis may monitor indirect information about the functioning of the device to break its security and recover sensible information such

as cryptographic keys.

In particular, as the cost of frauds on smart cards has increased in recent years, the security assessment of those devices is of importance. To do so, evaluators wish to measure the leak of cryptographic keys in electrical consumption or electromagnetic signals emitted by smart cards during operation. They can rely on statistical models such as Gaussian templates [21] or statistical tests [66] to test the leak of the key. In response to those security tests, manufacturers and developers have been designing efficient countermeasures to decrease the leak of information in signals, such as masking countermeasures [58] that share the secret about the key across variables in the algorithm, and jitter countermeasures [26] that desynchronize signals to perturb their statistical analysis. The introduction of deep learning methods in recent years [78, 28] has produced new state-of-the-art attack results that pose a threat against those countermeasures. At the same time, however, the absence of an explicit model of the side-channel signal in deep learning methods makes it difficult for the evaluator to fully grasp the nature of the leak and propose a more detailed security analysis to the manufacturers and developers.

In this thesis, we will adopt a signal processing approach in the analysis of side-channel signals. In particular, we will study the use of wavelet transforms for analyzing signals. Its use is not widespread in the side-channel community although it has been fruitfully employed as a preprocessing step in other fields presenting the same kind of problems such as for example in audio [6] or heart-rate analysis [2]. Wavelet transforms provide detailed time-frequency maps by projecting signals on a specific sequence of elementary signals. It provides multiresolution representation that helps in the identification of different components in the signal. We will study their use in the context of side-channel analysis and propose new attacks methods.

Outline

We wish this manuscript to be readable by people not particularly initiated to signal processing or probability theory. Thus, we devote the first two chapters to some elements of those fields that will be used throughout this manuscript. A third chapter will present the context of application of this thesis, namely information security and side-channel analysis. The fourth and fifth chapters are dedicated to our contributions to the field through the lens of wavelet analysis.

We present in the first chapter some concepts and tools in signal processing with a focus on time-frequency analysis. We recall the definitions of Fourier, Short-Time Fourier and Wavelet transforms. To understand the difference between various time-frequency transforms we detail the topic of time-frequency localization via the paving of the time-frequency space with Heisenberg areas or by the introduction of

time-frequency localization operators. In particular, we recall in this last case the construction of the superfamily of Generalized Morse Wavelets that will be used in this thesis for analyzing signals.

The second chapter is dedicated to information and probability theory. We present our notations and recall the definition of the Shannon entropy and the mutual information. Next, we introduce all the underlying statistical elements that will be used in our methods. We present in particular the learning algorithm behind the template attacks of [21].

In a third chapter, we detail some background on information security and side-channel analysis. It is an important chapter as it depicts the context in which this thesis has been carried out. We present the problem of evaluating the security of implemented cryptosystems on smart-cards and the goal of side-channel analysis. We will end by the literature in side-channel analysis and the positioning of the works of this thesis.

In a fourth chapter, we present different methods developed during this thesis where we did not consider a dynamical structure for the signals. We start with simple tools in wavelet analysis to visualize side-channel signals and extract patterns for realignment. We propose a novel method for the estimation of an adapted frame of Generalized Morse Wavelets which is applied in the context of side-channel analysis for analyzing extracted patterns from side-channel signals. It has been published in [37]. The chapter ends with the method published in [38] for reducing the impact of jitter countermeasures with the scattering transform, and for approximating a leakage model with an ensemble method.

In a fifth chapter, we present a generative and dynamical model for side-channel signals. In comparison with the fourth chapter, we do assume in this part a dynamical structure to side-channel signals and propose a model for it. More specifically, we model the jitter by a point process model and use our previous work on the estimation of an adapted frame to model the generation of patterns related to algorithmic operations.

Chapter 1

Time-Frequency Analysis of Signals

In this chapter, we introduce important concepts and tools in time-frequency analysis. Since the subject of this thesis is at the crossroad between signal processing, machine learning and cryptanalysis, our goal in this part is to provide essential elements of time-frequency analysis to readers less familiar with the field.

We start by giving some general mathematical properties to our signals which are seen as functions or vectors. Then we go through essential recalls on Fourier analysis in order to introduce in a subsequent section time-frequency analysis and wavelet analysis. Next, a section is dedicated to the topic of time-frequency localization as it gives, in its way, some theoretical arguments for choosing a wavelet basis.

For more details on the foundations of signal processing we refer the reader to [96] or [116], and elements of function analysis can be found in [104].

1.1 Spaces of signals

Signals are represented as functions f on $\mathbb{R}^k$ taking values in $\mathbb{K}$ where $\mathbb{K}$ designates either the set of real numbers $\mathbb{R}$ or complex numbers $\mathbb{C}$. For example, a temporal signal is represented by a function $f : \mathbb{R} \to \mathbb{R}$, while its wavelet transform Wf, which we still consider as a *signal*, can be defined as $Wf : \mathbb{R}^2 \to \mathbb{C}$.

After discretization, signals are internally represented in computer memory as n-multidimensional vectors y in $\mathbb{K}^n$. To continue the previous example, the sampling and discretization of a temporal signal f can result in a vector $\mathbb{R}^n$ where n is the number of samples acquired.

We use a different notation to index functions and vectors: we note $f(x)$ the value of f at the *continuous index* x with x defined in an uncountable space $\mathbb{R}^k$ and $y[u]$ the value taken by the vector y at the *discrete index* u in a countable space.

In particular, if we have an indexed sequence $\{v_a \in \mathbb{K}\}_{a\in A}$ with A countable, we can define a function f that maps the indices to the values of the sequence, i.e. $f : A \to \mathbb{K}$, $a \mapsto f[a] = v_a$. For example, A can be a countable subspace of $\mathbb{R}^n$.

In practice, the observation of signals and consequently the analysis is limited in a time and frequency domain of observation. It results that we can safely consider our signals of finite energy. Hence, we introduce the two following spaces of functions.

The Lebesgue space of integrable functions is defined by

$$\mathcal{L}^1(\mathbb{R}) = \left\{ f : \mathbb{R} \to \mathbb{K} \;\middle|\; \int_{\mathbb{R}} |f(x)|dx < \infty \right\},$$

and for the Lebesgue space of square integrable functions we write

$$\mathcal{L}^2(\mathbb{R}) = \left\{ f : \mathbb{R} \to \mathbb{K} \;\middle|\; \int_{\mathbb{R}} |f(x)|^2dx < \infty \right\}.$$

Later on, to introduce some analysis tools we will need a discrete version of the $\mathcal{L}^1(\mathbb{R})$. The space of finite energy sequences writes

$$\ell^2(\mathbb{Z}) = \left\{ y : \mathbb{Z} \to \mathbb{K} \;\middle|\; \sum_{u\in\mathbb{Z}} |y[u]|^2 < \infty \right\}.$$

$\mathcal{L}^2(\mathbb{R})$ and $\ell^2(\mathbb{Z})$ are Hilbert spaces. We have the canonical inner product on $\mathcal{L}^2(\mathbb{R})$

$$f, g \in \mathcal{L}^2(\mathbb{R}) \;, \langle f, g\rangle = \int_{\mathbb{R}} f(x)\overline{g(x)}dx \,,$$

with $\bar{\cdot}$ the complex conjugation. It induces the norm

$$\|f\| = \sqrt{\langle f, f\rangle}\,.$$

On $\ell^2(\mathbb{Z})$, the inner product is written

$$y, z \in \ell^2(\mathbb{Z}) \,, \langle y, z\rangle_\ell^2 = \sum_{u\in\mathbb{Z}} y[u]\overline{z[u]} \,,$$

and the norm is written in the same way.

As said previously, all our methods ultimately manipulate signals in the form of n-dimensional vectors in $\mathbb{K}^n$ that can be seen as continuous signals acquired over a time range $[-T/2, T/2]$ with $T = n/F_s$ and F_s the sampling frequency. Thus, as the continuous counterpart of $\mathbb{K}^n$, we will use $\mathcal{L}^2([-T/2, T/2])$ as the space of finite energy signals on $[-T/2, T/2]$.

The space of continuous functions over a domain $\mathbb{R}$ is noted $C(\mathbb{R})$. In addition to the continuity property, the k-th derivative of a signal f, if it exists, is noted $f^{(k)}$

and we note $C^k(\mathbb{R})$ the space of k-differentiable functions on $\mathbb{R}$.

1.2 Some tools of signal analysis

1.2.1 Operations

Integration with the Dirac distribution We define the Dirac distribution δ as the distribution on $\mathbb{R}$ fulfilling the property

$$\forall f \in \mathcal{L}^2(\mathbb{R}), \tau \in \mathbb{R}, \quad \int_{-\infty}^{+\infty} f(t)\delta(t-\tau)dt = f(\tau) \tag{1.1}$$

We note $\delta_\tau(t) = \delta(t-\tau)$ the τ-delayed Dirac distribution.

Convolutional operator The convolution operator is particularly useful to represent how a system (physical or idealized) acts on input signals. Let $g \in \mathcal{L}^2(\mathbb{R})$ a function that characterizes the action of the system and $f \in \mathcal{L}^2(\mathbb{R})$ an input signal, the effect of the system on f is the result of the convolution operator

$$f * g(\tau) = \int_{-\infty}^{+\infty} f(t)g(\tau - t)dt\,. \tag{1.2}$$

Now for time-limited signals $f, g \in \mathcal{L}^2([-T/2, T/2])$, we introduce the *circular* convolution operator

$$f \circledast g(\tau) = \int_{-T/2}^{+T/2} f(t)g(\tau - t)dt\,, \tag{1.3}$$

where it is implicitly assumed that the functions f, g are periodized with period T, i.e. $g(t) = g(t - kT)$ with $k \in \mathbb{Z}$, thus $f \circledast g$ is a T-periodic function.

The definition of convolution operators for discrete sequences comes naturally by discretizing the integrals in previous definitions.

For sequences $y, z \in \ell^2(\mathbb{Z})$ we keep the notation $y * x$

$$y * z[k] = \sum_{p\in\mathbb{Z}} y[p]z[k-p]\,. \tag{1.4}$$

For finite vectors $y, z \in \mathbb{C}^n$, $n \in \mathbb{N}$, the convolution is circular

$$y \circledast z[k] = \sum_{p=0}^{n-1} y[p]z[k-p]\,, \tag{1.5}$$

where it is implicitly understood that $z[p] = z[p \bmod n]$ making y and z circular vectors of period n.

Translation With these definitions, we define translations L_τ of length τ on $\mathcal{L}^2(\mathbb{R})$ as the convolution with a τ delayed Dirac distribution: $\forall f \in \mathcal{L}^2(\mathbb{R}), \tau \in \mathbb{R}$,

$$L_\tau f(t) = f * \delta_\tau(t) = \int_{-\infty}^{+\infty} f(u)\delta(t-\tau-u)du = f(t-\tau). \tag{1.6}$$

Now for time-bounded signals f on $\mathcal{L}^2([-T/2, T/2])$, the action of L_τ is the circular convolution with the delayed Dirac distribution, the resulting signal is circularly shifted around $[-T/2, T/2]$.

We have the same definitions for discrete sequences or vectors by considering translations with integer values.

1.2.2 Fourier Analysis

In this section, we recall some elements of Fourier analysis.

1.2.2.1 The Fourier Transform

For $f \in \mathcal{L}^2(\mathbb{R}) \cap \mathcal{L}^1(\mathbb{R})$ the Fourier Transform $\mathcal{F}f$ is defined as

$$\forall w \in \mathbb{R},\ \mathcal{F}f(w) = \int_{-\infty}^{+\infty} f(t)e^{-iwt}dt. \tag{1.7}$$

To ease the reading, we note $\widehat{f} = \mathcal{F}f$ the Fourier Transform of f.

For $\widehat{f} \in \mathcal{L}^2(\mathbb{R}) \cap \mathcal{L}^1(\mathbb{R})$, the Inverse Fourier Transform f of $\widehat{f}$ is given by

$$\forall t \in \mathbb{R},\ f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \widehat{f}(w)e^{iwt}dw. \tag{1.8}$$

The Fourier transform (and its inverse) can be extended to $\mathcal{L}^2(\mathbb{R})$ and is an isometry of $\mathcal{L}^2(\mathbb{R})$, see [17, Chap. C3].

As it is an isometry, the Fourier Transform preserves the scalar product on $\mathcal{L}^2(\mathbb{R})$, this is known as the Parseval's identity

$$f, g \in \mathcal{L}^2(\mathbb{R}),\ \langle f, g\rangle = \frac{1}{2\pi}\langle \mathcal{F}f, \mathcal{F}g\rangle. \tag{1.9}$$

1.2.2.2 Fourier Series

We also recall here the definition of Fourier series for T-periodic signals in $\mathcal{L}^2([-T/2, T/2])$.

For all f such that $\forall t \in \mathbb{R}, f(T+t) = f(t)$, the Fourier series of f is defined as

$$\forall k \in \mathbb{Z}, w_k = 2\pi k/T,\ \mathcal{F}f[w_k] = \frac{1}{T}\int_{-T/2}^{+T/2} f(t)e^{-iw_k t}dt, \tag{1.10}$$

with the Fourier Series inversion formula given by

$$t \in \mathbb{R},\ f(t) = \sum_{k \in \mathbb{Z}} \widehat{f}[w_k] e^{i w_k t}\,. \tag{1.11}$$

where the equality is taken almost everywhere, see [17, Chap. C4].

We also have the Parseval identity

$$f, g \in \mathcal{L}^2([-T/2, T/2])\,,\ \langle f, g\rangle = T\langle \mathcal{F}f, \mathcal{F}g\rangle\,. \tag{1.12}$$

1.2.2.3 Discrete Fourier Transform

For a n-dimensional vector f in $\mathbb{R}^n$, the Discrete Fourier Transform of f is defined as

$$\forall k \in \{0, \dots, n-1\}, w_k = 2\pi k/n,\ \mathcal{F}f[w_k] = \sum_{p=0}^{n-1} f[p] e^{-i w_k p}\,, \tag{1.13}$$

The definition is restricted to $\{0, \dots, n-1\}$, since for $k \in \mathbb{Z}$ we have $\mathcal{F}f[w_{k+n}] = \mathcal{F}f[w_k]$. Its inversion formula is given by

$$\forall p \in \{0, \dots, n-1\},\ f[p] = \frac{1}{N} \sum_{k=0}^{n-1} \widehat{f}[w_k] e^{i w_k p}\,. \tag{1.14}$$

The Parseval's identity gives

$$\langle f, g\rangle = \frac{1}{N}\langle \mathcal{F}f, \mathcal{F}g\rangle\,. \tag{1.15}$$

Properties

Convolution For $f, g \in \mathcal{L}^2(\mathbb{R})$, we have the useful identity

$$\forall w \in \mathbb{R}, \mathcal{F}(f * g)(w) = \widehat{f}(w)\widehat{g}(w) \tag{1.16}$$

On bounded domains or for finite dimensional vectors, *i.e.* for f, g in $\mathcal{L}^2([-T/2, T/2])$ or $\mathbb{C}^N$ with $n = TF_s$, F_s the sampling frequency, we have

$$\forall k \in \mathbb{Z}, w_k = 2\pi k F_s/n,\ \mathcal{F}(f \circledast g)[w_k] = \widehat{f}[w_k]\widehat{g}[w_k]\,. \tag{1.17}$$

Scaling and Translation Let L_u a translation operator such that for $f \in \mathcal{L}^2(\mathbb{R})$, $L_u f(t) = f(t-u)$, we have

$$\mathcal{F}L_u f(w) = e^{-iwu}\widehat{f}(w)\,. \tag{1.18}$$

Now, noting $S_a, a \in \mathbb{R}^*$ the scaling operator such that for $f \in \mathcal{L}^2(\mathbb{R})$, $S_a f(t) = f(t/a)$. The Fourier transform of a scaled function is

$$\mathcal{F} S_a f(w) = |a| \widehat{f}(aw) \,. \tag{1.19}$$

Composing the two previous transformations such that $S_a L_u f(t) = f\left(\frac{t-u}{a}\right)$ and taking the Fourier transform we obtain:

$$\mathcal{F} S_a L_u f(w) = |a| e^{-iwu} \widehat{f}(aw) \,. \tag{1.20}$$

Note that, if we commute S_a and L_u we get a different result. We have $L_u S_a f(t) = f(t/a - b)$ and the Fourier transform gives

$$\mathcal{F} L_u S_a f(w) = |a| e^{-iwua} \widehat{f}(aw) \,. \tag{1.21}$$

1.3 Time and Frequency Analysis

In previous section, we introduced the Fourier transform which can be represented as the projection of functions on the Fourier Basis

$$\{t \to e^{iwt}\}_{w \in \mathbb{R}} \,.$$

However, it is not always practical to manipulate and interpret the representation of a signal in the Fourier Basis. The complex exponentials of the Fourier transform capture information on the whole time-domain and it is only by the weighted combination of many elements of the Fourier basis that we are able to reproduce local variations. In other terms, if we wish to properly analyze an event in a bounded and small time domain, we will have to extract information from a lot of Fourier coefficients. In this thesis, it will be of particular importance to analyze non-stationary signals with random transients, it requires adapted tools to extract information from restricted domain of time and frequency.

Thus, the rest of this chapter is dedicated to *time-frequency* transformations. They are more adapted to the analysis of transients in bounded time-frequency domains. We will first start with the Short-Time Fourier transform (STFT), which historically is the first time-frequency transform presented by D. Gabor in [43]. Then we introduce the wavelet transform, proposed by J. Morlet in the 80s and whose mathematical foundations are laid in [47]. A small section will be dedicated to the topic of frame analysis, which is convenient to summarize both previous transforms.

More details on time-frequency analysis can be found in the books of Meyer [123], Daubechies [31] and Mallat [80].

1.3.1 The Short-Time Fourier Transform

We define the Short-Time Fourier Transform (STFT) of a signal f as the projection on a family of Gabor atoms

$$\left\{\phi_{w,u} : t \to g(t-u)e^{iwt} \;\middle|\; w, u \in \mathbb{R}\right\},$$

where g is a function, called the *window*, generally chosen real, symmetric and vanishing outside a bounded time domain.

The STFT of f is thus defined by

$$\forall u, w \in \mathbb{R},\ \mathrm{STFT}f(u,w) = \langle f, \phi_{w,u}\rangle\,. \tag{1.22}$$

In practice, a common approach is to compute the inner products by moving the window g along the time domain and by taking Fourier transforms. We can show that

$$\langle f, \phi_{w,u}\rangle = \mathcal{F}(f\overline{g_u})(w)\,, \tag{1.23}$$

with $\bar{\cdot}$ the complex conjugation and $g_u : t \to g(t-u)$.

Alternatively, the STFT can be computed by convoluting f with

$$\phi_w^-(t) = \overline{\phi_w}(-t)$$

for each $w \in \mathbb{R}$ with $\phi_w = \phi_{w,0}$, and by sampling each convolution.

Indeed, we have

$$\begin{aligned} f * \phi_w^-(u) &= \int f(t)\phi_w^-(u-t)dt && (1.24)\\ &= \int f(t)\overline{\phi_w}(t-u)dt && (1.25)\\ &= e^{iwu}\langle f, \phi_{w,u}\rangle\,. && (1.26) \end{aligned}$$

Thus by sampling the convolutions at each $u \in \mathbb{R}$ and by compensating the phase term we are able to form the STFT of our signal.

The STFT is commonly computed on a discrete and regular time-frequency grid

$$\{(u_k, w_p) \mid k, p \in \mathbb{Z}, u_k = u_0 + kc_u, w_p = w_0 + pc_w\}\,,$$

with $u_o, w_o \in \mathbb{R}$ and $c_u, c_w \in \mathbb{R}^+$.

In our case, we prefer computing the STFT through convolutions as it allows the processing of large signals with a convolution operator implemented with the overlap-add algorithm [116, Sec. 3.9]. Moreover, the convolutions can be computed

only for pre-selected frequencies of interest. It is a procedure that will be used later for computing continuous wavelet transforms.

Generalizations of the STFT on non-regular time-frequency grid are given in [10]. But, it is generally difficult to deal with true non regular time-frequency grids as a fast algorithm to perform the calculations is missing and the grid must be adapted to the signal. Frame analysis, which will be introduced later, provides a mathematical framework to deal with such ideas.

1.3.2 Wavelet Transform

We previously defined the short-time Fourier transform as the projection of a signal on a family of shifted and modulated time-window. The main limitation of this transformation is that the time-window of analysis is fixed. The wavelet transform uses elementary signals called *wavelets* whose scales are adapted in order to capture fast or slow variations in signals.

1.3.2.1 Wavelet Bases

A wavelet is a function $\psi_{a,u}$ with $a \in \mathbb{R}_*^+$ and $u \in \mathbb{R}$ such that

$$\forall t \in \mathbb{R},\ \psi_{a,u}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-u}{a}\right) \tag{1.27}$$

$\psi \in \mathcal{L}^2(\mathbb{R})$ is called the mother wavelet as it generates all translated and scaled variants $\{\psi_{a,u}\}_{a,u}$. For increasing values of a, ψ is dilated in time, and conversely it is contracted for decreasing values. It has zero mean

$$\int_{-\infty}^{+\infty} \psi_{a,u}(t) = 0 \text{ or equivalently } \widehat{\psi}_{a,u}(0) = 0 \tag{1.28}$$

And we say that a wavelet ψ is *admissible* if ψ is real or analytic complex (*i.e.* $\widehat{\psi} = 0$ on $\mathbb{R}^-$) and

$$C_\psi = \int_0^{+\infty} \frac{|\widehat{\psi}(w)|^2}{w} dw < +\infty\,. \tag{1.29}$$

1.3.2.2 Wavelet Transform

The wavelet transform Ψf of a signal $f \in \mathcal{L}^2(\mathbb{R})$ is defined as the set of inner products with scaled and dilated wavelets $\psi_{a,u}$

$$(\Psi f)(a,u) = \int_{\mathbb{R}} f(t)\overline{\psi_{a,u}}(t) = \langle f, \psi_{a,u}\rangle\,. \tag{1.30}$$

The criteria of admissibility (1.29) ensures that (Ψf) can be used to recover f using

$$f = \frac{1}{C_\psi} \int_0^{+\infty} \int_{-\infty}^{\infty} (\Psi f)(a,u)\psi_{a,u} \frac{da}{a^2} du \,. \tag{1.31}$$

The continuous wavelet transform is made of the convolutions $\{t \to (f * \psi_a^-)(t)\}_a$ with $\psi_a^-(t) = \frac{1}{\sqrt{a}}\overline{\psi(\frac{-t}{a})}$ for each scale $a \in \mathbb{R}^+$. The algorithm for computing the continuous wavelet transform is easy to implement, and with it we can get continuous representations of signals in the time-scale space.

Discrete wavelet transforms are optimized to compute wavelet transforms. The parameters $\{a, u\}$ are carefully chosen such that for a signal $f \in \mathbb{R}^n$, at least n scalar products $\langle f, \psi_{a,u} \rangle$ are required to reconstruct the signal f. For a suitable choice of the mother wavelet ψ, it requires that the chosen set of parameters $\{a, u\}$ makes $\{\psi_{a,u}\}$ a basis of $\mathbb{R}^n$.

For real signals, a fast wavelet transform presented in [79] cascades quadrature mirror filters and 1/2-subsampling to rapidly compute a discrete wavelet transform on an orthogonal wavelet basis. It requires a low and high pass filters mirrored at pulsation $\pi/2$ in the frequency domain.

1.3.3 Analysis in a Frame

Short-Time Fourier and Wavelet transforms are projections on particular sequences of functions. These sequences generally constitute a *basis* of the space of analysis which requires its elements to be chosen linearly independent and orthogonal to each other. This constraint can be dropped by considering *frames* of functions which are overcomplete or redundant bases that can still represent each element of the space of analysis. Frame theory has been first introduced in [39] for reconstructing signals from non harmonic Fourier series representations. Further details on frame theory can be found in [25] and [80].

1.3.3.1 Frame operator

Let H an Hilbert space, f a signal in H and Ψ an operator parametrized with a sequence of functions $\{\psi_\xi\}_{\xi \in \Xi}$ with the index set Ξ countable. The action of Ψ on f gives the projections $\Psi f[\xi] = \langle f, \psi_\xi \rangle, \xi \in \Xi$. Its adjoint Ψ^* is defined as

$$\forall\, x \in \ell^2(\Xi)\,,\ \Psi^* x = \sum_{\xi \in \Xi} x[\xi]\psi_\xi \,. \tag{1.32}$$

In general, the sequence $\{\psi_\xi\}_{\xi \in \Xi}$ does not necessarily span the whole space H. Thus the operator Ψ is not invertible on H but only on a subspace $V = \text{span}(\{\psi_\xi \mid \xi \in \Xi\})$.

We say that Ψ is a *frame operator* on a subspace V if there exist $0 < A \leq B$ such that [80, Def. 5.1]:

$$\forall f \in V,\ A\|f\|^2 \leq \sum_{\xi\in\Xi} |\langle f, \psi_\xi\rangle|^2 \leq B\|f\|^2 \tag{1.33}$$

The bound A (B) can be found by taking the infimum (supremum) of the spectrum of the self-adjoint operator $\Psi^*\Psi$. If A and B are finite positive, then $\Psi^*\Psi$ is invertible and we can look for the dual operator $\tilde{\Psi}$ defined by the dual frame $\{\tilde{\psi_\xi} = (\Psi^*\Psi)^{-1}\psi_\xi\}$, see [80, Th. 5.5]. This dual operator is also a frame operator with bounds $0 < B^{-1} \leq A^{-1}$ and its adjoint is the pseudo-inverse of Ψ, $\tilde{\Psi}^* = (\Psi^*\Psi)^{-1}\Psi^*$.

The dual operator is useful to get a reconstruction identity on V, we have $\mathrm{Id}_V = \tilde{\Psi}^*\Psi = \Psi^*\tilde{\Psi}$. For a signal f in H, its projection f_V on V can be computed with $f_V = \tilde{\Psi}^*\Psi f$

The set of indices Ξ of the frame operator allows us to uniquely identify each element of the frame. For example, the Short-Time Fourier transform can be seen as the action of a frame operator with a frame of modulated and shifted window functions $\{g(t-u)e^{iwt}\}_{(u,w)\in\Xi}$, and where we can choose Ξ as the discrete time-frequency grid

$$\Xi = [0, \ldots, n-1] \times [0, \ldots, 2\pi k/n, \ldots, 2\pi(n-1)/n],\ n \in \mathbb{N}\,.$$

Wavelet transforms are also represented this way by considering frames of scaled and shifted versions of a mother wavelet.

In the discrete setting, the adjoint of the frame operator will be written $\Psi^\dagger$ with $\cdot^\dagger$ the conjugate transpose operation for matrices. Each column of $\Psi^\dagger$ will contain the elements of the frame.

1.3.4 Paving the time-frequency space

To understand why wavelet transforms are adapted for analyzing fast varying signals on short time scales, we can look at the first and second-order energy moments of a family of wavelets. For an arbitrary signal $f \in \mathcal{L}^2(\mathbb{R})$ with Fourier transform $\widehat{f}$, its first and second energy moments in time μ_t, σ_t, and frequency μ_f, σ_f are given by:

$$\mu_t(f) = \frac{1}{\|f\|^2}\int_{\mathbb{R}} t|f(t)|^2 dt \qquad \mu_f(f) = \frac{1}{\|f\|^2}\int_{\mathbb{R}} w|\widehat{f}(w)|^2 dw$$
$$\sigma_t(f)^2 = \frac{1}{\|f\|^2}\int_{\mathbb{R}} (t-\mu_t(f))^2|f(t)|^2 dt \quad \sigma_f(f)^2 = \frac{1}{\|f\|^2}\int_{\mathbb{R}} (w-\mu_f(f))^2|f(w)|^2 dw$$

With these quantities, we can form the Heisenberg area $\sigma_t\sigma_f$, centered at (μ_t, μ_f), where most of the energy of f is localized.

Let ψ a mother wavelet and

$$\psi_a(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t}{a}\right)$$

the a-scaled versions, we assume that all wavelets here are centered at $t = 0$, *i.e.* $\forall a \in \mathbb{R}^+, \mu_t(\psi_a) = 0$. The first and second energy moments for ψ_a can be expressed in terms of the mother wavelet's, we get:

$$\mu_t(\psi_a) = \mu_t(\psi) \qquad \mu_f(\psi_a) = \frac{1}{|a|}\mu_f(\psi)$$
$$\sigma_t(\psi_a) = |a|\sigma_t(\psi) \qquad \sigma_f(\psi_a) = \frac{1}{|a|}\sigma_f(\psi)$$

For the wavelet transform, the time-frequency box for ψ_a of time length and frequency length $\sigma_t(\psi_a)$ and $\sigma_f(\psi_a)$ respectively has a constant area $A(\psi) = \sigma_t(\psi)\sigma_f(\psi)$ and centered at $(\mu_t(\psi), \frac{1}{|a|}\mu_f(\psi))$. As a increases the Heisenberg area is elongated along the frequency axis and simultaneously shrunk along time. Wavelet transforms are thus particularly adapted to capture high variations over short period of times.

For the short time Fourier transform, the modulated Gabor atoms $\{\phi_w(t) = g(t)e^{iwt}\}_w$ have the following relation between the energy moments of ϕ_0 and ϕ_w:

$$\mu_t(\phi_w) = \mu_t(\phi_0) \qquad \mu_f(\phi_w) = w + \mu_f(\phi_0)$$
$$\sigma_t(\phi_w) = \sigma_t(\phi_0) \qquad \sigma_f(\phi_w) = \sigma_f(\phi_0)$$

The time-frequency area covered in energy by both transformations are different. In comparison with wavelet transforms, the second-energy moments of Gabor atoms stay the same as the modulation w varies. We illustrate Fig. 1.1 the variation of the Heisenberg areas in the time-frequency domain for Wavelet and Short-Time Fourier transforms.

The topic of this section is related to the problem of localizing information in the time-frequency space. In Sec. 1.4, we go a bit further by considering projection operators restricting signals in bounded time-frequency regions.

1.4 Time-Frequency Localization

In previous sections, we explored ways to analyze signals through the use of a sequence of elementary functions. The projection of the signal on one of those functions

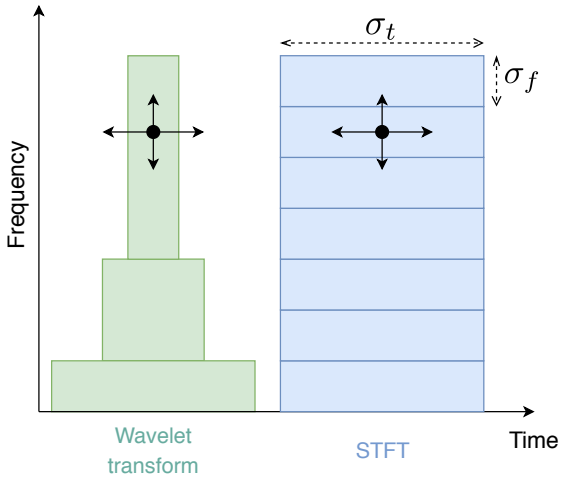


Figure 1.1: Illustration of the Heisenberg areas covered by Wavelet and Short-Time Fourier transforms. We represented the stability of both transformations under time and frequency translations by the displacement of the black dot. At high-frequency the wavelet transform is more sensitive to time translations, while the STFT is sensitive to frequency translations.

capture the energy of the signal in a time-frequency region. Instead of characterizing the signal through this sequence only, we present here a generalisation of this approach using a sequence of linear operators instead. Each operator will be localized in the time-frequency domain. In this section, we give some elements on the topic of time-frequency localization and how it led to the finding of a superfamily of wavelets, namely the Generalized Morse Wavelets, that will be used in our work. This problem has been initiated and studied in a series of articles [59, 109, 108].

The general idea is as follow. In order to analyze a signal f inside a time-frequency region S, we define an operator P_S such that the projection $P_S f$ is "localized" in S. If P_S is symmetric and positive semi-definite then we can look for the remaining energy $\langle f, P_S f\rangle \geq 0$ after projection which gives:

$$\langle f, P_S f\rangle = \sum_{k=0}^{+\infty} \lambda_k |\langle f, h_k\rangle|^2 \tag{1.34}$$

where λ_k are the eigenvalues sorted in decreasing order and h_k the eigenvectors associated. In particular, the eigenvectors and eigenvalues will depend on the region S considered.

Depending on the shape of the region S and the operator P_S, we will see that it is possible to get general expressions for the eigenvectors h_k. As a direct consequence, it allows us for example to use the first eigenvector h_0 for each region S as a elementary signal for analyzing f, i.e. for each S we may compute $\langle f, h_0\rangle$. We will be able to form a family of elementary signals to analyze f in many different regions S across the time-frequency space.

1.4.1 Analysis in a time-frequency box

Let P_T and Q_W the operators respectively truncating signals $f \in \mathcal{L}^2(\mathbb{R})$ on a time support $[-T, T]$ and a frequency bandwidth $[-W, W]$ such that:

$$(Q_T f)(t) = \begin{cases} f(t) & \text{if } |t| \leq T \\ 0 & \text{if } |t| \geq T \end{cases} \tag{1.35}$$

$$(P_W f)(t) = \int_{-\infty}^{+\infty} \frac{\sin(W(t-t'))}{\pi(t-u)} f(u)du \tag{1.36}$$

An operator $L_{T,W} = Q_T P_W$ can then be defined to model the analysis of f below a frequency W and during a time T. This operator will be bounded, positive semi-definite and symmetric for the inner product defined on $\mathcal{L}^2(\mathbb{R})$. Its time-frequency region $S = [-T, T] \times [-W, W]$ is shown on Fig. 1.2.

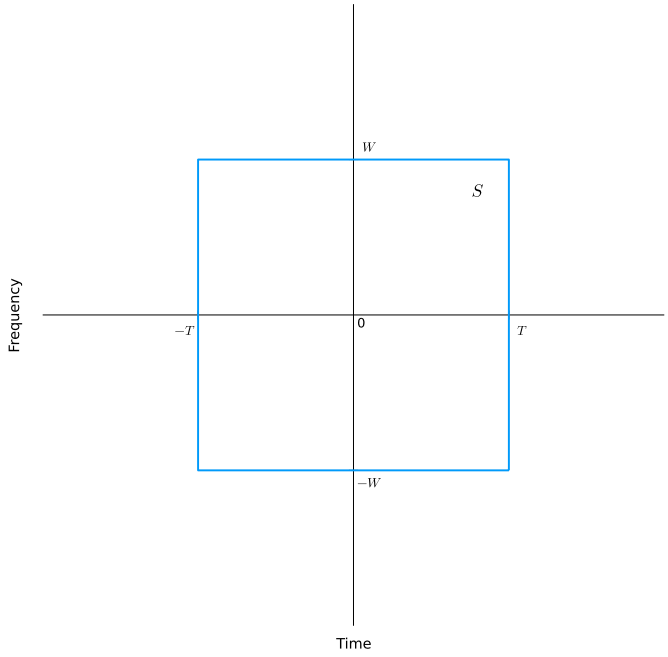


Figure 1.2: Illustration of the projection region of operator $L_{T,W}$.

Although we cannot restrict a function $f \in \mathcal{L}^2(\mathbb{R})$ on a compact time-frequency domain, we can still search for the eigenvectors h_k of the operator, on the time region $[-T, T]$ we have

$$L_{T,W} h_k = \lambda_k h_k \,, \tag{1.37}$$

The equality is true only in the time region $[-T, T]$, the eigenvectors h_k do not have a compact time support. The eigenvalue λ_k is the ratio of energy remaining after the projection by $L_{T,W}$.

Unfortunately, in this case, no analytic formula are available for h_k and λ_k. In the next two sections, the problem is reformulated by defining new operators acting

on different shapes of the time-frequency space.

1.4.2 Analysis in a time-frequency disk

In [32], a new operator P_S is constructed to project signals onto a spherical region in the time-frequency domain $S = \{(u, w) \mid u_0 + u_0 \leq R^2\}$ and through the use of a basis $\{\phi_{u,w} : t \to e^{iwt} g(t-u)\}$ with g a gaussian envelop $g : t \to 2\pi^{-1/2} e^{-t^2/2}$.

The projection of a signal f by P_S results in the operation

$$P_S f(t) = \int_{(u,w)\in S} \langle \phi_{u,w}, f \rangle \phi_{u,w}(t) dw du \,. \tag{1.38}$$

The resolution of identity, i.e. $P_S f = f$, is satisfied when $S = \mathbb{R}^2$. For bounded sets S, if we take ϕ_{u^*,w^*} with $(u^*, w^*) \notin S$ then $\langle \phi_{u^*,w^*}, P_S f \rangle$ does not totally vanish and decreases rapidly as (w^*, u^*) moves away from S. In comparison with the previous approach, we perform a *soft* slicing of the time-frequency space. Most of the energy of $P_S f$ is concentrated in S but a small amount of energy also leaks outside S. On Fig. 1.3, we show the time-frequency region of projection.

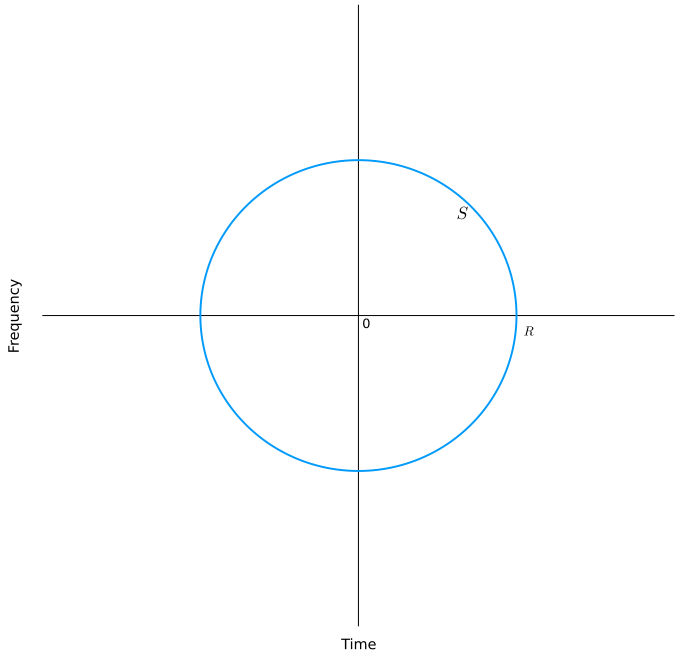


Figure 1.3: Illustration of the projection region of operator P_S.

We can check that this operator is positive definite, self-adjoint and bounded. The eigenvectors and eigenvalues of the operator indicate what types of signals "fit" inside S and how concentrated they are. In [32], the authors show that the

eigenvalues are given by

$$\lambda_k = \frac{1}{k!}\int_0^{R^2/2} s^k e^{-s} ds = \frac{1}{k!}\gamma(k+1, R^2/2)\,, \tag{1.39}$$

where $\gamma(k,a) = \int_0^a x^{k-1}e^{-x}dx$ is the incomplete gamma function. The eigenvalues are ordered by decreasing values, we have $\lambda_k > \lambda_{k+1}, \forall k \in \mathbb{N}$.

The eigenvectors have the following closed-form expression

$$h_k(t) = a_k H_k(t) e^{-t^2/2}\,, \tag{1.40}$$

with a_k a normalization constant, and H_k are the Hermite polynomials given by the formula [1, p. 22.11]:

$$H_k(t) = (-1)^k e^{t^2/2}\frac{d^k}{dt^k}e^{-t^2/2}$$

The eigenvalues measure what remains of the eigenvectors in term of energy after applying P_S. Noting $\rho(S)$ the ratio of energy with the operator restricted to S, we have for $f = h_k$:

$$\rho_{h_k}(S) = \frac{\|P_S h_k\|}{\|h_k\|} = \lambda_k$$

This ratio is maximized when $k = 0$, i.e. when the eigenvector is a Gaussian $h_k = h_0 = a_0 e^{-t^2/2}$.

We presented the case when S is a spherical region centered at the origin. But generalizations are given in [32] for disk regions arbitrarily centered and of elliptical form. The authors show that the eigenvectors have the same form up to a frequency modulation and a time shift of h_k when the region is not centered at the origin, and up to a scaling term in h_k for elliptical regions.

If the same time-frequency disk is positioned at each point of a regular grid of the time-frequency domain, we can extract the first eigenvectors of all the associated projectors to construct the basis of the Gabor transform, i.e. when the window of the STFT is chosen Gaussian. The problem of localizing information in time-frequency regions of disk shape is closely related to the STFT. The next section shows that the problem can again be reformulated so that general forms of wavelet bases appear.

1.4.3 Analysis in a time-scale disk

The previous work has been adapted to time-scale regions in [33] and further detailed in [94] and [72]. In order to define the operator, the authors in [94] start with a two parameter generalization of the Cauchy wavelets with Fourier transform

$$\widehat{\psi}^{+1}_{a,u,\beta,\gamma}(w) = A\,\mathbf{1}_{w>0}|w|^\beta e^{-(a+iu)w|w|^{\gamma-1}},\ a \in \mathbb{R}^+, u \in \mathbb{R}\,, \tag{1.41}$$

with A a normalizing constant such that $\|\widehat{\psi}^{+1}_{a,u,\beta,\gamma}\| = 1$ and with $\gamma \geq 1$,$\beta > (\gamma-1)/2$.

$\widehat{\psi}^{+1}$ is analytic, i.e. it vanishes for $w < 0$. In order to represent signals with positive and negative frequency components, an anti-analytic variant $\widehat{\psi}^{-1}$ is needed, it is defined by $\widehat{\psi}^{-1}(w) = \widehat{\psi}^{+1}(-w)$ for $w < 0$.

The parameters a and b are respectively the scaling and translation parameters. It is demonstrated in [94] that the resolution of identity holds for this choice and fixed β, γ.

Let P_S the operator that projects signals on a set $\{\psi_{a,u,\beta,\gamma}\}_{a,u\in S}$ with

$$S = \left\{(a,u) \in \mathbb{R}^+ \times \mathbb{R} \;\middle|\; (a-C)^2 + u^2 \leq C^2 - 1\right\},$$

which is a disk of radius $C^2 - 1$ centered at $(C, 0)$ with $C > 1$. The action of the operator P_S on f is

$$P_S f(t) = \sum_{\epsilon=-1 \text{ or } +1} \int_{a,u\in S} \psi^{\epsilon}_{a,u,\beta,\gamma}(t) \langle \psi^{\epsilon}_{a,u,\beta,\gamma}, f\rangle \frac{da}{a^2} du\,.$$

A time-frequency region $D_{\beta,\gamma}$ can be associated to P_S by looking at the time and frequency moments of $\psi^{\epsilon}_{a,u,\beta,\gamma}$. Noting them u and w, we have [33]:

$$b = \int t |\psi^{\epsilon}_{a,u,\beta,\gamma}(t)|^2 dt = C_2 u a^{1/\gamma - 1} \tag{1.42}$$

$$w = \int \nu |\widehat{\psi}^{\epsilon}_{a,u,\beta,\gamma}(\nu)|^2 d\nu = \text{sign}(\epsilon) C_1 a^{-1/\gamma} \tag{1.43}$$

with C_1, C_2 two constants depending on β, γ. The associated time-frequency region $D_{\beta,\gamma}$ is given by

$$D_{\beta,\gamma} = \left\{ (b,w) \in \mathbb{R}^2 \;\middle|\; \left(\left(\frac{C_1}{|w|}\right)^{\gamma} - C\right)^2 + \left(\frac{b}{C_2}\right)^2 \left(\frac{C_1}{|w|}\right)^{2\gamma-2} \leq C^2 - 1 \right\}$$

The operator projects the signal on a set of wavelets which have their time and frequency moments in the region $D_{\beta,\gamma}$. This region gets a particular shape that can be changed by varying the parameters β, γ and extended by increasing the radius C. We illustrate the time-scale region and its mapping in the time-frequency domain on Fig. 1.4.

The authors in [94] show that the eigenvalues of this operator are given by:

$$\lambda_{k,\beta,\gamma} = \mathcal{B}\left((C-1)/(C+1), k, r-1\right) \tag{1.44}$$

with $r = (2\beta+1)/\gamma$ and $\mathcal{B}(y,a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^y x^{a-1}(1-x)^{b-1} dx$ the incomplete Beta function.

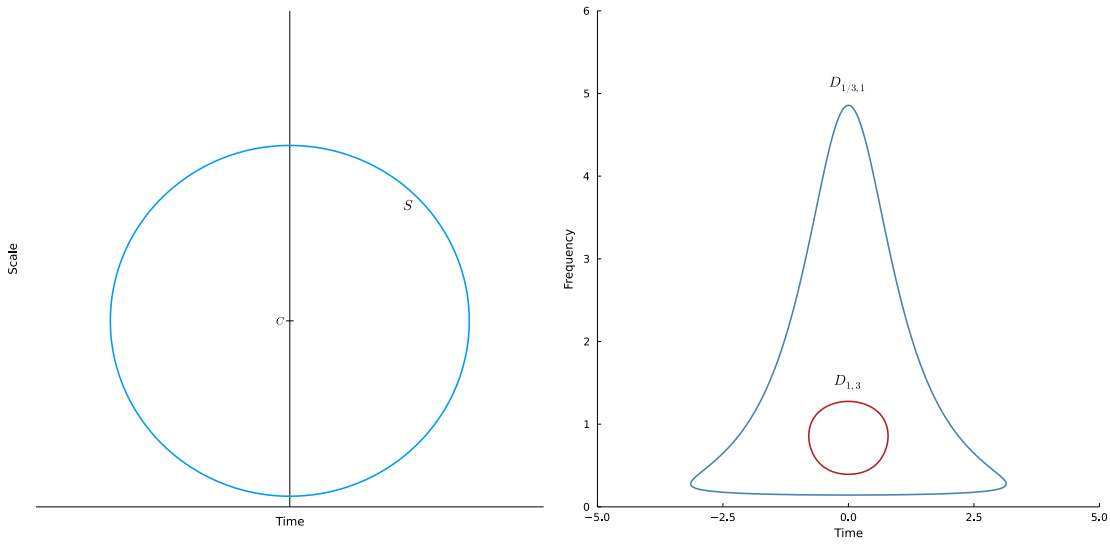


Figure 1.4: On the left the time-scale region S, on the right the mapping in the time-frequency space for parameters $(\beta = 1/3, \gamma = 1)$ and $(\beta = 1, \gamma = 3)$. For the same time-scale disk of fixed radius C, the time-frequency regions covered by the operators change drastically with their parameters. In particular, the $D_{1,3}$ region is much more concentrated.

The eigenvalues have a multiplicity of two; the expression of the eigenvectors are given in the Fourier domain [94, 33]. We have for $w > 0$:

$$h^{+1}_{k;\beta,\gamma}(w) = n_{k;\beta,\gamma} w^{\beta} e^{-w^{\gamma}} L^{c}_{k}(2w^{\gamma}) \tag{1.45}$$

For $w < 0$:

$$h^{-1}_{k;\beta,\gamma}(w) = n_{k;\beta,\gamma} |w|^{\beta} e^{-|w|^{\gamma}} L^{c}_{k}(2|w|^{\gamma}) \tag{1.46}$$

where $n_{k;\beta,\gamma} = \sqrt{\frac{\pi\gamma 2^{r+1}\Gamma(k+1)}{\Gamma(k+r)}}$, $c = r - 1$ and L^{c}_{k} are the generalized Laguerre Polynomials:

$$L^{c}_{k}(x) = \sum_{i=0}^{k} (-1)^{i} \frac{\Gamma(k+c+1)}{\Gamma(k-i+1)\Gamma(c+k+1)} \frac{x^{i}}{i!} \tag{1.47}$$

Unfortunately, for general β, γ no closed-form expressions exist in the temporal domain.

As previously, if we look at the ratio of energy remaining after the projection, the first-order eigenvectors for $k = 0$ maximizes it, i.e. for $h^{+1}_{0;\beta,\gamma}$ and $h^{-1}_{0,\beta,\gamma}$.

1.4.4 Analytic Wavelet Transforms with Generalized Morse Wavelets

In this thesis, we will generally compute wavelet transform with the Generalized Morse Wavelets (GMW) of previous sections. We will use the first-order and ana-

lytical eigenvector $h^{+1}_{0;\beta,\gamma}$ of the projecting operator of Sec. 1.4.3 to define a mother wavelet and its offsprings. We fix the parameters $\beta \in \mathbb{R}^+, \gamma \in \mathbb{R}^+$ and note ψ the mother wavelet, and we have for $w > 0$:

$$\widehat{\psi}_{\beta,\gamma}(w) = c_{\beta,\gamma} w^{\beta} e^{-w^{\gamma}} . \tag{1.48}$$

with $c^2_{\beta,\gamma} = \pi\gamma \; 2^r/\Gamma(r)$ where $r = (2\beta+1)/\gamma$. The mother wavelet is analytic and vanishes for $w \leq 0$.

Note that as in [73], we extended the domain of validity of the (β, γ) parameters of the Generalized Morse Wavelet to $\beta > 0$ and $\gamma > 0$, instead of $\gamma \geq 1$,$\beta > (\gamma-1)/2$ in the original works of [94, 33]. For $\beta > 0, \gamma > 0$, they are called Generalized Morse Filters in [73], in our case we will keep the name Generalized Morse Wavelets for simplicity.

The basis is naturally obtained by shifting and scaling the mother wavelet with parameters $a \in \mathbb{R}^+$ and $u \in \mathbb{R}$. Their expression in frequency is given by:

$$\widehat{\psi}_{a,u,\beta,\gamma}(w) = \sqrt{a} c_{\beta,\gamma} (aw)^{\beta} e^{-(aw)^{\gamma}} e^{-iwu} . \tag{1.49}$$

Frequency peak and duration. It is useful to associate the scale of a wavelet to a frequency. There are various ways to do so, see in particular [72, Sec. II.D]. In our case, we are interested in the *frequency peak* w_ψ of the wavelet, it is given by:

$$w_{\psi_{a;\beta,\gamma}} = \arg\max_w |\widehat{\psi}_{a,u,\beta,\gamma}(w)| = \frac{1}{a}\left(\frac{\beta}{\gamma}\right)^{1/\gamma} \tag{1.50}$$

and can be found by setting to zero the first derivative of (1.49). This quantity is helpful to position the maximum of amplitude of the wavelet at some frequency bands.

We are also interested in the spread of the wavelet in frequency to more or less cover a band of frequency. For that, we can compute the *duration* $d_{\beta,\gamma}$ of the wavelet. With w_ψ and $\widehat{\psi}$ the short notations for the frequency peak and the GMW considered, the *duration* is computed using [72]:

$$d_{\beta,\gamma} = \left| \frac{w^2_\psi}{\widehat{\psi}(w_\psi)} \frac{\partial^2 \widehat{\psi}}{\partial w^2}\bigg|_{w=w_\psi} \right| = \sqrt{\beta\gamma} . \tag{1.51}$$

With increasing values of $d_{\beta,\gamma}$ the wavelet shrinks in frequency and broadens in time.

These two quantities are helpful in practice to get an idea of how the wavelet behaves by varying β and γ. Other quantities are given in [72] to characterize the asymmetry and the shape around the frequency peak, they are related to skewness and kurtosis measures in statistics.

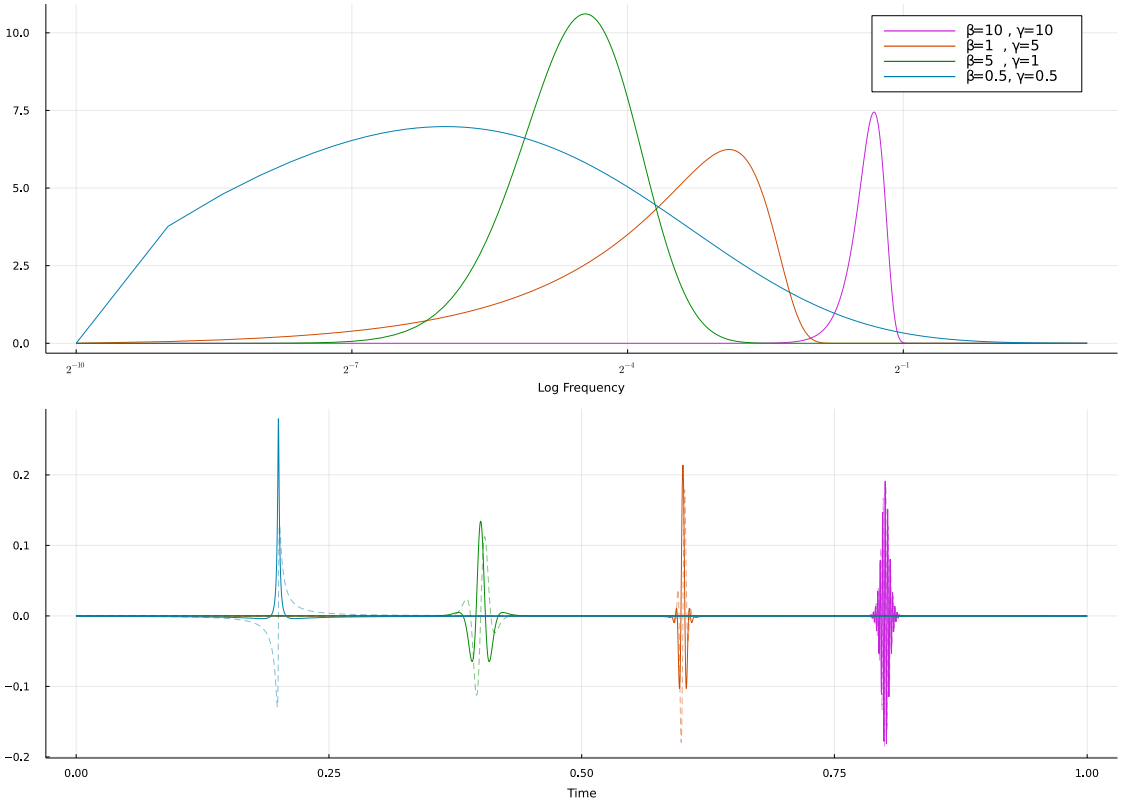


Figure 1.5: Some examples of Generalized Morse Wavelets with varying parameters β and γ.

We plot on Fig. 1.5 examples of first-order Generalized Morse Wavelets with varying parameters β and γ. We note that by varying the parameters (β, γ) the peak of frequency, the duration and the asymmetry around the peak change.

We will use this family for the estimation of a frame of wavelets in Sec. 4.3 and for modeling patterns in side-channel signals in Chap. 5.

Chapter 2

Elements of probability and information theory

The side-channel analysis field could not come up with attack methods without heavily relying on probability and information theory. As many other engineering domains, we have to deal with the unpredictable aspect of the physical world and its processes. It is successful at building stochastic models of the world and measuring their fitness to real data. The engineering community can rely on this framework to work on a common ground and compare their models. We introduce in this chapter the main tools of probability and information theory that are used in the field of machine learning and side-channel analysis.

We start with some basics and introduce our notations, then we move on to information theoretic tools which are of importance in the security domain. Next, we recall the maximum likelihood and a posteriori estimators. We then recapitulate some distribution laws used in this manuscript and present in particular the Gaussian Mixture Model. We also present the Expectation Maximization algorithm for learning in an unsupervised manner a GMM. Finally, we conclude on the Metropolis Hastings algorithm for sampling random variables.

2.1 Notations and basics

We note uppercase a random variable X that gets values x in a space $\boldsymbol{X}$. A measurable space is noted $(\boldsymbol{X}, \mathcal{X})$ with $\mathcal{X}$ a σ-algebra, we note P_X the probability measure of X on $\mathcal{X}$ and p_X its density with respect to an underlying measure μ (e.g. the Lebesgue measure on $\mathbb{R}^n$ or a counting measure for $\boldsymbol{X}$ countable). For a stochastic process X, we note $X(t)$ the random variable at time t and $x(t)$ the observed values $\forall t \in \mathbb{R}$.

To lighten the notations and if it is clear from the context, the underscript

for probability measure and densities will sometimes be dropped, and we will note $p_X(X = x) = p(x)$.

For X, Y two random variables, the joint and conditional probability densities of X and Y are noted $p_{X,Y}(X = x, Y = y) = p(x, y)$ and $p_{X|Y}(X = x|Y = y) = p(x|y)$ respectively.

Let f an arbitrary function from $\boldsymbol{X}$ to $\boldsymbol{Y}$ and X a random variable, the expectation of the random variable $Y = f(X)$ according to the probability measure of X is given in the discrete case by:

$$\mathbb{E}_X\left[f(X)\right] = \sum_{x \in \boldsymbol{X}} p(x) f(x) \tag{2.1}$$

In the continuous case, for f a measurable function, we have:

$$\mathbb{E}_X\left[f(X)\right] = \int_{\boldsymbol{X}} f(x) p(x) dx \tag{2.2}$$

In our case, we will always assume that $\boldsymbol{X} \subset \mathbb{R}^n$ for some $n \in \mathbb{N}$ and that dx indicates the use of the Lebesgue measure on $\mathbb{R}^n$ as a base measure to compute the integral.

In the discrete case, we have the following relation between the probability densities of Y and X with a function $f : \boldsymbol{X} \mapsto \boldsymbol{Y}$:

$$p(y) = \sum_{x \in \boldsymbol{X}} \mathbf{1}_{x \in f^{-1}(y)} p(x) \tag{2.3}$$

Finally, we have the Bayes rule for two random variables X, Y:

$$p(x|y) = \frac{p(x, y)}{p(y)} \tag{2.4}$$

$$= \frac{p(y|x)p(x)}{p(y)} \tag{2.5}$$

This formula has a particular interpretation when one variable represents some parameters to estimate and the other the available data. Let X the random variable coming as observations data $\{x_i\}_i$ and θ a variable (possibly random) representing some parameters. Assuming a model for the joint distribution $p(x, \theta)$, $p(\theta)$ is called the *prior* distribution on the parameters, it models our belief about θ and indicates where it is likely to lie in the space Θ. $p(\theta|x)$ is called the *posterior* or *a posteriori* distribution as it quantifies our belief about θ after the acquisition of observations x. Finally, $p(x|\theta)$ measures the *likelihood* of x with supposed θ.

2.2 Elements of Information Theory

2.2.1 Entropy and Mutual Information

For a random variable X taking discrete values in $\boldsymbol{X}$, the Shannon entropy of X is noted H_X and given by:

$$H_X = -\sum_{x \in \boldsymbol{X}} p(x) \log p(x) \tag{2.6}$$

This quantity is positive or null for discrete random variables since $\forall x \in \boldsymbol{X}, p(x) \in [0, 1]$. H_X vanishes when X is completely determined by an observation $x^* \in \boldsymbol{X}$ such that $p(x) = 1$ if $x = x^*$, 0 otherwise. It is maximized when X follows an uniform distribution on $\boldsymbol{X}$, i.e. when $p(x) = 1/|\boldsymbol{X}|$.

For a random variable X taking continuous values in $\boldsymbol{X}$, the differential entropy H_X of X is given by:

$$H_X = -\int_{\boldsymbol{X}} p(x) \log p(x) dx \tag{2.7}$$

It is not necessarily positive and can take values in $\mathbb{R}$.

We have the following relation for the entropy of a pair of random variables $(X, Y) \in \boldsymbol{X} \times \boldsymbol{Y}$, expressed in the continuous case here without loss of generality:

$$\begin{aligned} H_{X,Y} &= -\int_{\boldsymbol{X}} \int_{\boldsymbol{Y}} p(x, y) \log p(x, y) dx dy && (2.8) \\ &= -\int_{\boldsymbol{X}} p(x) \log p(x) dx - \int_{\boldsymbol{X}} \int_{\boldsymbol{Y}} p(x, y) \log p(y|x) dx dy && (2.9) \\ &= H_X + H_{Y|X} && (2.10) \end{aligned}$$

Where $H_{Y|X}$ is the conditional entropy of Y given X.

The conditional entropy of two random variables is of importance since it measures how the knowledge of one variable can influence the entropy of the other. If the knowledge of one variable decreases the entropy of the other, then intuitively we can assume that both variables are linked in some ways and that it could be possible to retrieve information about the unknown variable given the observed one. In consequence, we are interested in the difference:

$$\begin{aligned} H_Y - H_{Y|X} &= H_{Y,X} - H_{X|Y} - H_{Y|X} && (2.11) \\ &= H_X - H_{X|Y} && (2.12) \\ &= I_{X,Y} && (2.13) \end{aligned}$$

Where $I_{X,Y}$ is the mutual information given by:

$$I_{X,Y} = \int_{\boldsymbol{X}} \int_{\boldsymbol{Y}} p(x,y) \log \left[\frac{p(x,y)}{p(x)p(y)} \right] dxdy \tag{2.14}$$

We intuitively guess this quantity positive or null, since if H_X gives a measure of the unpredictability of a random variable X then the knowledge of Y can't increase its entropy, thus $H_X - H_{X|Y}$ stays positive or null.

Actually, we will show in next section that the mutual information can be cast as a measure of dissimilarity between the joint distribution $p_{X,Y}$ and the product of marginal distributions $p_X p_Y$. It gives another interpretation of $I_{X,Y}$: if X and Y are independent then $p_{X,Y} = p_X p_Y$ and the dissimilarity between both distribution vanishes.

2.2.2 Cross-entropy and Divergence between distributions

Given a set of observations $\{x_i\}_i$, it is natural to think of an idealized system that produced them according to a probability distribution p_X. Since this mathematical function p_X is never known, we can propose an approximate model with distribution q. To measure how well q matches the target distribution p_X, we can use the Kullback-Leibler divergence which is based on the Shannon's entropy of Sec. 2.2.1.

For two probability distributions p, q on a space $\boldsymbol{X}$, the Kullback-Leibler divergence is given by:

$$D_{KL}\left[p \| q\right] = \int_{\boldsymbol{X}} p(x) \log \left[\frac{p(x)}{q(x)} \right] dx \tag{2.15}$$

It is used as a measure of the dissimilarity between two distributions. It is positive and not symmetric, i.e. $D_{KL}\left[p \| q\right] \neq D_{KL}\left[q \| p\right]$ in general.

It is sometimes useful to decompose this divergence into two terms:

$$\begin{aligned} D_{KL}\left[p \| q\right] &= \int_{\boldsymbol{X}} p(x) \log p(x) dx - \int_{\boldsymbol{X}} p(x) \log q(x) dx & (2.16) \\ &= -H(p) + C(p,q) & (2.17) \end{aligned}$$

where $C(p,q) = -\int_{\boldsymbol{X}} p(x) \log q(x) dx$ is called the cross-entropy between p and q.

The cross-entropy is not positive but can be used as an alternative way to measure the dissimilarity between p and q when the target distribution to adjust is q. This function is commonly used in machine learning as a loss function to fit statistical models. Suppose that q is a parametrized distribution, i.e. that there exists a space of parameters Θ such that $\{q_\theta(X|\theta)\}_{\theta \in \Theta}$ is a set of proposal distributions, if we want to find the closest distribution q_{θ^*} to p according to $D_{KL}\left[p \| q\right]$ then we can optimize $C(p,q)$ instead of $D_{KL}\left[p \| q\right]$ since $H(p)$ is constant.

The alternative version $D_{KL}[q\|p]$ can also be used as an optimization criterion for q. In that case it is necessary to perform the optimization on $D_{KL}[q_\theta\|p]$ and not solely on the cross-entropy. It is used for example in variational inference methods and for training variational encoders in the field of deep learning.

This divergence presents particular properties when the distributions p, q belong to the exponential family, i.e. when probability distributions can be written $p(x) = \frac{1}{A(\theta)} e^{\langle \theta, h(x) \rangle}$ with θ the distribution parameters, $A(\theta)$ a normalization factor and h a function of x also called the sufficient statistics of X. In these conditions it is possible to derive optimal gradients that will take into account the metric induced by the divergence on the space of parameters Θ [4, 118, 52].

In the literature on side-channel analysis, the cross-entropy criterion has been used to train neural networks in a supervised manner to classify signals according to cryptographic keys, see for example [85]. The cross-entropy is also encountered in the Expectation Maximization (EM) algorithm of Sec. 2.6 that we envisage to use for learning the parameters of the generative model of Chap. 5.

2.3 Maximum likelihood estimator

Let X a random variable and $\{x \to p(x|\theta)\}_{\theta \in \Theta}$ a set of parametrized distributions. Given a set of N observations $\{x_i\}_{1 \leq i \leq N}$, the maximum likelihood estimator θ_{MLE} is the best parameters in Θ maximizing the likelihood:

$$\theta_{\text{MLE}} = \arg\max_{\theta \in \Theta} \prod_{i=1}^{N} p(x_i|\theta) \tag{2.18}$$

$$= \arg\max_{\theta \in \Theta} \sum_{i=1}^{N} \log p(x_i|\theta) \tag{2.19}$$

The second form is usually preferred for computational reason and feasibility. We note $\mathcal{L}_{\text{MLE}}$ the log-likelihood loss and write for a batch of data $\{x_i\}_{1 \leq i \leq N}$:

$$\mathcal{L}_{\text{MLE}}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \log p(x_i|\theta) \tag{2.20}$$

We can think of the maximum of likelihood as the minimization of the Kullback-Leibler divergence between an empirical distribution

$$\tilde{p}(x) = \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i)\,, \tag{2.21}$$

and a parametrized distribution p_θ, see [4, Sec. 2.8.3], we can write:

$$\mathcal{L}_{\text{MLE}}(\theta) = C(\tilde{p}, p_\theta) \tag{2.22}$$

$$= D_{KL}\left[\tilde{p} \| p_\theta\right] + H(\tilde{p}) \tag{2.23}$$

using (2.17), thus we have:

$$\arg\min_{\theta\in\Theta} D_{KL}\left[\tilde{p}\|p_\theta\right] = \arg\min_{\theta\in\Theta} \mathcal{L}_{\text{MLE}}(\theta)\,. \tag{2.24}$$

The maximum likelihood estimation method is used in Sec. 4.3 for estimating a frame of wavelet adapted to patterns in side-channel signals.

2.4 Some Distribution Laws

In this section we list some distributions laws that will be used throughout this manuscript. These distributions laws will be used in particular in the Chap. 5 for initializing the parameters of a generative model for side-channel signals.

Multivariate Gaussian Distribution. Let X a random variable in $\mathbb{R}^n$. We note $X \sim \mathcal{N}(\mu, \Sigma)$ to state that X follows a multivariate Gaussian distribution with mean $\mu \in \mathbb{R}^n$ and covariances $\Sigma \in \mathbb{R}^{n\times n}$. It is equivalent to use the following expression to evaluate the probability density at $x \in \mathbb{R}^n$:

$$p(x) = \frac{1}{(2\pi)^{n/2}\det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{2.25}$$

Instead of writing the density $p(x|\mu,\Sigma)$ we sometimes employ $\mathcal{N}(x|\mu,\Sigma)$ to recall the use of a multivariate Gaussian model for X in the equations.

Complex Multivariate Gaussian Distribution. Let X a random variable in $\mathbb{C}^n$ distributed according to a complex Gaussian distribution. We use $X \sim \mathcal{CN}(\mu, \Sigma)$ to when X follows a complex Gaussian distribution with mean $\mu \in \mathbb{C}^n$ and covariance $\Sigma \in \mathbb{C}^{n\times n}$. It has the following density $\forall x \in \mathbb{C}^n$:

$$p(x) = \frac{1}{(\pi)^{n/2}\det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\dagger\Sigma^{-1}(x-\mu)\right) \tag{2.26}$$

Depending on the applications other forms exist for this density, we used the convention of [46].

We do not introduce the univariate case for the Gaussian distribution as it can naturally be deduced by reducing to 1 the dimension of the distributions parameters μ, Σ in the previous identities.

Gamma Distribution. Let X a random variable on $\mathbb{R}^+$, we note $X \sim \text{Gamma}(\alpha, \beta)$ when X follows a gamma distribution with shape $\alpha > 0$ and rate $\beta > 0$. The density is expressed:

$$p(x) = \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x) \tag{2.27}$$

where $\Gamma(a) = \int_0^\infty x^{a-1} e^{-x} dx$ is the gamma function.

This distribution is useful to model the distribution of a random variable on $\mathbb{R}^+$. For example, the variance of univariate Gaussian random variable can be modeled using Gamma distributions.

We saw a generalized form of this expression in the Sec.1.4.3 on Generalized Morse Wavelets. The generalized form is known as the Generalized Gamma distribution [111] and enjoys an extra parameter that allows a better control on the moments of the distribution.

Wishart Distribution. Let X a random matrix variable on the set of positive definite matrices $\mathcal{S}_n^+ \subset \mathbb{R}^{n \times n}$. We note $X \sim \mathcal{W}(m, \Sigma)$ when X follows a Wishart distribution with degree of freedom m and mean $m\Sigma$. The distribution is defined on $\mathcal{S}_n^+$ when $m > n - 1$. Its density has the following expression $\forall x \in \mathcal{S}_n^+$[92]:

$$p(x) = \frac{1}{2^{mn/2} \Gamma_m(n/2) \det(\Sigma)^{n/2}} \det(x)^{\frac{n-m-1}{2}} \exp(-\frac{1}{2} \operatorname{tr}\left(\Sigma^{-1} x\right)) \tag{2.28}$$

with $\Gamma_m(x)$ the multivariate gamma function.

This distribution can be used to draw covariances to initialize the parameters of multivariate Gaussian models. This distribution has been extended to the set of positive semidefinite matrices $\mathcal{S}_{n,m}^+$ of rank $m < n$ in [115].

Exponential Distribution. As a particular case of the Gamma distribution, X follows an exponential distribution when $X \sim \text{Gamma}(0, \beta)$ with $\beta > 0$. In this case, we note $X \sim \text{Exp}(\beta)$ and β is still called the rate of the distribution.

Dirichlet Distribution. We use $X \sim \text{Dir}(\alpha)$ to indicate that the random variable X on the n-simplex

$$\mathcal{C}^n = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid \forall i,\, x_i \geq 0,\, \sum_{i=1}^n x_i = 1\}$$

follows a Dirichlet distribution with concentration vector $\alpha \in \mathbb{R}^n$, $\forall k\, \alpha_k > 0$. Its density is given by:

$$\forall x \in \mathcal{C}^n,\, p(x) = \frac{1}{B(\alpha)} \prod_{i=1}^n x_i^{\alpha_i - 1}, \tag{2.29}$$

where $B(\alpha)$ is the multivariate Beta function:

$$B(\alpha) = \frac{\prod_{i=1}^{n} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{n} \alpha_i)}\,.$$

This distribution is typically used to generate probability vectors $x \in \mathcal{C}^n$.

Categorical Distribution. The random variable $X \in [1, \ldots, n]$ follows a categorical distribution with probability vector $w \in \mathcal{C}^n$ if:

$$p(x) = \prod_{i=1}^{n} w_i^{\mathbf{1}_{x=i}}\,, \tag{2.30}$$

or equivalently

$$p(x = i) = w_i\,, \tag{2.31}$$

and we note $X \sim \mathrm{Cat}(w)$.

2.5 Gaussian Mixture Model

Let X a random variable in $\mathbb{R}^n$, we note $X \sim \mathrm{GMM}(w, \{\mu_i\}_{i=1}^K, \{\Sigma_i\}_{i=1}^K)$ when X follows a Gaussian mixture model. The probability density of X is the weighted sum of K Gaussian densities with parameters $\{\mu_i \in \mathbb{R}^n, \Sigma_i \in \mathbb{R}^{n\times n}\}_{i=1}^K$ and weight vector $w \in \mathcal{C}^K$. The overall density is expressed:

$$p(x) = \sum_{i=1}^{K} w_i\, \mathcal{N}(x|\mu_i, \Sigma_i)\,. \tag{2.32}$$

On top of X, a latent state vector $O \in [1, \ldots, K]$ such that $O \sim \mathrm{Cat}(w)$ with $w \in \mathcal{C}^K$ can be instantiated to control which Gaussian density is activated.

The joint distribution of (X, O) is given by:

$$p(x, o) = \prod_{i=1}^{K} w_i^{\mathbf{1}_{o=i}} \mathcal{N}(x|\mu_i, \Sigma_i)^{\mathbf{1}_{o=i}}\,, \tag{2.33}$$

or equivalently

$$p(x, o = i) = w_i\, \mathcal{N}(x|\mu_i, \Sigma_i)\,. \tag{2.34}$$

Given an observation x, the likelihood that o is responsible for the observation x evaluates to:

$$p(x|o = i) = \mathcal{N}(x|\mu_i, \Sigma_i)\,, \tag{2.35}$$

while the posterior distribution is:

$$p(o = i|x) = \frac{w_i \mathcal{N}(x|\mu_i, \Sigma_i)}{\sum_{i=1}^{K} w_i \mathcal{N}(x|\mu_i, \Sigma_i)} \,. \tag{2.36}$$

This model can be used to approximate arbitrary complex distribution with increasing K or in the context of classification by considering the state vector O as the label for the random variable X.

2.5.1 Quadratic Discriminant Analysis

In the context of supervised learning where the latent variable $O \in [1, \dots, K]$ is considered as the label for a random variable $X \in \mathbb{R}^n$, a Gaussian mixture model can be fitted to a labeled dataset $\{x_i, o_i\}_{i=1}^N$. It amounts to find the parameters of the model, i.e. the mean and covariances $\{\mu_j \in \mathbb{R}^n, \Sigma_j \in \mathbb{R}^{n\times n}\}_{j=1}^K$ of each Gaussian densities. We note these parameters θ.[1] To this end, we employ the principle of maximum likelihood of Sec. 2.3 to find an estimate of θ with the likelihood $p(x, o|\theta)$. The MLE estimator for θ is given by [49, Sec. 4.3]:

$$\theta_{\text{MLE}} = \arg\min_{\theta} - \sum_{i=1}^{N} \log p(x_i, o_i|\theta) \tag{2.37}$$

$$= \arg\min_{\theta} \mathcal{L}_{\text{MLE}}(\theta) \tag{2.38}$$

$$\tag{2.39}$$

where

$$\mathcal{L}_{\text{MLE}}(\theta) = -\sum_{i=1}^{N} \log \left[\prod_{j=1}^{K} w_j^{\mathbf{1}_{o_i=j}} \mathcal{N}(x_i|\mu_j, \Sigma_j)^{\mathbf{1}_{o_i=j}} \right] \tag{2.40}$$

$$= -\sum_{i=1}^{N} \sum_{j=1}^{K} \mathbf{1}_{o_i=j} \log \mathcal{N}(x_i|\mu_j, \Sigma_j) + c \tag{2.41}$$

$$= -\sum_{j=1}^{K} \sum_{(x_i,o_i)|\, o_i=j} \log \mathcal{N}(x_i|\mu_j, \Sigma_j) + c \tag{2.42}$$

$$= \sum_{j=1}^{K} \mathcal{L}_j(\mu_j, \Sigma_j) + c \tag{2.43}$$

[1] To simplify here, we suppose that the probability weights $\{w_i = p(o = i)\}$ are known or directly estimated from the training dataset.

where c is a constant capturing the empirical estimate of the entropy of O. We introduced subsidiary losses $\mathcal{L}_j$ with expression

$$\mathcal{L}_j = - \sum_{(x_i,o_i)\,|\,o_i=j} \log \mathcal{N}(x_i|\mu_j, \Sigma_j)\,. \tag{2.44}$$

We see that the initial loss $\mathcal{L}_{\text{MLE}}$ is divided into K losses $\mathcal{L}_j$ which are function of the parameters of each individual Gaussian distributions. We can show by studying the gradient of $\mathcal{L}_j(\mu_j, \Sigma_j)$ according to μ_j and Σ_j that a minimum is reached for the empirical mean:

$$\tilde{\mu}_j = \frac{1}{N_j} \sum_{(x_i,o_i=j)} x_i$$

and for the covariances

$$\tilde{\Sigma}_j = \frac{1}{N_j} \sum_{(x_i,o_i=j)} (x_i - \tilde{\mu}_j)(x_i - \tilde{\mu}_j)^\dagger$$

where we noted $N_j = \sum_{(x_i,o_i)} \mathbf{1}_{o_i=j}$.

In conclusion, the MLE estimator is reached for $\theta_{\text{MLE}} = \{\tilde{\mu}_j, \tilde{\Sigma}_j\}_{j=1}^K$.

In side-channel analysis, this method is used in template attacks [21] to learn the parameters of Gaussian Mixture Model and classify signals according to cryptographic keys.

In the next section, we introduce a method that can be used to fit a Gaussian mixture model in the unsupervised case where the labels $\{o_i\}_i$ are unknown.

2.6 Expectation Maximization

Let X a random variable, from which we get observations in practice. We suppose a hidden latent structure onto X represented by a random variable O, such that the probability density of X has the following expression:

$$p(x) = \int_{\boldsymbol{O}} p(x, o) do\,, \tag{2.45}$$

here in the continuous case and without loss of generality for the discrete case.

The latent variable O is characterized as hidden since it is not directly observed in practice. The latent structure can for example be assumed of lower dimensionality such as to compress the information about X into a latent variable O.

This requires a model for $p(x, o)$ that will be chosen among a family of parametrized distribution $\{p(x, o|\theta)\}_{\theta\in\Theta}$. Given a dataset of observations $\{x_i\}_{i=1}^N$, we want to min-

imize

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log p(x_i|\theta)\,, \tag{2.46}$$

according to θ. In some cases an analytical expression for $p(x|\theta)$ is not accessible or the optimization of $\mathcal{L}(\theta)$ according to θ is not possible. In these cases, we can take into account the latent model on X and use (2.45) such that (2.46) can be rewritten:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log \int_{\boldsymbol{O}} p(x_i, o|\theta) do \tag{2.47}$$

Starting from this equation, an upper bound depending on θ is found for the loss:

$$\begin{aligned}
\mathcal{L}(\theta) &= \mathbb{E}_{\tilde{p}_X}\left[-\log p_{X|\theta}\right] && (2.48)\\
&= \mathbb{E}_{\tilde{p}_X}\left[-\log \mathbb{E}_{p_{O|X,\theta}}\left[p_{X|\theta}\right]\right] && (2.49)\\
&= \mathbb{E}_{\tilde{p}_X}\left[-\log \mathbb{E}_{p_{O|X,\theta_0}}\left[\frac{p_{X,O|\theta}}{p_{O|X,\theta_0}}\right]\right] && (2.50)\\
&\leq \mathbb{E}_{\tilde{p}_X}\left[\mathbb{E}_{p_{O|X,\theta_0}}\left[-\log\left[\frac{p_{X,O|\theta}}{p_{O|X,\theta_0}}\right]\right]\right] && (2.51)
\end{aligned}$$

where we used the Jensen inequality in (2.50)

$$\begin{aligned}
&= C(\tilde{p}_X p_{O|X,\theta_0}, p_{X,O|\theta}) + c && (2.52)\\
&= \tilde{\mathcal{L}}(\theta_0, \theta) + c\,. && (2.53)
\end{aligned}$$

We introduced in (2.50) a referential distribution for $p(O|X,\theta_0)$ with parameters θ_0. It serves as a referential point on the set of parametrized distribution to drive an iterative optimization algorithm. The constant c absorbs the expectation with parameter θ_0, i.e. $\mathbb{E}_{p_{O|X,\theta_0}}\left[\log p_{O|X,\theta_0}\right]$.

In the literature, the Expectation-Maximization algorithm [36] iteratively maximizes the negative loss -$\tilde{\mathcal{L}}(\theta_0, \theta)$. At step t, $\tilde{\mathcal{L}}(\theta_t, \theta)$ is computed with the previous estimate θ_t, this is the Expectation step (E-step), then the maximum is searched according to θ, called the Maximization step (M-step). The parameters are updated such that at $t+1$, $\theta_{t+1} = \theta^*$ with θ^* the optimal parameters found. The procedure is repeated until convergence, a proof of convergence is shown in [36]. A generalisation of this approach by the introduction of distributions of a different family of the true posterior $p_{O|X,\theta_0}$ is termed variational inference and is presented for example in [118, 105].

The expectation maximization algorithm is proposed as a learning algorithm in the learning strategy of Sec. 5.6.

2.7 Metropolis Hastings

Markov Chain Monte Carlo methods can be used to draw samples from probability distributions. Among these methods, the Metropolis-Hastings method [77, p. 365] uses a proposal distribution q to draw a sequence of samples $\{x_t\}_t$ such that for t sufficiently large, x_t can be assumed drawn from the true distribution p_X, which might be known up to a normalisation factor. The procedure goes as follows:

1. Start with an initial random sample x_0
2. At $t+1$, draw a new sample x^* from the proposal distribution $q(x|x_t)$ conditioned on the previous sample x_t
   - Compute
   $$r = \min\left[1, \frac{p(x^*)q(x^*|x_t)}{p(x_t)q(x_t|x^*)}\right]$$
   - Update $x_{t+1} = x^*$ with probability r
   - otherwise, $x_{t+1} = x_t$.

   This step is repeated until a maximum number of iterations is reached.

The advantage of such method is that the proposal distribution for sampling can be chosen simple. Often, the proposal distribution for $q(x^*|x_t)$ is a Gaussian distribution centered at x_t where the variance (or covariance in higher dimension) is properly adapted.

This method will be of particular importance to estimate the times of occurrence of operations in the generative model of Sec. 5.7.

Chapter 3

Side Channel Analysis

According to David Kahn in its book *The Codebreakers* the first recorded use of cryptography dates back to the Ancient Egypt around 1900 B.C. Afterwards, many civilizations, such as the Mesopotamian civilization or Ancient Greece, demonstrated uses of cryptography in military and politics. After a stagnation during the Middle age, the development of cryptology restarted during the Renaissance and rushed during the World Wars. Now, the use of cryptography is pervasive in most communicating devices and has been democratized to the point that anyone can secretly convey messages.

But such a level of sophistication could not have been reached without the concurrent development of cryptanalysis and cryptography. As introduced at the beginning of this manuscript, many different types of attacks have been developed to test the security of cryptosystems. We will focus here on side-channel attacks.

To start with, we will introduce some elements of information security to understand the test objectives of the security of cryptosystems. The side-channel attacks methods presented in this thesis can be used for different electronic devices, but we chose to present smart-cards in this section as they are widely used in cryptographic-related applications and make a good example of devices that can be targeted by attackers. We will summarize some of their characteristics and their functioning. Finally, we will introduce side-channel analysis and discuss on its historical development and recent advances.

3.1 Information Security

3.1.1 Cryptosystem

The secure communication of messages between two parties can be represented by a *cryptosystem*. It is mathematically defined as a 5-tuple $(\boldsymbol{E}, \boldsymbol{Z}, \boldsymbol{K}, \boldsymbol{C}, \boldsymbol{D})$ where $\boldsymbol{E}$ and $\boldsymbol{Z}$ are finite sets of plaintexts and ciphertexts, and $\boldsymbol{K}$ is a set of keys such that each key $k \in \boldsymbol{K}$ uniquely identifies a ciphering rule $c_k \in \boldsymbol{C}$, $c_k : \boldsymbol{E} \to \boldsymbol{Z}$ and a deciphering rule $d_k \in \boldsymbol{D}$, $d_k : \boldsymbol{Z} \to \boldsymbol{E}$ which satisfy for all $e \in \boldsymbol{E}$, $e = d_k(c_k(e))$.

In the literature on cryptography, it is common to consider the following objectives that cryptosystem should address:

- *Confidentiality*: The ciphertext is unreadable by potential eavesdroppers.
- *Data Integrity*: Any alteration of the ciphertext can be detected by the receiver.
- *Authentication*: The ability to verify the identity of a person or an entity.
- *Non-Repudiation*: After enciphering a message, the responsible cannot deny he is the author.

In modern cryptology, a cryptosystem is always assumed publicly known. This principle, attributed to Kerckhoff (1839-1903), implies that the security of communication relies solely on the secrecy of the key and not on the knowledge of the cryptosystem's design. Thus, the security of a cryptosystem depends on its reliability at keeping secret the key against an attacker trying to decipher the communication.

The best way to test the security of a cryptosystem, according to Kerckhoff and some of its predecessors, is to put ourselves in the place of an attacker. It is reminiscent of the "scientific method" which consists in verifying a hypothesis by conducting experiences. Here, the assumption that the cryptosystem is secure needs to be tested by carrying out attacks on the system. If it resists to all known attacks and until an attack is found, the cryptosystem is assumed secure.

There are various ways to characterize the security of a cryptosystem. In our case, we are interested in the *computational security* of a cryptosystem, i.e. the amount of computations that is required to perform an attack. Under this definition, a cryptosystem is computationally secure against an attacker if current technologies do not provide enough resources to carry out the attack. In theory, an exhaustive search of the key, also called a bruteforce attack, is possible given a pair of plaintext and ciphertext. It consists in testing all the keys until a ciphering or deciphering rule makes the pair match.

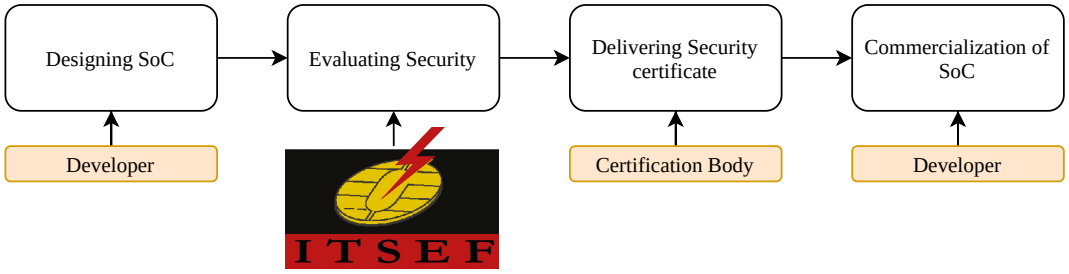


Figure 3.1: Illustration of the certification procedure before commercializing a system on chip (SoC).

The evaluation of the security of cybersecurity products is crucial to this day. Developers and manufacturers of cybersecurity solutions may obtain security visas from third-party bodies to demonstrate the security of their product. For that, many countries agreed on the elaboration of the Common Criteria certification that permits comparability between the results of independent security evaluations. The Common Criteria certification classifies the security of a product according to an increasing scale matching the attack potential of attackers. We can see this as an extension of the concept of *computational security* by considering other factors such as the availability of design documents on products or by the cost and feasibility of attacks. In France, the evaluations are provided by independent laboratories accredited by the French accreditation committee, the certificate is then obtained from the French National Cybersecurity agency. We resumed Fig. 3.1 the certification procedure with the example of a SoC, but the same procedure is applied for each evaluated product (component, smart card, etc.).

To illustrate how much resources are required to perform attacks we consider the two following examples. The Caesar cipher, which replaces each letter of a plaintext with the letter positioned k-steps (the offset number playing the role of the key) before in the alphabet, can be attacked by anyone using pen and paper and by testing the 25 deciphering rules for an alphabet of 26 letters. In comparison, an attack on the 12-round DES cipher, presented in an article [86] from 1993, required knowledge in statistics and cryptology, and 50 hours of computations on a rather powerful computer at that time.

In order to make an exhaustive key search impossible, the cardinality of key space is made very large. The Advance Standard Encryption (AES) cipher can use keys of length 256-bits, for this version an exhaustive key search will require testing $2^{256} \approx 10^{77}$ keys which is practically unfeasible. The computational security of a cryptosystem will be undermined if, in one way or another, the amount of computational work to get the key is noticeably decreased.

A naive evaluator performing an exhaustive key search will assume that the

entropy of the key is maximal, i.e. that keys are equiprobable. Consequently, he will also assume that the computational cost of his attack will be maximal. We understand intuitively that any gathered information that can reduce the entropy of the key is welcome as it transforms the evaluator's distribution on the keys into a distribution with a lower entropy. In the later case, the evaluator would be able to test each key according to its likeliness and consequently reduce in average the cost of a bruteforce attack.

For example, if it is known that French is used to write the plaintexts, then we could theoretically lower the entropy on the keys if we know the ciphertexts. Let $|\boldsymbol{Z}| = |\boldsymbol{E}|$, if $H_E < \log|\boldsymbol{E}|$, it is shown that[113, Th. 2.10]:

$$\begin{aligned} I_{K,Z} &= H_Z - H_E && (3.1)\\ &= H_K - H_{K|Z} && (3.2)\\ &> 0 && (3.3) \end{aligned}$$

where H_X is the Shannon entropy of a random variable X introduced in Sec. 2.2.1. $H_E < \log|\boldsymbol{E}|$ reflects that plaintexts are written in French.

This implies that $H_{K|Z} < H_K$, i.e. that $p_{K|Z}$ has a lower entropy than p_K.

Of course, in practice the evaluator usually cannot compute the probability density $p_{K|Z}$, but it illustrates that some acquired knowledge related to the cryptosystem or the communication can be used to reduce the entropy of the key. This will be of importance later in the context of side-channel attacks, we will use the knowledge related to some physical signals, emitted by the electronic implementation of a cryptosystem.

3.1.2 Guessing Entropy

We note D some knowledge acquired by the evaluator, this could be for example, a set of power consumption signals in the form of real data vectors. Let $p_{K|D}$ the probability of the keys given that knowledge. The *guessing entropy* [83] is used in cryptanalysis to measure the average number of keys an evaluator has to test before finding the correct one. It is expressed as:

$$\mathrm{GE}_{\mathrm{th}}(p_{K|D}) = \sum_{i=1}^{N} \sigma_p(i) p_{K|D}(i)\,, \tag{3.4}$$

where we numbered here for convenience the keys from 1 to the total number of keys N and σ_p is a permutation of $\{1, \ldots, N\}$, depending on $p_{K|D}$, such that the

probabilities are sorted in decreasing order

$$\forall i, j \in \{1, \ldots, N\},\ \sigma_p(i) < \sigma_p(j) \text{ if } p_{K|D}(i) > p_{K|D}(j)\,.$$

In [83], it is shown for a distribution $p_{K|D}$ with entropy $H_{K|D} > 2\log 2$, that its guessing entropy is lower bounded by

$$\mathrm{GE}_{\mathrm{th}}(p_{K|D}) > \frac{1}{4}e^{H_{K|D}} + 1\,, \tag{3.5}$$

it confirms the intuition that the average number of attempts to guess the key increases with the entropy of the key given the available knowledge. But, it is also proven in [83] that no upper bounds on $\mathrm{GE}_{\mathrm{th}}$ exists, so even if we know $H_{K|D}$ for example, we cannot tell how many key attempts are needed to hope to find the key.

As stated before, it is not always possible to derive the true conditional distribution $p_{K|D}$. In practice, we may only be able to get an approximation of $p_{K|D}$, noted $\tilde{p}_{K|D}$. Hence, in this case, we cannot use the previous definition (3.4) of the guessing entropy.

Alternatively, in the process of evaluating the security of a cryptosystem against an attack model with guessing probability $\tilde{p}_{K|D}$, the key k^* is fixed, and we define the guessing entropy as the rank of this key in the sorted sequence of probabilities:

$$\mathrm{GE}(\tilde{p}_{K|D}) = \sigma_{\tilde{p}}(k^*)\,, \tag{3.6}$$

with $\sigma_{\tilde{p}}$ the permutation for sorting the probability density $\tilde{p}_{K|D}$ on $\{1, \ldots, N\}$.

This is the definition of the Guessing Entropy that we will use throughout this thesis. It requires that we have access to the device in order to fix the key k^*.

3.1.3 Classification of attacks

We can classify attacks according to the way the knowledge D about the cryptosystem is gathered. For a time, let's consider a cryptosystem implemented on a smart-card, that we will introduce in the next section.

An attack can be either considered *active* or *passive* [1]. It is considered *active* if the evaluator tampers with the functioning of the card, either on hardware or software. Each tampering action of the evaluator is aimed at disclosing new information. These attacks have to be carried out with caution as some installed countermeasures may react and lock or kill the chip. If the evaluator only monitors the functioning of the card to gather information, the attack is considered passive.

[1] It reminds the distinction made in machine learning between reinforcement learning and unsupervised/supervised learning.

The other criterion is whether the attack is *invasive* or *non-invasive*. For an invasive attack, the evaluator has full access to the device's material and can plug for example measuring instruments or tampering tools directly on the chip's wires. In the case of a non-invasive attack, the evaluator relies solely on available external information.

More details on the classification of attacks can be found in [48].

3.1.4 The Advanced Encryption Standard (AES)

In this thesis we will perform attacks on the Advanced Encryption Standard. It is a widely used cipher algorithm. It is based on the Rijndael algorithm [30]. It processes plaintexts of 128 bits using keys of length 128, 192 or 256 bits. The data is arranged in a 4×4 array of bytes (8-bit vectors). We give below the pseudocode of the AES:

Algorithm 1: Pseudo code for the AES. The variable are indexed done relative to bytes.

Input: 128-bit Plaintext e, 128/192/256-bit Cryptographic key k
Output: 128-bit Ciphertext z
$k \leftarrow$ `KeyExpansion`(k)
$z \leftarrow e$
$z \leftarrow$ `AddRoundKey`$(z, k[0, 3])$
for $r \leftarrow 1$ **to** $N - 1$ **do**
    $z \leftarrow$ `SubBytes`(z)
    $z \leftarrow$ `ShiftRows`(z)
    $z \leftarrow$ `MixColumns`(z)
    $z \leftarrow$ `AddRoundKey`$(z, k[4r, 4(r + 1)])$
end
$z \leftarrow$ `SubBytes`(z)
$z \leftarrow$ `ShiftRows`(z)
$z \leftarrow$ `AddRoundKey`$(z, k[4N, 4(N + 1)])$
return z

- `KeyExpansion` transforms a cryptographic key of size 128/192/256-bit into an expanded key of size $128(N + 1)$-bit where the number of rounds N is $10, 12$ or 14 depending on the key size.
- `AddRoundKey` performs a bitwise `xor` operation between the state variable z and the expanded subkey of size 128-bit. The state variable z is flattened column-wise before.
- `SubBytes` is a nonlinear operation on each byte of z.

- `ShiftRows` cyclically shifts each row of z respectively of $0, 1, 2$ and 3 bytes.
- `MixColumns` multiplies each column of z by a fixed matrix.

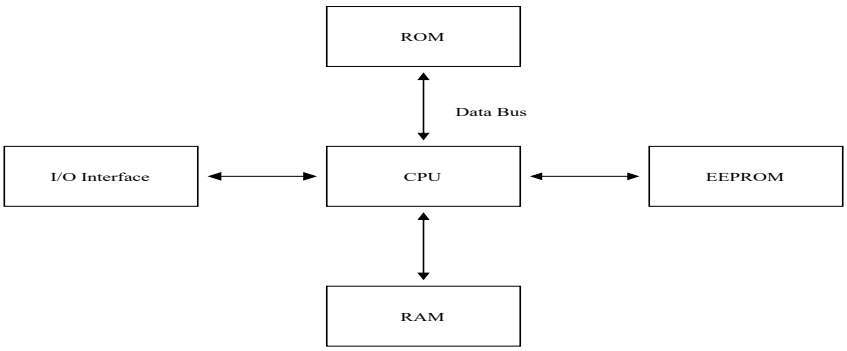


Figure 3.2: General architecture for a smart card. It is made of an Input/Ouput interface in charge of exchanging data with the exterior, a central processing unit (CPU), an Electronic Erasable Programmable Read Only Memory (EEPROM) that stores user related information such as cryptographic key, a ROM memory containing the software and a RAM memory used to store intermediate results during the algorithm execution. This diagram is inspired from [107]

3.2 Smart-Cards

3.2.1 History

A smart-card is a device with an integrated circuit chip, we usually picture it as a plastic card with a small chip on it. The idea of smart-cards emerged around 1970 in Germany, Japan and France. They were first produced by Motorola Semiconductor in conjunction with Bull in 1977, and first tested in French cities in 1980. Now, smart-cards are widely used worldwide, especially in Europe, North-America and Asia-Pacific. They are present in various applications such as user authentication, finance, healthcare and transportation, to name a few.

Between 2013 and 2019, the cost of fraud on credit cards doubled from 13.70 to 28.65 billions of dollars [103]. Thus, the study of smart-card security is of particular importance. Side-channel attacks, introduced in the next section, present a prominent threat to these devices.

We present Fig. 3.2 a general architecture for smart-cards. It is not representative of all encountered smart cards, but it gives a general understanding of their functioning.

To properly understand why side-channel attacks can be performed on smart-card, we present roughly in the next section the physical characteristics of integrated chips and why information relative to computations can leak into electromagnetic signals.

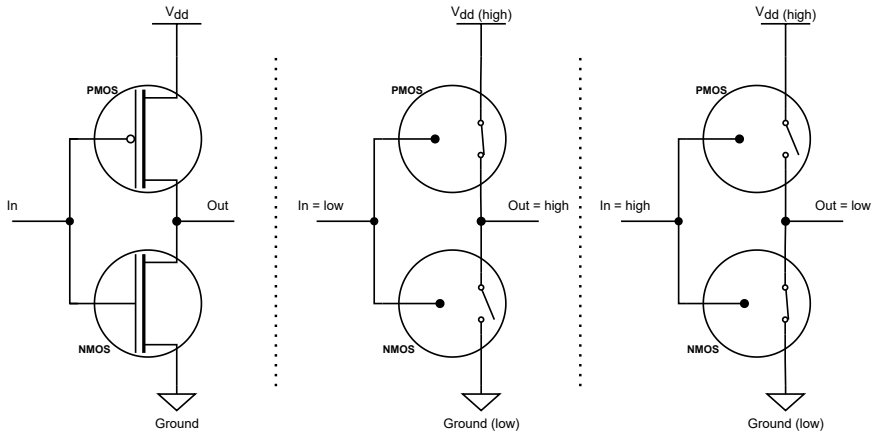


Figure 3.3: A CMOS inverter and its representation with switches for high/low voltage input values. To simplify, on righthand switch-models, we omitted the switching resistance and oxide capacitance of both transistors.

3.2.2 CMOS circuits

Metal Oxide Semiconductor Field Effect Transistors (MOSFET) are the elementary building blocks of the smart-card's chip circuit. They exist in two types, those that are positively doped (pMOS) and those that are negatively doped (nMOS). In digital circuits, they are controlled by logic signals oscillating between two voltage values that represents logic states, 1 at high potential and 0 at low potential. Under control of logical signals, they act as switches that pass or cut the current between two regions of different potentials. A voltage output can be placed to measure if the switch is on or off, such that the MOSFET becomes a system processing input logic signals. For example, the nMOS can be used as an inverter to transform low voltage levels into high and vice-versa, while the pMOS works in the opposite way and let logic signals pass. More complex logical functions can be implemented by combining these two types of MOSFET into an electronic circuit. The Complementary Metal Oxide Semiconductor (CMOS) is a widely used circuit design that combines both MOSFET. To illustrate this, we depict Fig. 3.3 an example of a CMOS inverter composed of a nMOS and a pMOS.

Many other devices can be constructed using MOSFETs such as memory components or oscillators to generate clock signals. The book [9] is a good reference on the physical properties of MOSFET and on the design of modern CMOS circuits.

The chip from a smart-card is formed by many basic CMOS circuits that are interconnected to perform calculations and store results. These circuits are controlled by a clock signal in order to organize and run a step-by-step computational algorithm. We understand then that each time a logic signal, such as the clock, changes from one logic state to the other, it is likely to switch many MOSFETs that will

trigger in the chip a flow of electrons between supply and ground voltage regions. This current can be observed by monitoring the power consumption of the device, or alternatively by measuring the scattered electromagnetic field.

3.2.3 Electromagnetic scattering

If a current of electrons flows through the chip, a magnetic field is scattered, this is modelled by the Maxwell-Ampère's law (the 4th Maxwell's equation). If an electromagnetic probe made of a looped wire is placed close to the chip, the temporal variation of the magnetic flux through the looped wire will induce a difference of potential in the wire, this is the Faraday's law of induction (the 3rd Maxwell's equation). By measuring the potential in the looped wire, we can deduce the variation of the magnetic field enclosed by the looped wire and consequently a variation of a current in the chip. In practice, the magnetic field scattered by the chip is complex and hard to evaluate analytically.

Although we cannot predict how the magnetic field scattered by a functioning chip behaves, the observation of electromagnetic signals has some advantages since the measuring probe can be placed around specific areas of the chip in order to measure the variation of current in this region. For example, we might be interested in measuring the activity of a region dedicated to cryptographic computations, such as a cryptoprocessor.

3.3 Side-Channel Attacks

At the end of Sec. 3.1, we classified attacks according to the knowledge acquired by the evaluator. We can go a bit further and form the family of side-channel analysis. Side-channel analysis relies on the unintentional dissemination of information in the form of physical signals, and whose leakage depends on the design of the electronic devices.

3.3.1 History

As an academic field, side-channel analysis is mainly focused on the security analysis of cryptographic devices and started around the 90s with founding papers [67, 64]. However, we have some historical examples that remind us of side-channel analysis as they rely on the acquisition of information from compromising signals of electrical origin.

The first example dates back to the first World War, where German and Allied soldiers used wireless equipments on the field to gather intelligence from enemy telephone lines. This led A. C. Fuller, a senior British officer at the time, to invent a current DC signalling phone (the Fullerphone) that limited the emanation of induced currents and thus potential eavesdropping on telephonic lines. This is a good historical example of the design of a countermeasure in response to eavesdropper taking advantage of how the information, in the form of electrical signals, were relayed on the field.

After the second World War, the U.S. Government started in the mid-50s the TEMPEST program for the investigation and study of compromising emanations. This was in response of many examples at the time of spying techniques relying on compromising emanations. A famous example is the spying equipment "The Thing" that was installed by Russian intelligence in a US embassy in Moscow until it was discovered in 1950. Its functioning puzzled the Americans who turned to British officials and in particular Peter Wright, scientific officer at the time, to understand the device. Many years later in the 60s, he participated to operation STOCKADE whose goal was to decipher secured communications to and from the French embassy in London. In our case, it is an interesting example as it is one of the first recorded and declassified use of side-channel (or TEMPEST, using military terminology) techniques for analyzing signals from cipher machines. Many other details on the last two examples can be found in Peter Wright's book "the Spycatcher" [120], and declassified information on the TEMPEST program along with a timeline of TEMPEST attacks can be found in [29].

3.3.2 Goal

The goal of a side-channel analysis is to reduce the entropy of a cryptosystem's key by the analysis of a set of signals acquired during the working of an electronic device implementing the cryptosystem. We illustrate Fig. 3.4 the relation between electromagnetic signals and cryptographic keys.

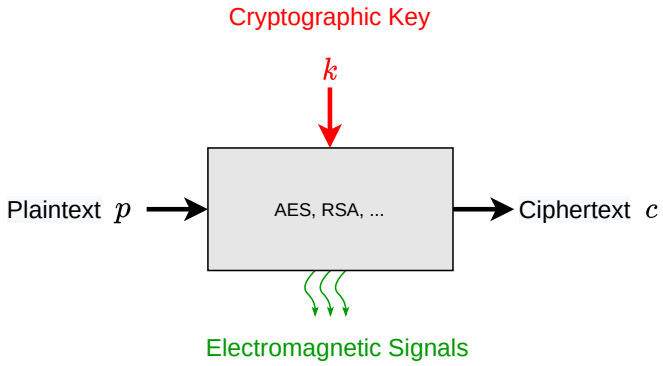


Figure 3.4: Side-channel analysis with electromagnetic signals.

It can be seen as a decision-making process, enlightened by some knowledge D, and relying on a model distribution $\tilde{p}_{K|D}$. Although, in practice, we are not restricted to the sole approach of approximating $p_{K|D}$. In general, the decision can be made according to an arbitrary function or algorithm, called a *distinguisher*, designed or fed with the gathered knowledge D, and outputting decision scores in $\mathbb{R}$, e.g. a statistical test based on a leakage model.

We could draw comparisons with classification tasks in machine learning. But, we identify two reasons for which it slightly differs from a pure classification task.

In a pure classification task, if we are given a signal s and its label o then we impose naturally that the true distribution $p_{O|S}$ is a Dirac at o[2]. In side-channel analysis, we have reasons to think that the true distribution $p_{K|S}$, where K plays the role of the label here, may not be a Dirac at the true key k^*, i.e. all the information about the key may not leak in signals such that the entropy of $p_{K|S}$ does not vanish.

The other reason is that we may not be able to directly get enough information about the key in signals, or that we cannot or do not want to change the cryptographic key on the electronic device. Thus, we may try to get the value of subsidiary variables, called *sensitive variables* whose value depends on the cryptographic key and the inputs. In that case, the information recovered on the sensitive variables from the cryptographic algorithm are recovered and propagated back through the algorithm towards the key. In complex situations, with many sensitive variables, it can be done for example with a Belief Propagation algorithm on the factor graph rep-

[2]For the classification of everyday images in supervised learning, it seems reasonable to assume that $p_{O|S}$ is a Dirac since we have examples of systems (us) that have labeled these images as such.

resentation of the cryptographic algorithm, see [117, 70]. If more than one sensitive variable must be recovered to retrieve the key, the side-channel attack is considered of *higher-order*.

In the following of this thesis, we are interested in the recovery of sensitive variable instead of the direct recovery of the key in signals. For example, we will perform attacks with known plaintexts on the output of the `SubBytes()` operations in AES, see Sec. 3.1.4.

3.3.3 Side-channel attack against AES

We detail here the attack procedure on the `SubBytes()` operations in AES. We note g the `SubBytes()` operation in AES that is computing a sensitive variable Z with a plaintext E and a key K. During its execution, the procedure is leaking signals S, or traces, e.g. electromagnetic or current consumption signals. Traces are acquired in the form of real d-dimensional vectors. From the perspective of the evaluator, all of these variables are considered as random and written uppercase.

The procedure $g(\,.\,, K) : \boldsymbol{E} \rightarrow \boldsymbol{Z}$ is here bijective and maps the set of plaintexts $\boldsymbol{E}$ to the set of sensitive variables $\boldsymbol{Z}$. A *profiled attack* consists of training a classifier h on signals S to recover Z, which gives clues on K given E. The training requires a set of observations labeled with their associated sensitive variable. The training set $\mathcal{D}_t$ is made of tuples $\mathcal{D}_t = \{(s_1, z_1), \ldots, (s_{N_t}, z_{N_t})\}$ with N_t being the size of the training set. An attack set $\mathcal{D}_a = \{s_1, \ldots, s_{N_a}\}$ of size N_a has a fixed key k^* and allows us to evaluate the performance of the attack. Here it is assumed that plaintexts are always known, thus for each realization $(s_i, z_i) \in \mathcal{D}_t$ or $s_i \in \mathcal{D}_a$ a plaintext e_i is associated, i.e. $z_i = g(e_i, k)$. A classifier, noted h here, is trained on $\mathcal{D}_t$ in order to have an approximation of $p(Z|S)$. During an attack, we can get an estimation of the target key k^* with a realization $s_i \in \mathcal{D}_a$:

$$p(K = k|S = s_i) = p(Z = g(e_i, k)|S = s_i) \tag{3.7}$$

However, if the quality of estimations are too poor, one-shot estimation of the key k is in general not enough, i.e. given an observation s_i,

$$k^* \neq \arg\max_k p(K = k|S = s_i)\,.$$

Thus the evaluator has to use many observations to obtain better predictions:

$$p(K = k|\mathcal{D}_a) = \prod_{i=1}^{N_a} p(Z = g(e_i, k)|S = s_i) \tag{3.8}$$

After sorting $\{p(K = k_j|\mathcal{D}_a)\}_{k_j \in \boldsymbol{K}}$ in decreasing order, the rank is defined as the

position of $p(K = k^*|\mathcal{D}_a)$ in the sorted list $p(K = k_i|\mathcal{D}_a) > ... > p(K = k_j|\mathcal{D}_a)$. The Guessing Entropy of Sec. 3.1.2 is estimated by taking the empirical mean of rank values obtained for many attacks. Note that the less attack data N_a is required to have a low rank, the better is the attack. The attack involves the task of estimating the posterior $p(Z|S)$ or the likelihood $p(S|Z)$ from the data.

In our attack methods, we will use this canonical procedure to recover keys from AES implementations.

3.3.4 Literature

We distinguish three main categories in Side-Channel Analysis (SCA):

- *Timing Analysis* is based on the relative time it takes for instructions and more generally algorithms to execute, e.g. the recent Meltdown and Spectre attacks on CPUs [65, 75].

- *Power analysis* monitors the electrical power consumption of devices, by analyzing the waveform of signals we may deduce what type of operations are executed.

- *Electromagnetic analysis* differs from Power analysis by the fact that it does not require a direct access to the wires of the system, instead it uses magnetic probes to measure the variation of the consumed electrical current.

In our presentation of the literature, we will focus on Power and Electromagnetic analysis. The literature has been shaped by the concomitant development of attacks and countermeasures. We find best to start by presenting the countermeasures.

Countermeasures. In order to perturb the analysis of side-channel signals, it has been early suggested that signals should be desynchronized between instructions to undermine their statistical analysis, such as the computation of their first and second order statistical moments. We called them *jitter* countermeasures. One of the first jitter countermeasure is the Random Process Interrupts [26] method that randomly adds dummy instructions between legitimate instructions from the cryptographic algorithm. This method has then been improved to produce random delay between instructions in [16, 27]. In particular, the distribution of the delay is aimed to be uniform over a bounded region of time. More sophisticated hardware techniques, such as the employ of non-deterministic processors has been envisaged in [57, 87], and asynchronous logic has been studied in this context in [89, 90].

As an important and efficient category of countermeasure, *masking* countermeasures are software based methods that split the secret about the key into multiple

sensitive variables. Hence it requires a careful analysis of the cryptographic algorithm to identify leaking instructions and to recombine information to get the key. This is the type of countermeasure that led to the establishment of higher-order side-channel attacks. It has been first presented in [58] with a security proof in [102].

Single and Differential Power analysis. The analysis of the power consumption of implemented cryptosystems has been first published in [66]. Two types of attacks based on the analysis of power consumption are considered and differ in the key decision process: Simple Power Analysis (SPA) can infer the key directly given one power signal but requires specific knowledge about the device; Differential Power Analysis (PDA) uses many power signals to infer the key, but the decision is based on general assumptions about the device. Both attacks do not require the same type of knowledge during the key decision process. When using electromagnetic signals (also named EM signals), both attacks are termed Simple Electromagnetic Attacks (SEMA) and Differential Electromagnetic Attacks (DEMA).

Leakage Model. Kelsey et al. early proposed in 1998 in [64] that if a bit-vector is manipulated through a CMOS circuit then the amount of energy consumed is proportional to the Hamming Weight of the vector, i.e. to the amount of bits equals to one. Brier et al. in [18] go further by proposing the Hamming distance which generalizes the Hamming weight, it assumes that the power is proportional to the number of flipped bits between the output and the input of a targeted logical operation. Later, some works have shown that bits are actually leaking dissymmetrically, suggesting that the leak is of complex nature, for example Suzuki et al. in [114] proposed leakage models that consider operations on bits in CMOS logic circuits to explain biases in power consumption. Schindler et al. in [106] make a linear regression of the leakage model by assuming that the power consumption can be approximated by a weighted sum of a basis of functions defined on the logical operation. The optimality of distinguishers for partially known linear models has been studied in [50]. The derivation of a general leakage model is particularly difficult as it depends on the device, electrical wiring and the installed countermeasures.

Statistical tests. Originally in [66, 64] the key decision process has been based on the difference of means between two datasets separated according to a key hypothesis. It laid the ground for many other statistical test based methods. We refer to the theses of Y. Linge and T. Le [74, 69] on this topic. More recently, the Welsch t-test has been used in several work as a rapid test for the security of an implemented cryptosystem, see [112] and references therein. A statistical test based on

the mutual information has early been considered as a potential statistical test for distinguishing keys, see for example [12], but the inherent difficulties first for modelling the distributions $p_{K,S}$, p_K and p_S, and to perform the calculations, actually undermined its use in comparison with other tests. Recently, the method proposed by [28] which rely on the Mutual Information Neural Network of [14] makes the use of Mutual Information viable again. The absence of strong assumptions on the leakage model and its ability to model and evaluate the mutual information, makes it a strong candidate for evaluating leakage in SCA.

Machine learning models. In 2002, Chari et al. [22] proposed Template Attacks modeling side-channel signals with multivariate Gaussian models. It is a popular method based on the construction of "templates" (assimilated to a learning) for each value of the sensitive variable. The underlying method and its learning algorithm is presented in Sec. 2.5.1. This approach has been completed with the use of kernel functions through Kernel Discriminant Analysis for classification and dimension reduction in [19, 124]. As popular machine learning techniques, Support Vector Machine (SVM) and Random Forest has been studied for side-channel analysis in machine learning focused articles [54, 71].

Deep learning models has been used with success in side-channel analysis for both treating the problem of jitter and masking countermeasures. Multi-layer perceptrons has been used in [82] and convolutional neural networks in [78, 20, 15]. The leakage model is learned in a black-box manner through the training of a deep network. This allows the establishment of higher-orders attacks against masking countermeasures by automatically drawing relations between different regions in signals. Convolutional neural networks are particularly efficient in the acquisition of sensitive information in jitter protected signals without requiring specific realignment of signals. We refer to [84] on the topic of deep-learning methods for side-channel analysis.

Hidden Markov models has early been considered in the context of side-channel analysis to model the underlying cryptographic algorithm [63, 41, 40] and drive a more general approach in the treatment of side-channel signals. The operations are considered as states that leak in signals through a generative model. This approach has been extended to more general probabilistic graph in [117, 70]. Such approaches are interesting as they readily take into account a representation of the algorithm in the side-channel analysis of signals. Inductive reasoning on probabilistic graphs is a major field of study in machine learning and artificial intelligence[99, 68, 118], and the recent coupling with deep learning methods in time-series analysis presents interesting direction for the automatic analysis of side-channel signals.

Time-Frequency Analysis for Side-Channel Analysis.

Fourier analysis. It has been early shown by [3] that EM signals of various cryptographic implementations can be analyzed in the Fourier domain and that DEMA can be successfully carried with carefully chosen frequency bands. Differential power analysis, usually carried with temporal signals, has been transposed in the frequency domain in [44, 100, 13]. The use of Fourier transforms in side-channel presents the advantage of being robust, to some extent, against small jitter in signals and as long as the signal is not too much parasitized by other patterns. The use of spectrograms as been proposed as a preprocessing step in convolutional neural networks in [122], they obtain the same efficiency of CNN-based attacks with raw traces.

Wavelet analysis. Discrete Wavelet transforms has first been used in [23] to realign side-channel signals with a simulated annealing method. In [35], the authors employ template attacks with discrete wavelet transforms of signals. They demonstrate that for synchronized signals, discrete wavelet transforms present better results in comparison with attacks using raw temporal signals. They also show that attack results get better by increasing the level of frequency resolution of the wavelet transform. More recently, [91] present a realignment algorithm based on the extraction of features through wavelet transforms with block wavelets. To this day, no CNN-based side-channel attacks with scalograms has been proposed in the literature. However, as part of the supervision of the internship of P. Afro at the ITSEF, we showed that in the context of side-channel analysis CNN-based attacks with scalograms and spectrograms presented similar results.

Waveform analysis. If we are able to acquire waveforms related to targeted operations, we can use them as a basis of analysis. It can be seen as a particular time-frequency analysis where the elements of the basis are learned a priori. This method has been used in [53, 41] in order to detect operations in signals.

Position of this thesis. Generally speaking, the use of time-frequency analysis is rather timid in the community of side-channel analysis and not perceived as a useful step for preprocessing signals. We will try in this thesis to show some advantages brought by wavelet analysis. While masking countermeasures are currently being tackled by deep learning methods or probabilistic graph learning algorithms, we aim in this thesis at proposing ways of handling jitter countermeasures through wavelet analysis. We will adopt two different approaches; either we try to suppress the jitter noise by mapping the signals to representations stable under small translations; or

we detect and extract each individual patterns in signals to completely suppress the jitter. We will show that in both approaches wavelet analysis can play an important role.

The first approach is developed in Sec. 4.4. We study the use of the scattering transform of S. Mallat [81] to reduce the effect of the jitter present in side-channel signals. The scattering transform average scalograms obtained with continuous wavelet transforms along the time domain to make it robust against small translations. With the scattering transform, we show that we efficiently improve template attacks on desynchronized signals.

To deal with the second approach, we start by presenting in Sec. 4.2 a simple wavelet-based method for extracting patterns from scalograms and resynthesize them as waveforms for realignment, thus extending the previous works on waveform analysis for side-channel signals. Actual works in the side-channel literature making use of wavelet transforms for analyzing side-channel signals have been comparing many different wavelet bases for discrete wavelet transforms. Usually, the choice of a wavelet basis is motivated by some analytical properties of signals. In our case, we do not have analytical arguments that may motivate a particular choice of wavelet basis, in other words the patterns encountered in side-channel signals greatly vary according to the acquisition conditions, and a particular choice of wavelet basis for a specific experience may not work in other conditions. Thus, we propose to work with a superfamily of wavelets, namely the Generalized Morse Wavelets, that present some flexibility in their design such that this family of wavelets may be adapted for each particular dataset of side-channel signals. In Sec. 4.3, we study the learning of a frame of Generalized Morse Wavelets given a set of extracted patterns, and we use it to carry out side-channel attacks. Last, in Chap. 5, we argue that we cannot solely rely on the correlation-based technique that is behind waveform analysis to detect patterns in signals. We have to consider the underlying dynamical structure of side-channel signals to efficiently locate patterns. This amounts to consider the effect that the jitter and the algorithm can have on the dynamics of the signal. Thus, we propose a point process model for modeling a continuous jitter and rely on previous work in the literature to model the algorithm with Hidden Markov Models (HMM). This allows the construction of a statistical generative model for side-channel signals. We envisage a learning algorithm to fit the model to side-channel signals and a method to efficiently detect the patterns in signals. In addition, this model can be used to simulate side-channel signals to test side-channel attack methods and design countermeasures.

Chapter 4

Static Approach

The main motivation for pushing time-frequency preprocessing is to consider bases of analysis in which side-channel signals are represented in terms of elementary signals whose characteristics are closer to emanations from physical phenomena. In the case of side-channel analysis, we do not know a priori neither what form the signals leaking sensitive information have, nor at which time scales the sensitive variable are manipulated. However, we know that the physical processes involved are non-stationary and lasting in time, e.g. the current consumption of a CMOS during a switch. Thus, it seems reasonable to analyze signals with a basis of functions which at least respect these properties. As presented in Sec. 1.3.2, wavelets are oscillating elementary signals whose time scales can be adapted to capture information at different resolution. We will demonstrate in this chapter that the analysis with wavelets presents advantages that will led to the design of new side-channel attack methods.

We will present in this chapter different tools of wavelet analysis for processing side-channel signals, but first we will show some side-channel signals and present their time-frequency properties. We will then propose a simple realignement method based on the extraction, denoising and resynthesis of patterns from scalograms. Next, we will study the estimation of a frame of wavelets for the analysis of patterns in side-channel signals. Finally, we will propose an attack method based on the scattering transform to compensate the effect of jitter countermeasures, and we will derive an ensemble method for the approximation of a leakage model.

4.1 Characteristics of Electromagnetic Signals

In this section, we study electromagnetic signals (EM signals) from different datasets and present their characteristics.

4.1.1 A variety of signals

We present Fig. 4.1 electromagnetic signals and their periodogram (spectral density) from four different datasets. The ASCAD [15] and DPAv4 [93] datasets are publicly available, they contain signals from an AES implementation with masking countermeasures. The CHAXA dataset contains signals from an AES implementation with an implemented noise emission countermeasure that is activated upon perturbation of the near electromagnetic field around the device (it can be activated for example by a probe). The JIT dataset contains signals from an AES implementation with an implemented jitter countermeasure with random delays between operations.

The datasets JIT and ASCAD will be extensively used for side-channel attacks. We detail below some quantities about each dataset. The ASCAD dataset is composed of EM signals emitted from a device running a masked AES implementation, an artificial jitter is simulated by randomly translating signals with an uniformly distributed random variable $\delta_N \sim \mathcal{U}\{0, N\}$. Three sets of signals are available, the first one ASCAD_0 is composed of aligned signals while ASCAD_{50} and ASCAD_{100} are desynchronized respectively with δ_{50} and δ_{100}. Each set consists of $60,000$ signals of 700 points.

The JIT is composed of EM signals acquired from an AES hardware implementation on a modern secure smartcard with a strong jitter of unknown nature. The `SubBytes()` are processed sequentially and all signals are desynchronized and start shortly before the processing of the first byte. In total $160,000$ signals of $400,000$ points are acquired, $150,000$ signals have random keys and $10,000$ signals with a fix key are used to test the attack.

Acquisition conditions We remark on Fig. 4.1 that all datasets present different time and frequency properties. The operations are clearly identified in signals by relatively high variations over bounded periods of time. We will zoom on the properties of these patterns later but for now we note that their forms vary between each dataset. Moreover, in the case of the acquisition of EM signals using magnetic probe, the form of these patterns will vary with the orientation and position of the probe.

The decrease of amplitude at low and high frequency in the periodograms is due to the combination of the bandpass effect of the electromagnetic probe and by low

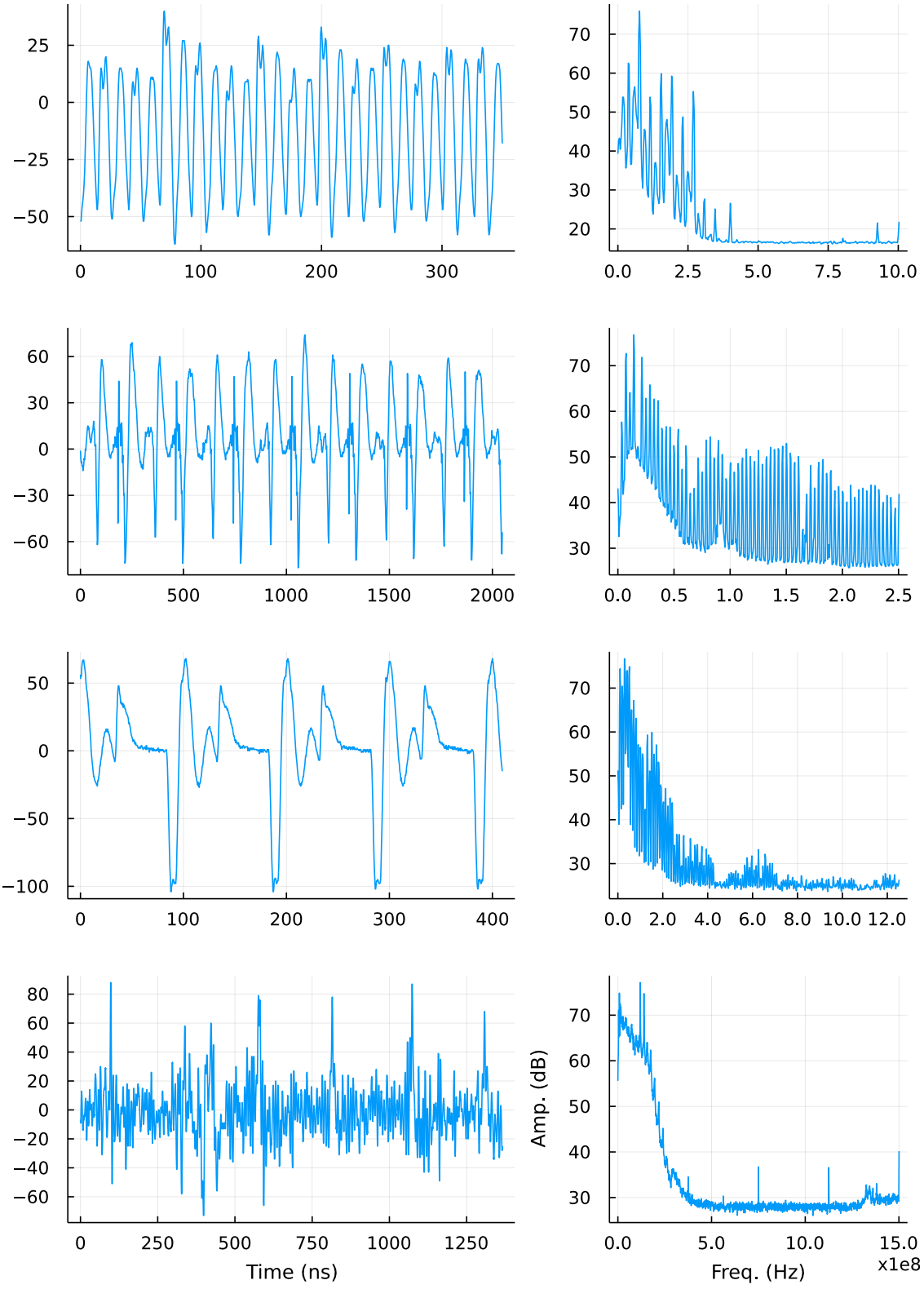


Figure 4.1: On the left: temporal signals. On the right: periodograms. From top to bottom, the signals come from the datasets: ASCAD, DPAv4,CHAXA and JIT.

pass filters before digital conversion. The presence of high frequency harmonics may be caused by distortions induced by amplifiers.

The sampling frequency (F_s) in these examples is of the order of 1GHz. Thus, we could expect to get frequency content up to the Shannon frequency ($F_s/2$), but the bandpass effect of magnetic probes and the use of low pass filters limit the acquisition of information below 1MHz and after 200MHz (roughly).

Jitter countermeasure The presence of Dirac combs in the spectral density of signals indicates a periodicity of patterns. By measuring the spacing of the comb, we can deduce the frequency operation of the device. In these examples, the frequency operation is of the order of 10Mhz.

A jitter countermeasure reduces the amplitude of the Dirac comb. We see the effect of the jitter in the JIT dataset by the low amplitude of the Dirac comb when compared with the periodograms of other datasets.

4.1.2 Noise

The power of the noise in the presented examples of Fig. 4.1 is pretty low in comparison with the amplitude of patterns. With the acquisition conditions of these datasets, the noise does not present a veritable challenge in comparison with masking or jitter countermeasures.

During the acquisition of EM signals using magnetic probe, the noise may come from multiple sources. Signals can be contaminated with signals that come from other areas on the chip. For example if an oscillator is positioned too close to a monitored area or if other computations unrelated to the cryptographic algorithm are performed in parallel and in the vicinity.

When observing electrical signals, it is expected to observe "natural" noises such as thermal noise which is approximately white, shot noise characterized by random spikes or flicker noise with a spectral density following $1/|w|^{\alpha}$, $\alpha \in \mathbb{R}^{+}$. In our case, the lowpass filtering of the probe does not allow us to observe flicker noise but instead a thermal noise such as the one depicted Fig. 4.2.

Finally, the amplitude of signals is usually encoded on 8-bit values thus a quantization noise assumed white is also present in signals.

4.1.3 Information localization in time and frequency

We understand intuitively that the algorithmic operations that may leak information about the key will be localized in time, and that the physical signal related to these leaks will have a bounded frequency spectrum upon observation.

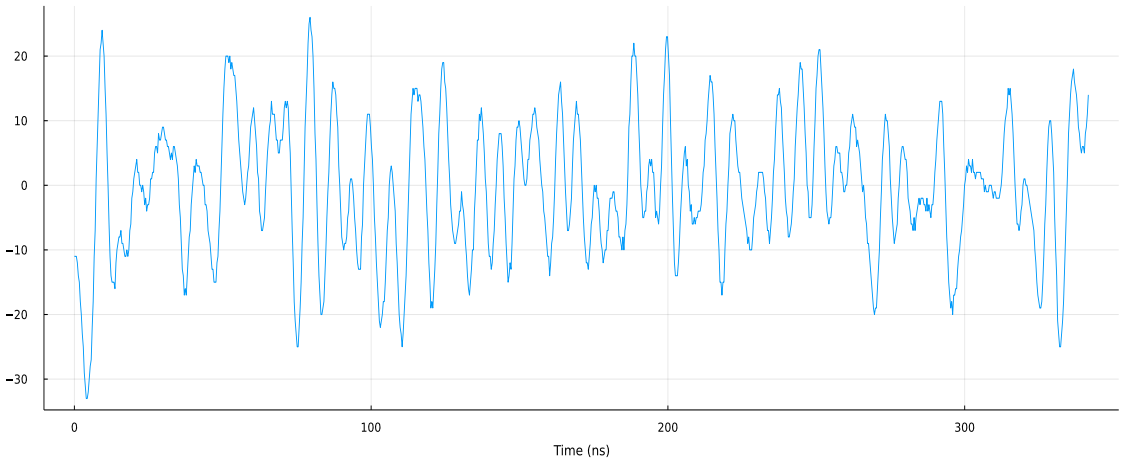


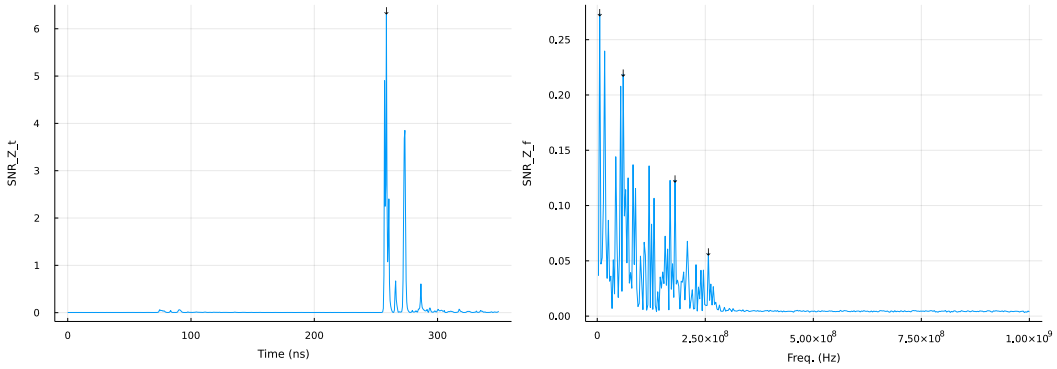
Figure 4.2: A sample of the noise encountered in EM signals, taken between two patterns of instructions in JIT signals.

In simple situations, i.e. without strong countermeasures, we can measure the leak of information by computing a Signal to Noise Ratio (SNR) metric commonly used in the side-channel analysis community. To illustrate the use of this metric, we take the ASCAD dataset as an example, and we compute the SNR in the time and frequency domains. We note Z the sensitive variable, which is the output of the `SubBytes()` operation of the AES, we write S for the acquired signals and $\widehat{S}$ their Fourier density spectrum. The SNR in time and frequency are noted SNR_t and SNR_f, they are given by:

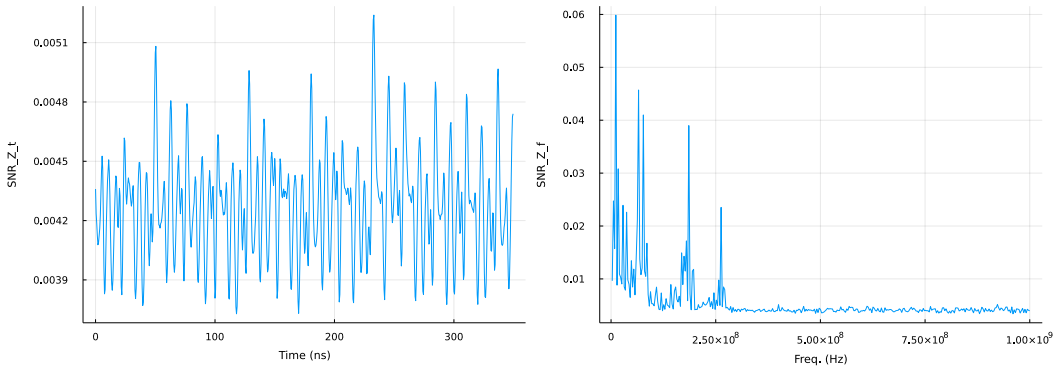
$$\mathrm{SNR}_t = \frac{\mathrm{Var}_Z\left[\mathbb{E}_{S|Z}\left[S\right]\right]}{\mathbb{E}_Z\left[\mathrm{Var}_{S|Z}\left[S\right]\right]}, \tag{4.1}$$

$$\mathrm{SNR}_f = \frac{\mathrm{Var}_Z\left[\mathbb{E}_{S|Z}\left[|\widehat{S}|^2\right]\right]}{\mathbb{E}_Z\left[\mathrm{Var}_{S|Z}\left[|\widehat{S}|^2\right]\right]}. \tag{4.2}$$

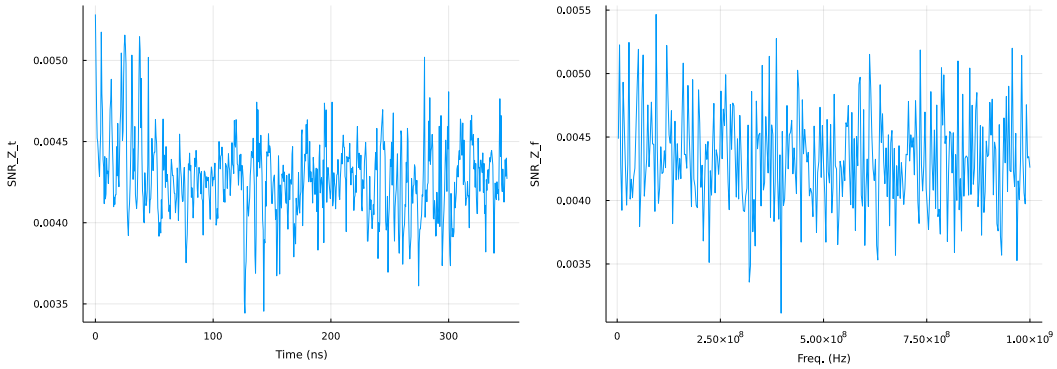
We plot on Fig. 4.3 the evaluation of the two metrics in presence of jitter and masking countermeasures. We first remark that, without countermeasures, the SNR metric efficiently localizes time regions and frequency bands that are correlated to the value of the sensitive variable. The jitter countermeasure, here an artificial desynchronisation, makes the SNR vanish in the time domain and slightly modify the leakage in frequency. However, it does not mean that a Fourier analysis is robust against jitter countermeasure, if the jitter is strong enough the SNR also vanishes in frequency. The masking countermeasure is particularly strong against both time and frequency SNR leakages.



(a) No countermeasure.



(b) Jitter countermeasure.



(c) Masking countermeasure.

Figure 4.3: Leakage localization in time and frequency domains for the ASCAD dataset. On the left the leakage in time, on the right in frequency. From top to bottom, the leakage varies with the countermeasure in place.

4.2 A Wavelet Analysis of SCA signals

In this section, we present wavelet analysis tools for the comprehension of SCA signals in the presence of additive noise and jitter. We present pattern extraction and denoising techniques in the aim of constructing adapted filters for the automatic extraction and realignment of signals.

4.2.1 Multiscale analysis and pattern identification

Continuous wavelets transforms of Sec. 1.3.2 provide alternative representations that help to identify structures in signals. In comparison with the Short-Time Fourier transform, the analysis with wavelets is more sparse and requires fewer coefficients to represent transients.

We take two signals from the JIT and DPAv4 datasets of previous sections and plot Fig. 4.4 their scalograms. The scalograms are computed with a wavelet family $\{\psi_{a,u,1,3}\}_{a\in\mathbb{R}^+,u\in\mathbb{R}}$ made of Generalized Morse Wavelets of the form given in (1.49) in Sec. 1.4.4 and with parameters $(\beta = 1, \gamma = 3)$. The frequency peak of a reference wavelet with scale a_0 is positioned at the highest frequency band we wish to analyze. Starting from this reference other scales are computed with $a_j = a_0 2^{j/Q}, 0 \leq j \leq JQ - 1$ where J is the number of dyadic scales and Q the number of inter scales. In total, JQ wavelets with different scales are constructed. The scalogram is made by convolving the signal with each wavelet and by taking the absolute value. The scalogram at time u and scale a_j is equal to:

$$\left|s * \psi_{a_j,\beta,\gamma}(u)\right|. \tag{4.3}$$

The scalograms presented on Fig. 4.4 help to visualise the patterns and distinguish them. In both datasets, we are able to discern different types of patterns and to localize their occurrence in time. We remark on the scalogram of the JIT signal that the noise is particularly present at low scales and confirms the presence of a thermal type of noise.

4.2.2 Pattern extraction and Denoising

The information gathered through the visualization of scalograms helps in the design of a side-channel attack adapted to a particular dataset of signals. In particular, the patterns can be used to construct a set of adapted filters. These filters will be used for the automatic extraction of patterns and for realignment in the next section.

Patterns are extracted from scalograms by simply cropping them into rectangular patches. Given the extracted portions of the scalograms, we can apply a threshold

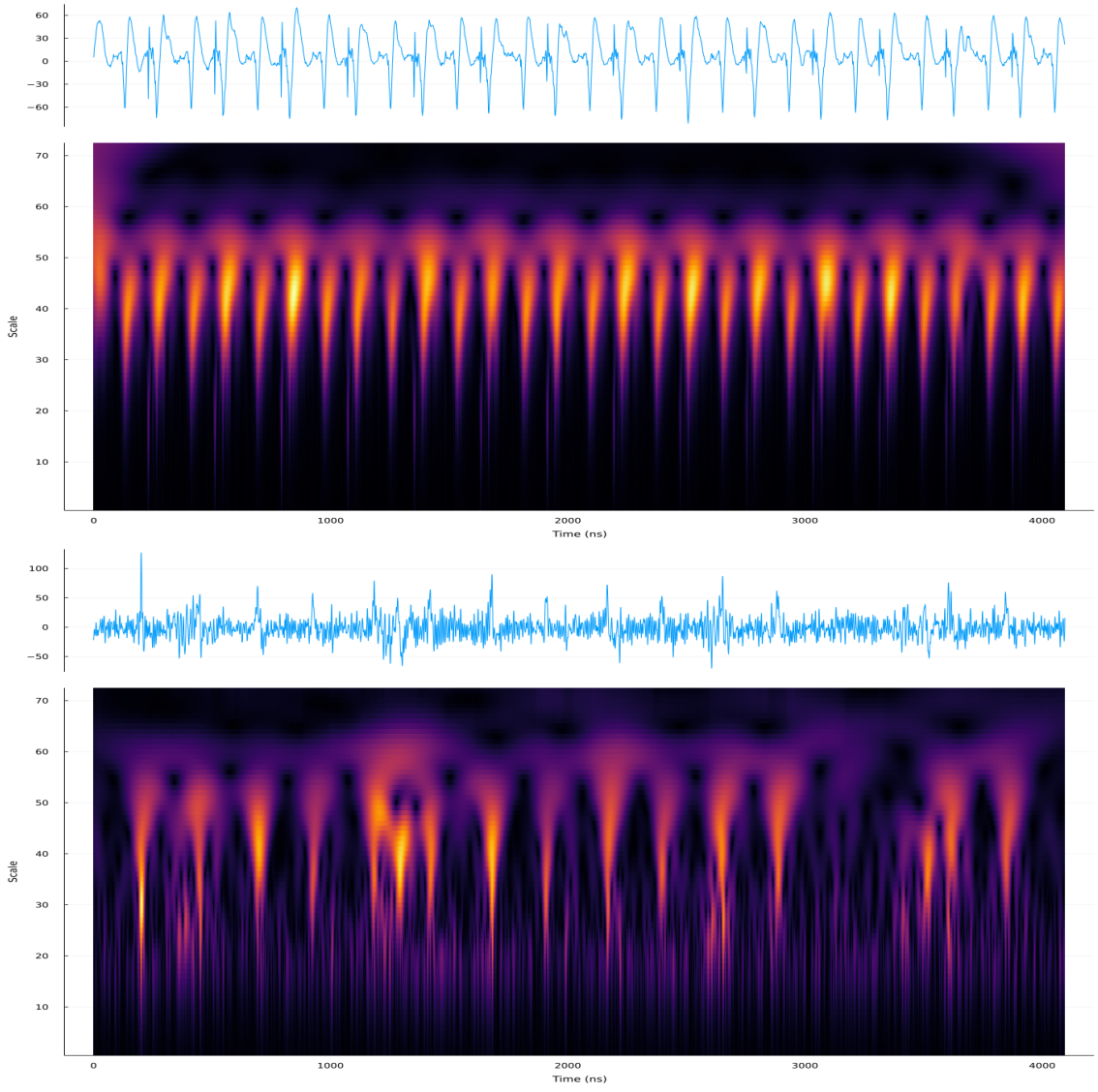


Figure 4.4: Side-channel signals from CHAXA and JIT and their scalograms.

based denoising technique [80, Sec. 11.2.2] to remove the noise with low energy coefficients. Finally, this reduced set of coefficients is used to reconstruct a denoised signal in the time domain.

We use the framework of frame theory of Sec. 1.3.3 to reinterpret our scalogram as a set of coefficients acquired with a frame operator Ψ_Ξ with sequence of wavelets $\{\psi_{a,u}\}_{(a,u)\in\Xi}$. In particular, we recall that the sampling of a convolution product is related to the inner product via:

$$\langle \psi_{a,u}, s\rangle = \psi_a^- * s(u)\,, \tag{4.4}$$

with $\psi^-(t) = \overline{\psi(-t)}$. Thus, the sampling of the convolution products to form the scalogram is equivalent to the computation of inner products with a frame where we adjusted each element of the frame with $\cdot^- : f \to f^-$.

Let s denote our signal, the original coefficients $x = \Psi_\Xi s$ are obtained with

$$\forall (a,u) \in \Xi\,, x\,[a,u] = \langle \psi_{a,u}, s\rangle\,. \tag{4.5}$$

The tuples (a,u) are the elements of the index set Ξ and play the role of the scaling and time translation parameters of each wavelet, see Sec.1.3.3.

The cropping and threshold-based denoising amounts to reduce the index set to

$$\widetilde{\Xi} = \{(a,u) \mid (a,u) \in R,\, |\langle \psi_{a,u}, s\rangle| < \delta\}\,, \tag{4.6}$$

with $R \subset \mathbb{R}^+ \times \mathbb{R}$ the region outlined during the cropping and δ the denoising threshold. The new set of wavelet parameters $\widetilde{\Xi}$ will depend on the signal s. Here, we used a hard type of thresholding, but other strategies presented in [80, Sec. 11.2.2] may be used.

This reduced set is associated to a reduced frame $\{\psi_{a,u}\}_{(a,u)\in\widetilde{\Xi}}$ and hence to a new frame operator $\Psi_{\widetilde{\Xi}}$. The original coefficients have been reduced to $\tilde{x} = \{\langle \psi_{a,u}, s\rangle\}_{(a,u)\in\widetilde{\Xi}}$. With these coefficients, we can search for a denoised signal $\tilde{s}$ such that $\tilde{x} = \Psi_{\widetilde{\Xi}}\tilde{s}$.

4.2.3 Resynthesis

In order to obtain the denoised signal in the new space spanned by the reduced frame, we search solution of the linear system

$$\Psi_{\widetilde{\Xi}}^\dagger \Psi_{\widetilde{\Xi}} y = \Psi_{\widetilde{\Xi}}^\dagger \tilde{x}\,, \tag{4.7}$$

with y the unknown signal we wish to recover. Noting $G = \Psi_{\widetilde{\Xi}}^\dagger \Psi_{\widetilde{\Xi}}$ and $b = \Psi_{\widetilde{\Xi}}^\dagger \tilde{x}$, this linear system is written $Gy = b$. We remark that G is a positive semidefinite matrix

and that b is clearly in the image of G, therefore the system accepts solutions.

By reducing the size of the frame, we take the risk of making G singular, and in consequence the linear system will accept an infinite number of solutions, of the form $y + y_\perp$ with y in the image space of G satisfying $Gy = b$ and $y_\perp$ in the kernel space of G.

In any case, as long as b remains in the image of G, an approximate solution of (4.7) can be found by using an iterative method, such as a conjugate gradient method. We refer to [80, Sec. 5.1.3] on the use of conjugate gradient for linear system solving in the context of wavelet frame analysis and to [62] and references therein on the conjugate gradient method in general.

Computation Cost In order to apply the conjugate gradient method, it is required to compute the matrix G and vector b. If the patterns are extracted from a redundant wavelet transform, which is the case for continuous wavelet transform of finite length vectors, the number of coefficients obtained after extraction and denoising can stay high. Thus, the frame remains large and the cost of computation of G and b is high. To circumvent this problem we propose different methods.

First, the convolutions at high scales (low frequency) can be subsampled to decrease the number of extracted coefficients. Second, if possible, the matrix G can be decomposed into a block diagonal matrix

$$G = \begin{bmatrix} G_1 & 0 & 0 \\ 0 & G_2 & 0 \\ 0 & 0 & \ddots \end{bmatrix},$$

with $b = [b_1, b_2, \dots]$, and the conjugate gradient method can be applied in parallel on the CPU/GPU for solving each linear block $G_i y_i = b_i$, the results $\{x_i\}_i$ are then concatenated to form the approximate solution. This case is encountered when denoised time-scale patterns present regions that are sufficiently separated in time and with null coefficients in-between.

Finally, the construction of G can be performed in parallel and by using the BLAS[1] rank-k update of Hermitian matrix, indeed $G = \Psi_{\widetilde{\Xi}}^{\dagger} \Psi_{\widetilde{\Xi}}$ can be written as the

[1]Basic Linear Algebra Subprograms

sum

$$G = \sum_{(a,u)\in\widetilde{\Xi}} \psi_{a,u}\psi_{a,u}^{\dagger} \tag{4.8}$$

$$= \sum_{i=1}^{p} \sum_{(a,u)\in\widetilde{\Xi}^i} \psi_{a,u}\psi_{a,u}^{\dagger} \tag{4.9}$$

$$= \sum_{i=1}^{p} G_i \tag{4.10}$$

with p the number of processes and $\widetilde{\Xi}^i, 1 \leq i \leq p$ are the partitions of $\widetilde{\Xi}$ dispatched to each process.

As a final note, we remark that the computation cost is drastically reduced if the wavelet transform is computed using an orthogonal basis since in that case the solution is given by $y = \Psi_{\widetilde{\Xi}}^{\dagger}\tilde{x}$. Unfortunately, it constraints the construction of the frame and we lose the advantages in terms of visualization of continuous wavelet transforms.

Results We plot on Fig. 4.5 an example of the extraction of a pattern in the time-scale domain and its denoising with different thresholds. We show examples of the signals obtained with the conjugate gradient method. We used the conjugate gradient method implementation in Julia of IterativeSolvers [61]. Since, the solutions are obtained in $\mathbb{C}^n$, n being the size of the pattern, we show the real part of the solutions.

We also show in Fig. 4.6 the evolution of the residual norm $\|Gx_t - b\|$ for the solution x_t at iteration step t in the conjugate gradient method. During iterations, round-off errors may accumulate in the kernel space of G and make the solution diverges, the residual norm $\|Gx_t - b\|$ does not account for the divergence since Gx_t is in the image space of G. Thus, we can have a decreasing residual norm with a diverging solution. This is the case for example on Fig. 4.6 where we stopped the iterations early as the residual norm was low enough (around 20 iterations). In particular, divergence happens when b do not exactly belong to the image space of G, see [62].

4.2.4 Automatic Detection of Patterns for Realignment

We can now derive a method based on the resynthesis of extracted patterns in scalograms to automatically detect in the temporal domain the other patterns present in the signal. We propose the following procedure:

1. With a family of Generalized Morse Wavelets $\{\psi_{a,u,\beta,\gamma}\}_{(a,u)\in\Xi}$ with β, γ fixed,

we compute the wavelet transform $x = \{\langle \psi_{a,u,\beta,\gamma}, s_1 \rangle\}_{(a,u)\in\Xi}$ of a given signal s_1. For a continuous wavelet transform, it can be computed using convolutions.

2. By visualizing the scalogram $\{|\langle \psi_{a,u,\beta,\gamma}, s_1 \rangle|\}_{(a,u)\in\Xi}$, we crop an interesting region outlining a pattern in the scalogram and denoise it using (4.6) with an appropriate threshold. It gives a reduced set of wavelet parameters $\widetilde{\Xi}$ associated to coefficients $\tilde{x}$.

3. We construct the frame operator associated $\Psi_{\widetilde{\Xi}}$. To resynthesize the denoised pattern $\tilde{y}$, we get linear system $G\tilde{y} = b$ with $G = \Psi^{\dagger}_{\widetilde{\Xi}}\Psi_{\widetilde{\Xi}}$ and $b = \Psi^{\dagger}_{\widetilde{\Xi}}\tilde{x}$. The linear system is solved using a conjugate gradient method, see Sec. 4.2.3.

4. To detect the presence of other patterns in a signal s_2, we use $\tilde{y}$ as an adapted filter and compute the correlation:

$$|\tilde{y}^{-} * s_2| \tag{4.11}$$

The position of the peaks in (4.11) indicate the presence of portions of the signal that are correlated with the denoised pattern $\tilde{y}$. We apply a threshold on the peaks and a minimum time criterion between peaks to filter false positives and extract the position of the detected patterns.

5. We extract the detected patterns $\{y_i\}_i$ in the signal s_2 around the peaks of $|\tilde{y}^{-} * s_2|$. They are then concatenated into a new signal s_2^* containing the detected patterns. It gives a realigned signal s_2^* with a constant delay between detected patterns.

We show Fig. 4.7 the results obtained for each adapted filters computed in the previous section. We notice that the threshold efficiently reduce the noise and the amount of false positive peaks.

We plot on Fig. 4.8 an example of the detection of the patterns in the signal using $|s * g|$ with g the adapted filter of Fig. 4.7 with a high threshold. Given the temporal locations of each pattern, we can extract the detected pattern and concatenate them to form a new synchronized signal.

Application in side-channel Once the patterns are located in the temporal signal, they can be concatenated to form new signals in which the algorithm instructions will be synchronized. The method presented in this section is useful to remove natural clock jitter from oscillators or `nop` operations ("do nothing" operation). A pattern is related to the consumption of energy in some frequency bands

and over a short period of time. Thus, we cannot handle with this method countermeasures that repeat operations or introduce fake ones, as these operations will also be detected and mixed with other pertinent operations.

This method will be used in Sec. 4.3 to automatically extract a set of patterns for estimating an adapted frame of wavelets.

4.2.5 Conclusion

By the visualization of side-channel signals with continuous wavelet transform, the knowledge of an evaluator can easily be involved in order to identify patterns related to the cryptographic algorithm. We presented a method to extract the wavelet coefficients of those patterns in the time-scale domain, a threshold-based denoising procedure for the removal of noise and an inversion scheme of the coefficients for the resynthesis of time-domain signals. These resynthesized signals are then used as adapted filters for the automatic detection and extraction of patterns which are in turn concatenated to form resynchronized signals.

The intervention of the evaluator in the identification of patterns led to the construction of a matrix G that was used in the resynthesis context. This matrix G can also be seen as a metric, assuming that G is made positive definite, e.g. by the addition of a rank-k update matrix, and we can define the following distance between two signals s_0, s:

$$\|s_0 - s\|_G^2 = (s_0 - s)^\dagger G^{-1} (s_0 - s)\,. \tag{4.12}$$

This new distance will shape the space of signals according to G. For signals equally distant in the $\|\cdot\|$ sense, they will be seen closer or farther with $\|\cdot\|_G$. Since G carries the information about the shape of the patterns, signals that do not "look like" true patterns will be pushed away by $\|\cdot\|_G$.

In the next section, we study the estimation of a frame operator Ψ_Ξ by proposing a statistical model. It will allow the construction of G where it will play the role of the covariance of patterns.

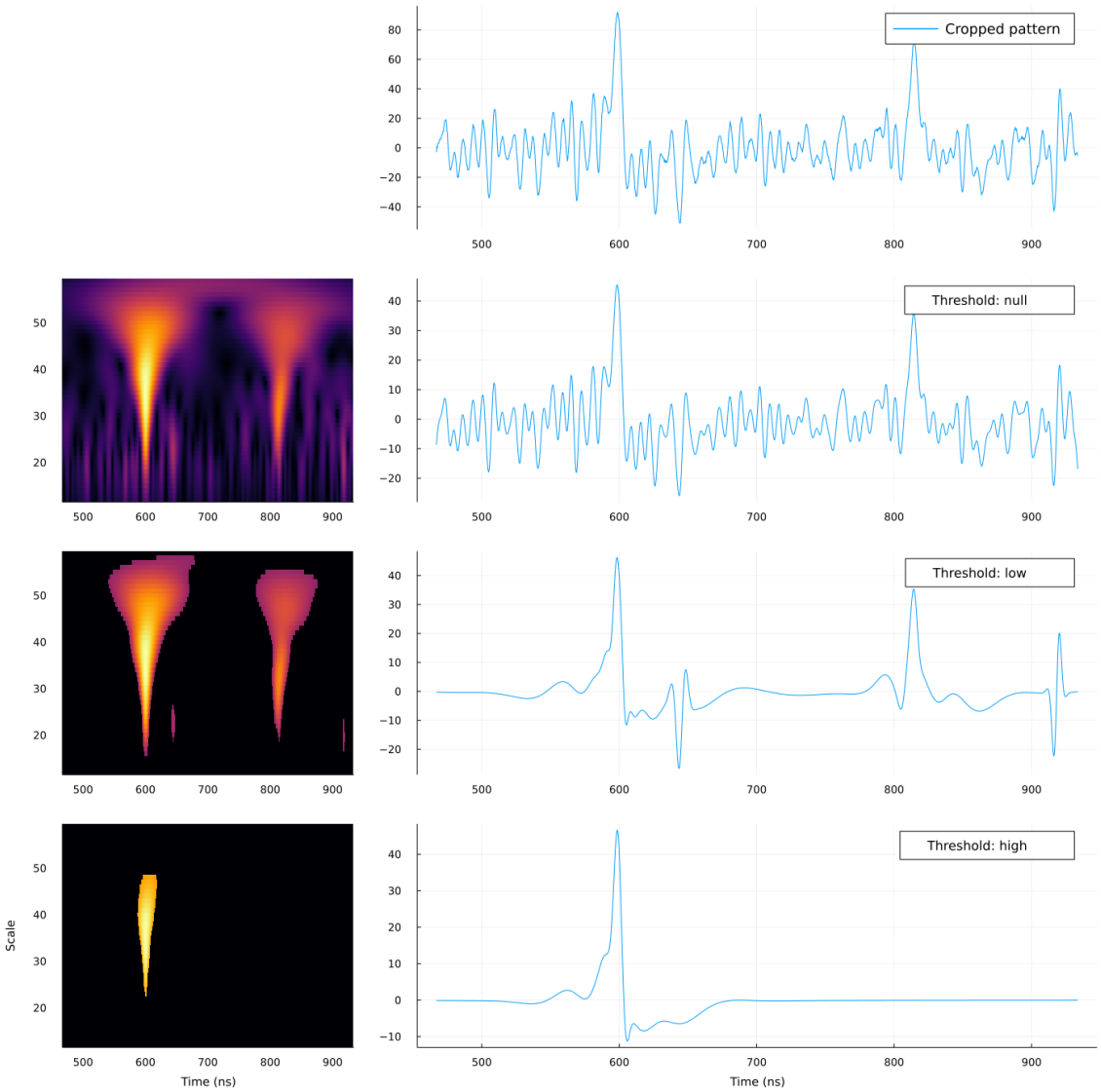


Figure 4.5: Example of the extraction of a pattern from a continuous wavelet transform, of the application of threshold-based denoising, and of the resynthesis of denoised signals in the time domain. At the top, signal in the time domain containing the pattern to extract. For the three bottom lines: on the left a heatmap resulting from the application of three different thresholds (null, low and high) on the extracted time-scale pattern, and on the right the resynthesized signal in the time domain.

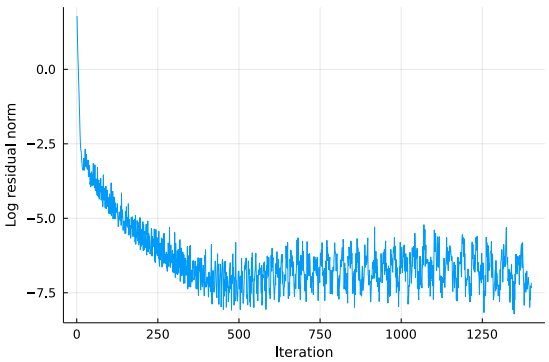


Figure 4.6: Evolution of the log of the residual norm during the conjugate gradient method. It has been obtained during the resynthesis of the cropped pattern of Fig. 4.5 without threshold.

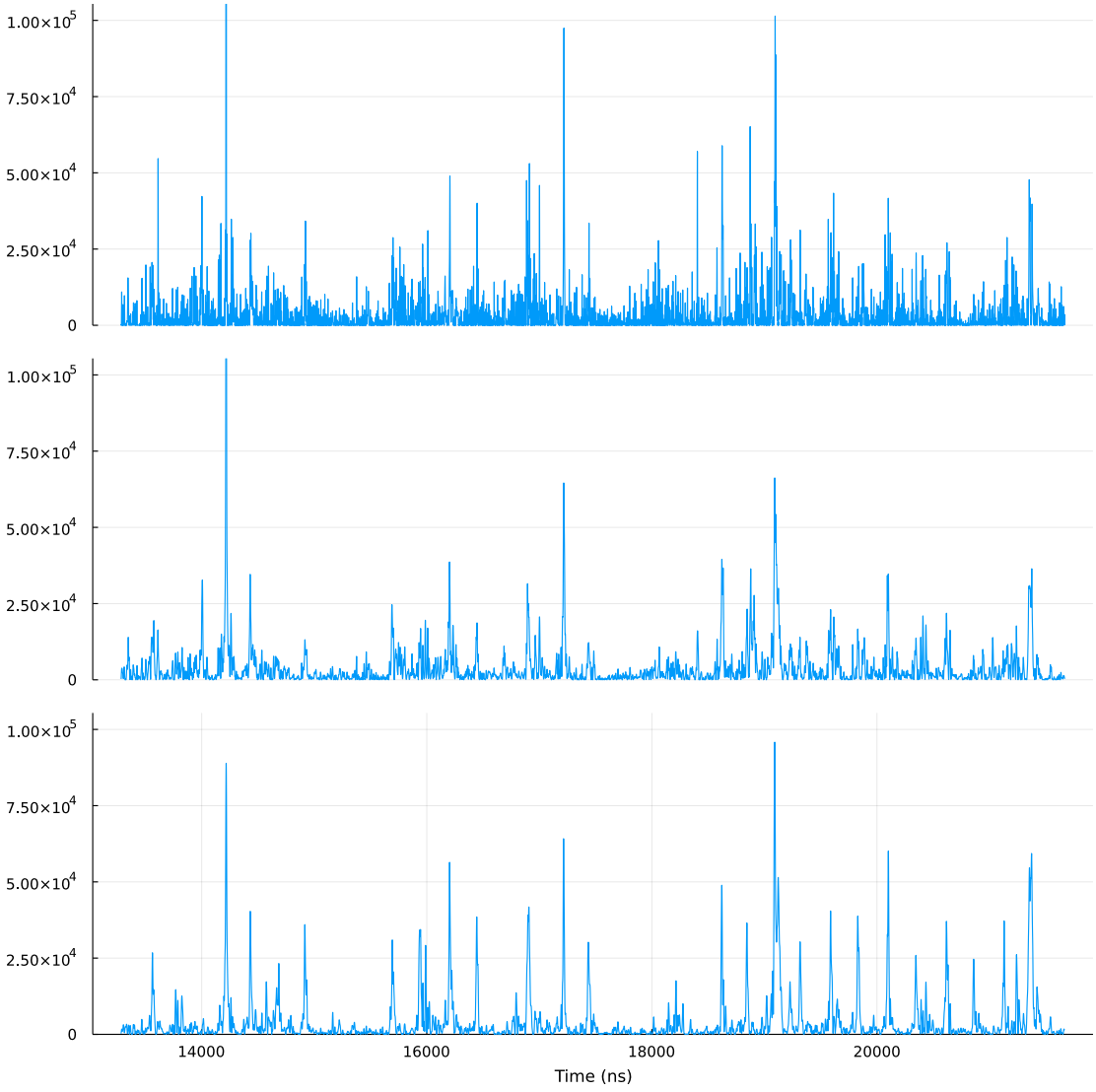


Figure 4.7: Pattern detection using the adapted filters obtained on Fig. 4.5. From top to bottom, the adapted filters obtained after the threshold-based denoising with: a null threshold, low threshold and high threshold

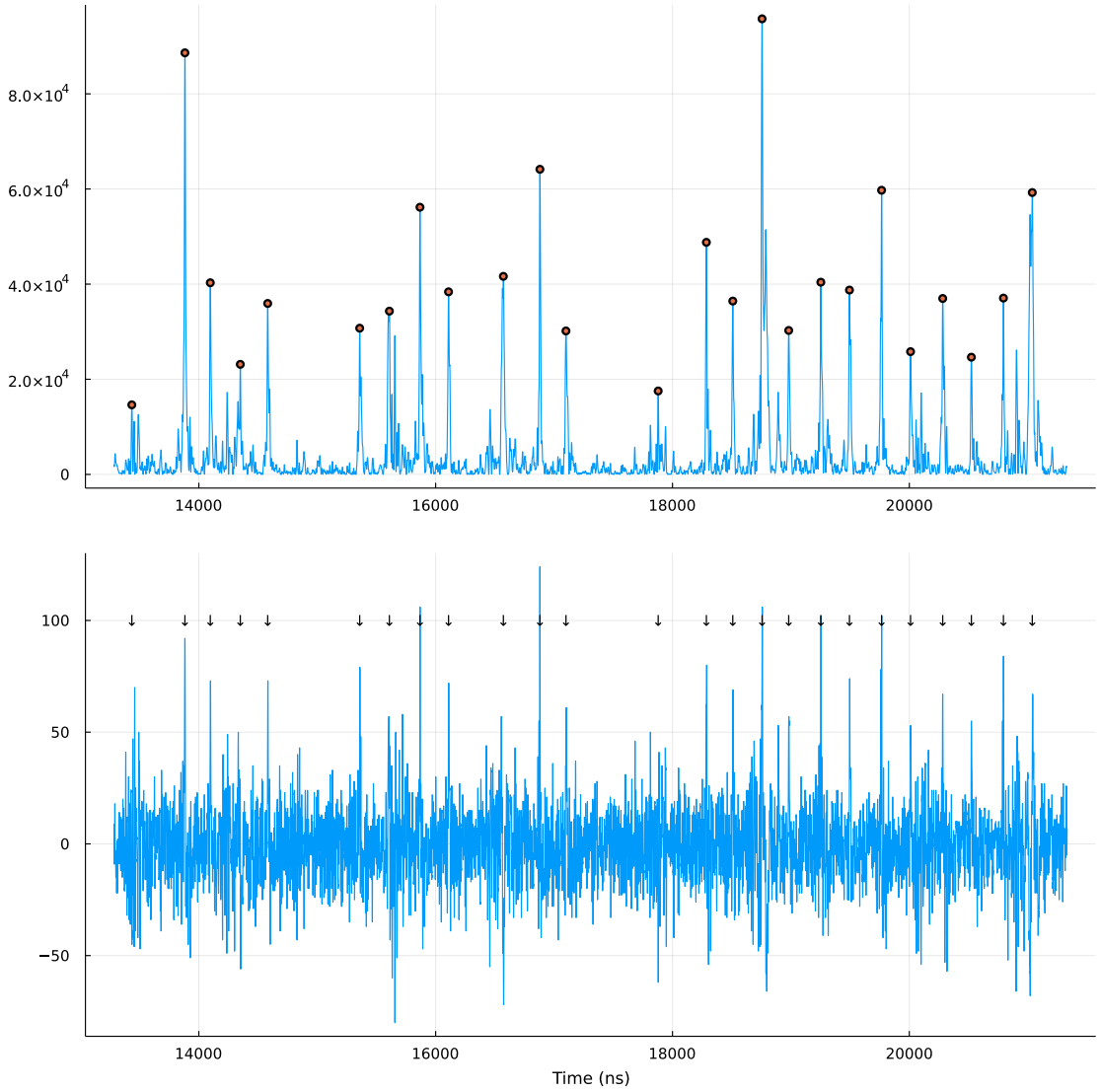


Figure 4.8: Illustration of the use of $|s * g|$, with g the resynthesized pattern obtained with a high threshold, for peak detection and pattern extraction. We represent at the bottom the corresponding signal in the time domain with arrows indicating the detected patterns in the signal. To filter peaks that are too close in time, we use a minimum time criterion between peaks.

4.3 Generalized Morse Wavelet frame Estimation

As discussed at the end of Sec. 4.2, we were able to design a simple realignment method based on a frame operator found with the help of an evaluator's knowledge. In this section we study the learning of an adapted frame directly from a set of extracted patterns. We adopt a maximum likelihood approach for estimating a frame operator that best explains in the statistical sense the patterns encountered in a given dataset of side-channel signals. In particular, we choose to work in the superfamily of Generalized Morse Wavelets presented in Sec. 1.4.4 to build frames. This way, we answer the need to embed prior information on the time-frequency properties of signals while keeping some flexibility in the optimisation and estimation of the frame.

To derive such a frame, we suppose a multivariate Gaussian prior distribution on signals and optimise a maximum likelihood loss through gradient descent. Ultimately, we aim with this work to facilitate the use of wavelet frames in larger statistical models with optimisation methods based on gradient descent.

4.3.1 Analysis in a Frame of Generalized Morse Wavelets

4.3.1.1 Problem Formulation

We wish to extract from a noisy signal y a signal of interest y_* which has been generated with coefficients x_* via a frame $\{\psi_\xi\}_{\xi\in\Xi_*}$ with index set Ξ_*. In the context of Sec. 4.2, the coefficients x_* play the role of the coefficients that have been carefully selected by the evaluator, but in our situation we suppose that we do not have access to those. We suppose that our pattern writes:

$$y = y_* + e_s = \Psi_{\Xi_*}^\dagger x_* + e_s \ , \tag{4.13}$$

where e_s is a statistical noise and $\Psi_\Xi^\dagger$ is a matrix with a frame of Generalized Morse Wavelets as column entries and with $\cdot^\dagger$ the adjoint operator for matrices.

Let Ξ be the estimated index set assumed to be of same size as the true index set Ξ_*, Ψ_Ξ the corresponding operator, and x some coefficients, equation (4.13) can be rewritten as:

$$y = \Psi_\Xi^\dagger x + e_s \ , \tag{4.14}$$

where $\Psi_\Xi^\dagger = \Psi_{\Xi_*}^\dagger T$ and $x = T^{-1}x_*$, with T an invertible transformation.

We recognize a factor analysis problem where our goal is to estimate both the set of wavelet parameters Ξ that defines the frame operator and the coefficients x (in comparison with a regression analysis where only the coefficients x are unknown).

Additionaly, we remark that the problem can be formulated either in the time

domain or in the frequency domain. Since the Fourier Transform is an isometry, i.e. by Parseval theorem the scalar products are conserved with Fourier Transform, the properties of our operator Ψ with wavelets in the time domain $\{\psi_\xi\}_{\xi\in\Xi}$ can be transferred into the frequency domain by redefining the operator with a frame of wavelets in the frequency domain $\{\widehat{\psi}_\xi\}_{\xi\in\Xi}$. In the following we choose to use the frame operator in the frequency domain as the generalized Morse Wavelets are analytically given in the frequency domain. In comparison with driving the estimation in the time domain, it will saves inverse Fourier Transforms during the gradient descentthat will be required to transpose back the updated frame in the time domain. Consequently, in our experiments, we will work with signals in the Fourier domain.

We recall here the expression for the Generalized Morse Wavelet that we will use for building the frame. For $\xi = (a, u, \beta, \gamma)$, the wavelet in the Fourier domain is expressed:

$$\widehat{\psi}_{a,u;\beta,\gamma} = \sqrt{a} c_{\beta,\gamma} (aw)^\beta e^{-(aw)^\gamma} e^{-iwu}\,. \tag{4.15}$$

with $c^2_{\beta,\gamma} = \pi\gamma\ 2^r/\Gamma(r)$ and $r = (2\beta+1)/\gamma$.

Model To answer our problem, we adopt a probabilistic approach and reformulate (4.14) as a factor analysis problem [101, 110] with stochastic variables:

$$Y = \Psi_\Xi^\dagger X + E + \mu\,. \tag{4.16}$$

Where we assume the following prior distributions for the random coefficients X, the random noise E and patterns Y:

$$X \sim \mathcal{CN}(0, \Sigma_x) \tag{4.17}$$

$$E|\Lambda \sim \mathcal{CN}(0, \Lambda) \tag{4.18}$$

$$Y|\mu, \Xi, \Lambda, \Sigma_x \sim \mathcal{CN}(\mu, \Sigma_y), \qquad \Sigma_y = \Psi_\Xi^\dagger \Sigma_x \Psi_\Xi + \Lambda \tag{4.19}$$

with $\mathcal{CN}$ the complex multivariate distribution introduced in Sec. 2.4. In the following, we reserve the notation Σ_x for the covariance of the m-dimensional coefficients X and note Σ_y for the covariance of random n-dimensional signals Y.

The goal is to estimate the unknown variables of the model (4.16) are $\mu \in \mathbb{C}^n, \Lambda \in \mathbb{C}^{n\times n}, \Sigma_x \in \mathbb{C}^{m\times m}$ and $\Xi \subset \mathbb{R}^4$. To do so, we derive in the next section a maximum likelihood approach to estimate the parameters.

4.3.2 Maximum Likelihood estimation

We present now the method to learn the frame of Generalized Morse Wavelets and the remaining parameters of the statistical model (4.16). We propose to optimise a likelihood loss through a gradient descent algorithm whose derivatives according to the parameters are given.

4.3.2.1 The Likelihood Loss

Given a set of data $D = \{y_1, \dots, y_L\}$ of size L, a point estimation of the parameters of the model can be found by optimising the following likelihood loss:

$$\mathcal{L}_{\mathrm{MLE}}(\mu, \Xi, \Lambda, \Sigma_x) = -\log \prod_{i=1}^{L} p(y_i|\mu, \Xi, \Lambda, \Sigma_x) \tag{4.20}$$

where we have noted $p(y_i|\mu, \Xi, \Lambda, \Sigma_x) = p(y_i|\mu, \Sigma_y)$ to emphasize that the distribution of Y actually depends on Ξ and Λ through Σ_y. We search for $\mu^*, \Xi^*, \Lambda^*, \Sigma_x^*$ minimizing $\mathcal{L}_{\mathrm{MLE}}$:

$$(\mu^*, \Xi^*, \Lambda^*, \Sigma_x^*) = \underset{\mu, \Xi, \Lambda, \Sigma_x}{\arg\min}\, \mathcal{L}_{\mathrm{MLE}}(\mu, \Xi, \Lambda, \Sigma_x) \tag{4.21}$$

4.3.2.2 Derivatives

In order to minimize (4.21) we study its derivatives. Using formula (2.26) for n-multivariate Gaussian distribution, we develop (4.20) to get:

$$\begin{aligned}\frac{1}{L}\mathcal{L}_{\mathrm{MLE}}(\mu, \Xi, \Lambda, \Sigma_x) = n \log \pi + \log\det \Sigma_y \\ + \frac{1}{L}\sum_{j=1}^{L} (y_j - \mu)^\dagger \, {\Sigma_y}^{-1} \, (y_j - \mu)\end{aligned} \tag{4.22}$$

recalling that $\cdot^\dagger$ is the conjugate transpose and that $\Sigma_y = \Psi_\Xi^\dagger \Sigma_x \Psi_\Xi + \Lambda$.

Using matrix calculus identities we get the differential form [76, Eq. (15.60)]:

$$\begin{aligned}\frac{1}{L} d\mathcal{L}_{\mathrm{MLE}} = \operatorname{tr}\left({\Sigma_y}^{-1} d\Sigma_y\right) \\ - \frac{1}{L}\operatorname{tr}\left({\Sigma_y}^{-1}\left(\sum_{j=1}^{L}(y_j - \mu)(y_j - \mu)^\dagger\right){\Sigma_y}^{-1} d\Sigma_y\right) \\ + 2\Re\left((\mu - \mu_e)^\dagger {\Sigma_y}^{-1} d\mu\right),\end{aligned} \tag{4.23}$$

with $\mu_e = \frac{1}{L}\sum_{j=1}^{L} y_j$ the empirical mean, and $\operatorname{tr}()$ denotes the trace operator. This

is rewritten after some manipulations:

$$\begin{aligned}\frac{1}{L}d\mathcal{L}_{\text{MLE}} &= \operatorname{tr}\left(h[\Sigma_y, S]d\Sigma_y\right) \\ &\quad + 2\Re\left((\mu-\mu_e)^\dagger {\Sigma_y}^{-1} d\mu\right)\end{aligned} \tag{4.24}$$

where we have introduced the notations

$$h[A,B] = A^{-1}(A-B)A^{-1}\ ,\ S = \frac{1}{L}\sum_{j=1}^{L}(y_j-\mu)(y_j-\mu)^\dagger$$

Clearly, $d\mathcal{L}_{\text{MLE}}/d\mu$ vanishes for $\mu = \mu_e$ the empirical mean, and $d\mathcal{L}_{\text{MLE}}/d\Sigma_y$ for $\Sigma_y = S$ the (biased) covariance matrix. Remark that the data in S is not properly centered with the empirical mean μ_e thus the stationary point for Σ_y will vary with μ. We also note that the optimisation becomes ill-posed when Σ_y is not full rank definite since we can find an infinite number of stationary points by choosing an update $\tilde{\mu}$ making $\mu_e - \tilde{\mu}$ orthogonal to the eigenspace of Σ_y. In practice, we make Σ_y strictly positive by choosing a positive diagonal matrix for Λ. To avoid dealing with degenerate complex Gaussian distributions we aim Σ_x to always stay strictly positive definite, this way it is convenient to search for a lower complex triangular matrix with strictly positive diagonal values C such that the Cholesky decomposition of Σ_x is $\Sigma_x = CC^\dagger$. We note $\Psi_\Xi^\triangleleft = C^\dagger \Psi_\Xi$.

The differential form of Σ_y now gives:

$$d\Sigma_y = d\Lambda + d\Psi_\Xi^{\triangleleft\dagger}\,\Psi_\Xi^\triangleleft + \Psi_\Xi^{\triangleleft\dagger}\,d\Psi_\Xi^\triangleleft\,, \tag{4.25}$$

with $d\Psi_\Xi^\triangleleft = dC^\dagger\,\Psi_\Xi + C^\dagger\,d\Psi_\Xi$.

Using (4.25) in (4.24) and with the previous notations, we find the following Jacobians for μ, Ψ, Λ, C:

$$\nabla_\mu \mathcal{L}_{\text{MLE}} = (\mu-\mu_e)^\dagger \Sigma_y^{-1} \tag{4.26}$$

$$\nabla_{\Psi_\Xi} \mathcal{L}_{\text{MLE}} = \operatorname{vec}\left(C\ h[\Sigma_y, S]\Psi_\Xi^\triangleleft\right)^\dagger \tag{4.27}$$

$$\nabla_\Lambda \mathcal{L}_{\text{MLE}} = \operatorname{vec}\left(h[\Sigma_y, S]\right)^\dagger \tag{4.28}$$

$$\nabla_C \mathcal{L}_{\text{MLE}} = \operatorname{vec}\left(\Psi_\Xi^\dagger\ h[\Sigma_y, S]\Psi_\Xi^\triangleleft\right)^\dagger \tag{4.29}$$

where $\operatorname{vec}(\cdot)$ is the vectorize operator, i.e. $\operatorname{vec}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = [a\ c\ b\ d]^T \in \mathbb{R}^4$. We used the differentiation conventions of [51], i.e. if f is a real valued function of complex matrix arguments Z and $\overline{Z}$ ($\overline{Z}$ being the complex conjugate of Z) then its

derivatives are defined by

$$\nabla_Z f = \frac{\partial \mathrm{vec}\left(f(Z,\overline{Z})\right)}{\partial \mathrm{vec}\,(Z)^T} = \begin{bmatrix} \frac{\partial f_{1,1}}{\partial Z_{1,1}} & \cdots & \frac{\partial f_{1,1}}{\partial Z_{n,1}} & \cdots & \frac{\partial f_{1,1}}{\partial Z_{n,p}} \\ & & \vdots & & \\ \frac{\partial f_{m,1}}{\partial Z_{1,1}} & \cdots & \frac{\partial f_{m,1}}{\partial Z_{n,1}} & \cdots & \frac{\partial f_{m,1}}{\partial Z_{n,p}} \\ & & \vdots & & \\ \frac{\partial f_{m,k}}{\partial Z_{1,1}} & \cdots & \frac{\partial f_{m,k}}{\partial Z_{n,1}} & \cdots & \frac{\partial f_{m,k}}{\partial Z_{n,p}} \end{bmatrix}$$

with $f(Z,\overline{Z}) \in \mathbb{R}^{m\times k}$ and $Z \in \mathbb{C}^{n\times p}$.

Additionally, given a differential form $df(Z,\overline{Z}) = \mathrm{tr}\left(A^T dZ + B^T d\overline{Z}\right)$ with arbitrary matrices A, B, we have

$$\nabla_Z f = \mathrm{vec}\,(A)^T \tag{4.30}$$

$$\nabla_{\overline{Z}} f = \mathrm{vec}\,(B)^T . \tag{4.31}$$

Also, if f is real-valued, we have: $\nabla_Z f = \overline{(\nabla_{\overline{Z}} f)}$ [51, Theorem 3]; thus in our case we have for example $\nabla_{\Psi_\Xi}\mathcal{L}_{\mathrm{MLE}} = \overline{\left(\nabla_{\overline{\Psi_\Xi}}\mathcal{L}_{\mathrm{MLE}}\right)}$. In consequence, we do not write here the derivatives for the conjugate variables.

We are in particular interested in the derivatives according to the wavelet parameters $\xi \in \Xi$. Using (4.27) and the chain rule [51, Theorem 1] we find:

$$\nabla_{\xi_i}\mathcal{L}_{\mathrm{MLE}} = 2\Re\left(\nabla_{\Psi_\Xi}\mathcal{L}_{\mathrm{MLE}}\,\nabla_{\widehat{\psi}_\xi}\Psi_\Xi\,\nabla_\xi\widehat{\psi}_\xi\right) \tag{4.32}$$

where

$$\nabla_{\widehat{\psi}_{\xi_i}}\Psi_\Xi = \partial\mathrm{vec}\left(\begin{bmatrix}\widehat{\psi}_{\xi_1}\\ \vdots \\ \widehat{\psi}_{\xi_K}\end{bmatrix}\right) \Big/ \partial\mathrm{vec}\left(\widehat{\psi}_{\xi_i}\right)^T$$

with the property that $\nabla_{\widehat{\psi}_{\xi_i}}\Psi_\Xi{}^T\mathrm{vec}\,(d\Psi_\Xi) = \mathrm{vec}\left(d\widehat{\psi}_{\xi_i}\right)$. And $\nabla_\xi\widehat{\psi}_\xi = \left[\frac{\partial\widehat{\psi}_\xi}{\partial a}\ \frac{\partial\widehat{\psi}_\xi}{\partial u}\ \frac{\partial\widehat{\psi}_\xi}{\partial\beta}\ \frac{\partial\widehat{\psi}_\xi}{\partial\gamma}\right]$ is computed by deriving (4.15) according to a, u, β and γ (given in Appendix A.1).

Now that we have all the derivates according to each parameters in equations (4.26), (4.32), (4.28) and (4.29), we optimise the likelihood loss (4.20) via gradient descent.

4.3.2.3 Overview of the estimation method proposed

We give below an overview of the method for estimating the parameters of the statistical model proposed in (4.16). From now on, the statistical model and the method to optimise it is noted GMW-MLE.

1. Given a dataset of patterns $\{y_i\}_i$, compute the empirical mean μ_e, the empirical covariance S, and initiliaze the parameters of the model (4.16), i.e. μ, Ξ, Λ

and Σ_x (more precisely for the latter its Cholesky factor C).

2. Optimise the likelihood loss (4.20) by updating through gradient descent the parameters Ξ, Λ and Σ_x using respectively the expressions for the jacobians (4.32), (4.28) and (4.29). μ can directly be set to μ_e as $\nabla_\mu \mathcal{L}_{\text{MLE}}\big|_{\mu=\mu_e} = 0$. The gradient updates are iterated until convergence.

3. Obtain the estimates μ, Ξ, Λ and Σ_x. As an important and direct side result, we also get an estimate of the covariance of the patterns Σ_y expressed

$$\Sigma_y = \Psi_\Xi^\dagger \Sigma_x \Psi_\Xi + \Lambda \,,$$

where $\Psi_\Xi^\dagger \Sigma_x \Psi_\Xi$ is the covariance of the signal of interest we wish to analyze, and Λ is the covariance of the noise. We discuss further on the form obtained for Σ_y with our model in Sec. 4.3.2.5.

4.3.2.4 Parameters Initialization and Learning

The parameters need to be properly initialized to ensure fast convergence and stability. For Ξ, we always start with a stationary frame of Generalized Morse Wavelets logarithmically spaced in frequency and linearly spaced in time, with parameters β and γ initially fixed to $(1, 3)$. The time-frequency domain covered by the frame can be chosen such that it covers the domain of interest, it is particularly useful if it is known that some frequency bands do not carry our signal of interest.

We restrict Λ to be strictly diagonal positive and C to be triangular inferior with positive diagonals.

During gradient updates, it is a common problem to keep parameters into a predefined subspace, e.g. a noisy update of C can make some of its diagonal elements negative or noisy updates of β, γ can make them negative for which no Generalized Morse Wavelets are defined. In consequence the learning is regularized by using softplus rectifier functions [45] $\text{SP} : x \mapsto \log(1 + e^x)$ to keep parameters into positive domains when needed. It amounts to append the Jacobian of SP into the differentiation chain rules.

In practice, we perform gradient updates through pullback differentiation [42]. It is programmed in Julia using Zygote [55] as automatic differentiation tool, with the machine learning library Flux [56] and with custom pullback rules written with ChainRules [119].

4.3.2.5 Further optimising the learned frame for dimension reduction

It has been proven efficient to reduce the size of the data before performing side-channel attacks [24], and more generally in machine learning it is always welcome to

reduce the dimensionality of the data beforehand. To do so, with the estimates of the parameters of the model (4.16), we can use the frame as a dimension reduction tool. As discussed in the end of Sec. 4.3.2.3, we also get an estimate of the covariance of the pattern in function of the parameters of the model. With the SVD of $\Sigma_x = U_x D_x U_x^\dagger$ with $D \in \mathbb{R}^{m\times m}$ a positive diagonal matrix containing the eigenvalues of Σ_x sorted in decreasing order along the diagonal, and $U_x \in \mathbb{C}^{m\times m}$ a unitary matrix with the eigenvectors of Σ_x, we have:

$$\Sigma_y = \Psi_\Xi^\dagger \Sigma_x \Psi_\Xi + \Lambda \tag{4.33}$$

$$= \Psi_\Xi^\dagger U_x D_x U_x^\dagger \Psi_\Xi + \Lambda \tag{4.34}$$

$$= \Phi^\dagger D_x \Phi + \Lambda \tag{4.35}$$

with $\Phi = U_x^\dagger \Psi_\Xi$ a new frame operator which is function of the estimates Σ_x and Ξ. The size of the new frame Φ can be further reduced by selecting the eigenvectors of Σ_x with high eigenvalues (as we would do with a PCA but here on the coefficients). Let p the number of eigenvectors selected, we can write $U_x = [U_{x,i}\, U_{x,e}]$ with $U_{x,i} \in \mathbb{C}^{m\times p}$ the selected eigenvectors and $U_{x,e} \in \mathbb{C}^{m\times m-p}$ the remaining ones, and

$$D_x = \begin{bmatrix} D_{x,i} & 0 \\ 0 & D_{x,e} \end{bmatrix}$$

with the diagonal matrices containing the corresponding eigenvalues $D_{x,i} \in \mathbb{R}^{+p\times p}$ and $D_{x,e} \in \mathbb{R}^{+m-p\times m-p}$. Let $\Phi_i = U_{x,i}^\dagger \Psi_\Xi$ and $\Phi_e = U_{x,e}^\dagger \Psi_\Xi$, we can reinterpret Σ_y as

$$\Sigma_y = \underbrace{\Phi_i^\dagger D_{x,i} \Phi_i}_{\text{Cov. signal}} + \underbrace{\Phi_e^\dagger D_{x,e} \Phi_e + \Lambda}_{\text{Cov. noise}} \tag{4.36}$$

where we put in the new covariance of the noise some part of the previous covariance of the signal of interest, i.e. $\Psi_\Xi^\dagger \Sigma_x \Psi_\Xi$ or equivalently $\Phi^\dagger D_x \Phi_\Xi$.

Now, to compress the patterns into p-dimensional vectors, we can use the dual of Φ_i:

$$\widetilde{\Phi}_i = \Phi_i^\dagger (\Phi_i \Phi_i^\dagger)^{-1}\,. \tag{4.37}$$

The Principal Component Analysis (PCA) method uses the covariance of the patterns to identify the subspace of the signal of interest, here the SVD is performed on the covariance of the coefficients and the selected eigenvectors are projected back into the signal space through the frame. In comparison with the eigenvectors acquired by a PCA, the components of Φ are restricted to a space of signal with the time-frequency properties of the learned frame of Generalized Morse Wavelets.

4.3.3 Information Retrieval in Side-Channel Signals

Our goal in this section will be to use the previous model to analyze patterns in side-channel signals. The model (noted GMW-MLE) is used as an analysis and dimension reduction tool to facilitate side-channel attacks. It is also compared with other methods of compression such as the Principal Components Analysis (PCA) and UMAP [88].

4.3.3.1 Dataset and Experiment

We use $10,000$ electromagnetic signals from the JIT dataset (see Sec. 4.1.1), we recall that these signals were acquired during an AES encryption algorithm. The sensitive variable Z targeted is the output of the `SubBytes()` operation of the second round. We recall, see Sec. 3.3.3, that this step outputs a sensitive variable computed with a plaintext e and a key k^*. To assess the security of the device, we want to estimate z knowing the plaintexts e in order to get information about the cipher key k^*. We use the attack method presented in Sec. 3.3.3, we use $8,000$ traces for the train set and $2,000$ for the test set.

We display on Fig. 4.10 an example of a signal representing the first three rounds of AES. We use the automatic detection of patterns of Sec. 4.2.4 with a wavelet as the adapted filter to extract a set of patterns from signal in the JIT dataset. In particular, we remark that the wavelet corresponding to the frequency band around $5.5\text{e}-4$ matches with many patterns in the continuous wavelet transform in Fig. 4.10. From the filtered signals, we look at their energy through time and extract 16 patterns of size 512 samples around the peaks.

We train our model GMW-MLE on this set of patterns to learn a frame of 25 wavelets along with the covariance of coefficients Σ_x. We follow the procedure in 4.3.2.4 to initialize the parameters. We then compute the pseudo-inverse (4.37) and compress our patterns into a set of wavelet coefficients (called features). We compare our model with other dimension reduction techniques such as the PCA [60] and a more recent technique UMAP [88] which is based on manifold learning techniques. These methods are trained on all individual patterns and used to compute features that are concatenated back to reform compressed version of our original signals. The dimension of reduction is fixed to 10 for all experiments. The 16 extracted patterns from each signals are compressed into 10-dimensional feature vectors, thus leading to compressed signals of size 160.

Finally, we use this new dataset of features to perform template attacks [21], whose learning procedure, named Quadratic Discriminant Analysis (QDA), is presented in Sec. 2.5.1. This step amounts to train a QDA model on compressed signals to get probability predictions on keys $p(k|\mathrm{DR}(y)) = p(k|x)$ for a key hypothesis k,

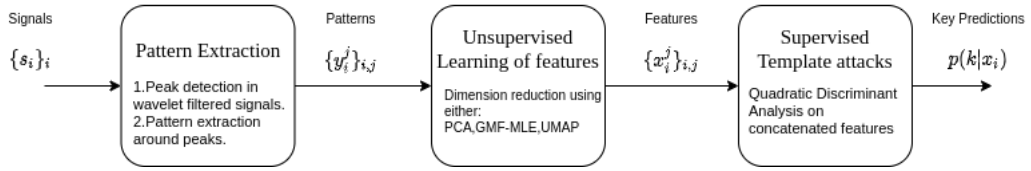


Figure 4.9: Diagram showing the consecutive steps to carry out the side-channel attack. The technique for extracting the patterns is introduced in Sec. 4.2.4, the reduction of dimension with the GMW-MLE method corresponds to the method in Sec. 4.3.2.5, the supervised attack is performed using the QDA presented in Sec. 2.5.1.

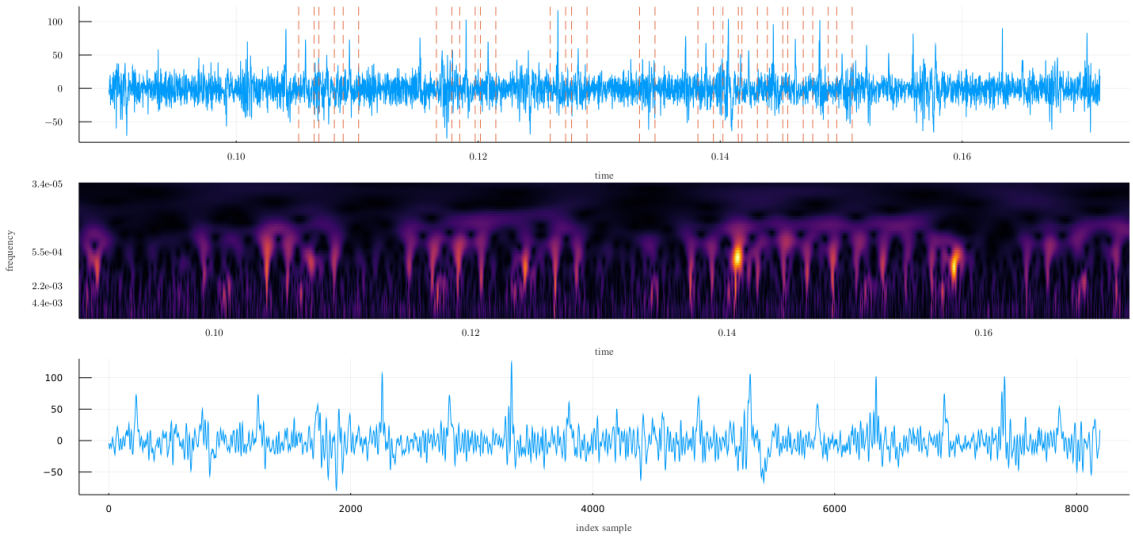


Figure 4.10: Top: One EM signal from JIT. Middle: Continuous Wavelet Transform with a basis of Generalized Morse Wavelets ($\beta = 2, \gamma = 3$), the y-axis units are the normalized central frequency of each wavelet in log scale. Bottom: concatenated extracted patterns from the top signal. 16 patterns are displayed and originally lay between couples of vertical lines.

a pattern y and its vector of concatenated features x and where DR is the compression technique used (PCA,GMW-MLE or UMAP). The performance of attacks are evaluated using the Guessing Entropy metric [83], see Sec. 3.3.3. To properly evaluate the results of the models, we use a 10-fold Monte Carlo cross-validation, i.e. models are re-trained and evaluated on randomly drawn train and test sets 10 times.

The full experiment is resumed in Fig. 4.9.

4.3.3.2 Results

In Fig. 4.11, we show the components learned by the PCA and our method. We notice that both methods led to components that are mainly centered in time. We also illustrate on Fig. 4.13 the time-frequency properties of the frame after conver-

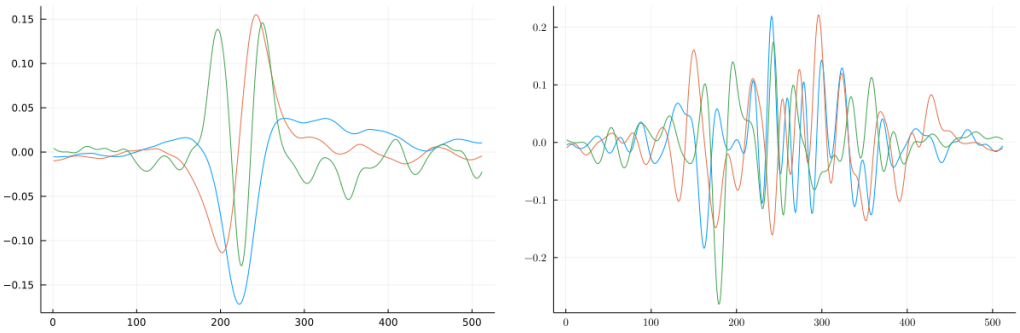


Figure 4.11: Visualisation of the first three main components of the PCA (left), of Φ in (4.37) for GMW-MLE (right). For GMW-MLE we took the real part of the components.

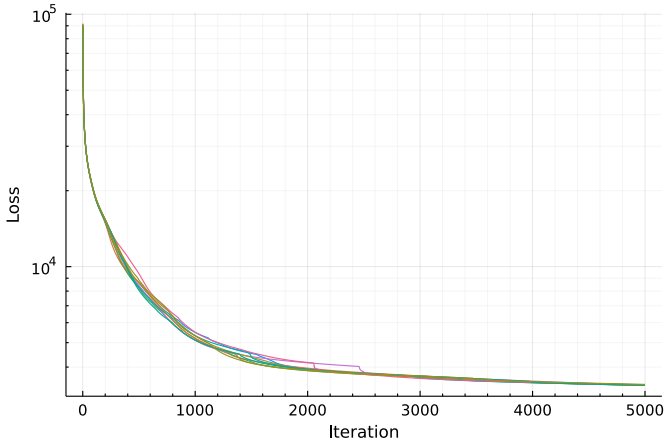


Figure 4.12: Evolution of the log likelihood loss during the GMW-MLE training of Sec. 4.3.2.3.

gence and on Fig. 4.11 the Fourier spectrum of some PCA components (bottom) and wavelets from the frame (top). We display in Fig. 4.12 the loss during the training the GMW-MLE method.

We represent in Fig. 4.15 some plots of the features learned by the different methods. We choose the first three dimensions of features and plot each dimension versus the other. We want to see if some patterns can be isolated from the others by looking at their features. Only the UMAP dimension reduction method leads to a representation that could help at differentiating patterns. The PCA and the GMW-MLE methods do not help to visually identify group of points. The results of attacks with each dimension reduction technique are displayed Fig. 4.16. GMW-MLE and UMAP lead to the fastest decreasing guessing entropy. GMW-MLE can be seen as a PCA with a restriction on the time-frequency properties of its components. This restriction in our case allows to focus the learning on a subspace that is most likely to contain the signal of interest. While UMAP presents very interesting properties and provides good results here, its main drawback is that it is not straightforward

to inverse the features and regenerate patterns, and it is difficult to integrate into statistical models.

Conclusion and Perspectives

In this section, we presented a novel technique for estimating a frame of Generalized Morse Wavelets from a given dataset of signals. We formulated our problem as a statistical model for which a likelihood loss is derived and whose derivatives according to the parameters are calculated.

The model is used in the context of unsupervised dimension reduction for compressing patterns from side-channel signals and improve template attacks. More generally, we think that this model could be used as a potential building block for other statistical models such as state-space models, MLP and CNN. In particular, a regularisation constraint can be added to the loss to restrict the duration or the frequency peak of the Generalized Morse Wavelets such that the learning focuses on particular time-frequency properties of the data.

We also envisage the following potential improvements:

- Since the parameters of the frame are updated in the Fourier domain, any update of the phase of a wavelet makes it loops around its time domain. We suspect that it has some impact on the learning. A thorough study is required in this direction. If we allow wavelets to "disappear" on the border of the time domain occupied by the signal, it requires to consider the alternative model:

$$y = M\Psi_{\Xi}^{\dagger}x + e_s \tag{4.38}$$

  with $M = [0_{n/2}\, I_n\, 0_{n/2-1}]$ a masking matrix of size $n \times 2n-1$, where the size of the wavelets are increased to $2n-1$ and patterns y stay of size n. The covariance of the model becomes $\Sigma_y = M\Psi_{\Xi}^{\dagger}\Sigma_x\Psi_{\Xi}M^{\dagger}+\Lambda$. The masking matrix needs to be added to the jacobians, alongside with the Fourier transform matrix[2] to pass the derivatives of the Generalized Morse wavelets into the time domain.

- Although the optimisation converge rapidly, the method remains computationally intensive in comparison with a simple PCA. To increase the convergence rate a study of the sensibility of the likelihood loss according to the wavelet parameters is required. It amounts to study the Fisher information matrix [97, Chap. 8] of (4.20). For that, some already implemented work in Julia has been initiated and the in-depth study is reserved for further work.

[2]In practice we do not have to explicitly construct the Fourier transform matrix as we would use the fast Fourier transform in the differentiation chain rules.

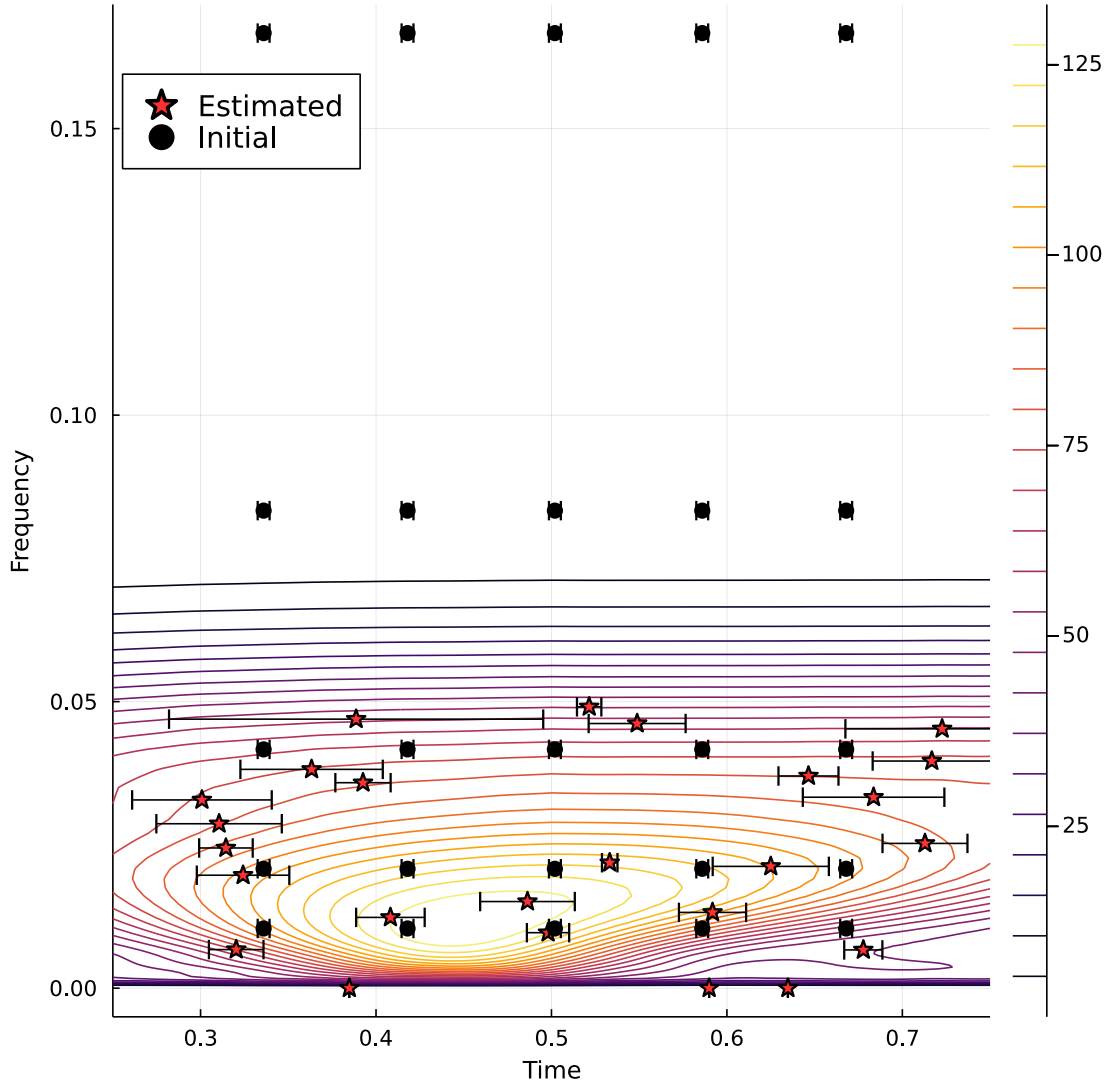
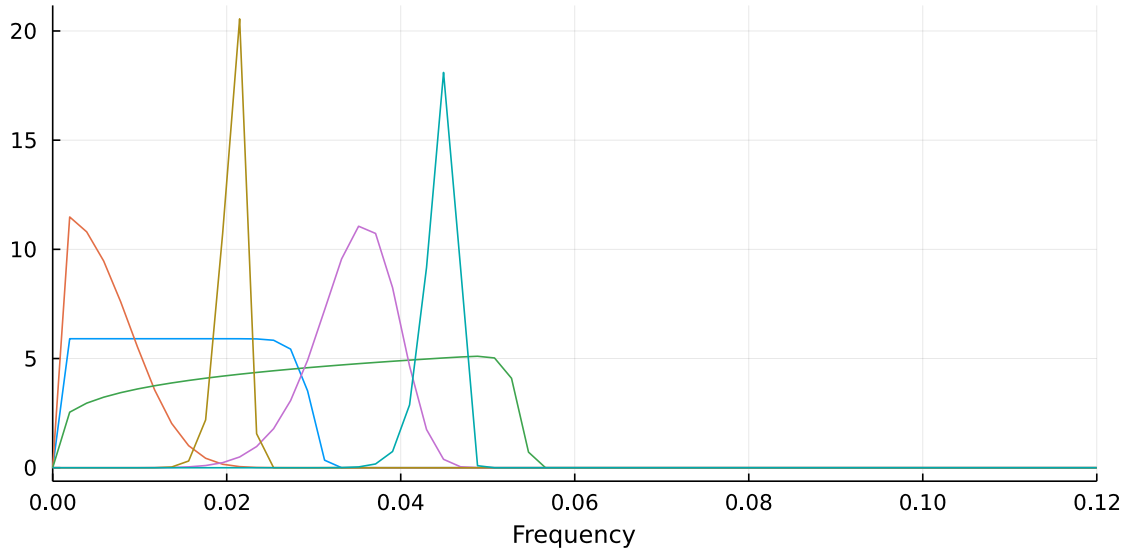
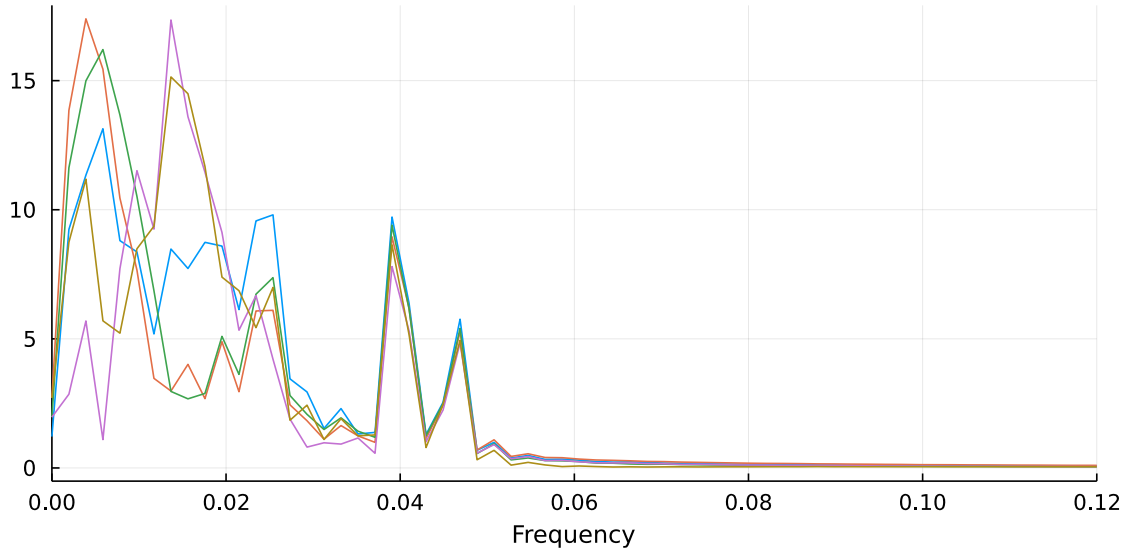


Figure 4.13: Comparison between an initial and a learned frame of Generalized Morse Wavelets (GMW). We show the contours of the average scalogram of patterns $\{y_i\}_i$. Error bars along the x-axis show the duration (1.51) of Sec. 1.4.4 of the GMW. Time and frequency axis are normalized.



(a) Ondelettes ψ_Ξ en Fourier



(b) Composantes PCA en Fourier

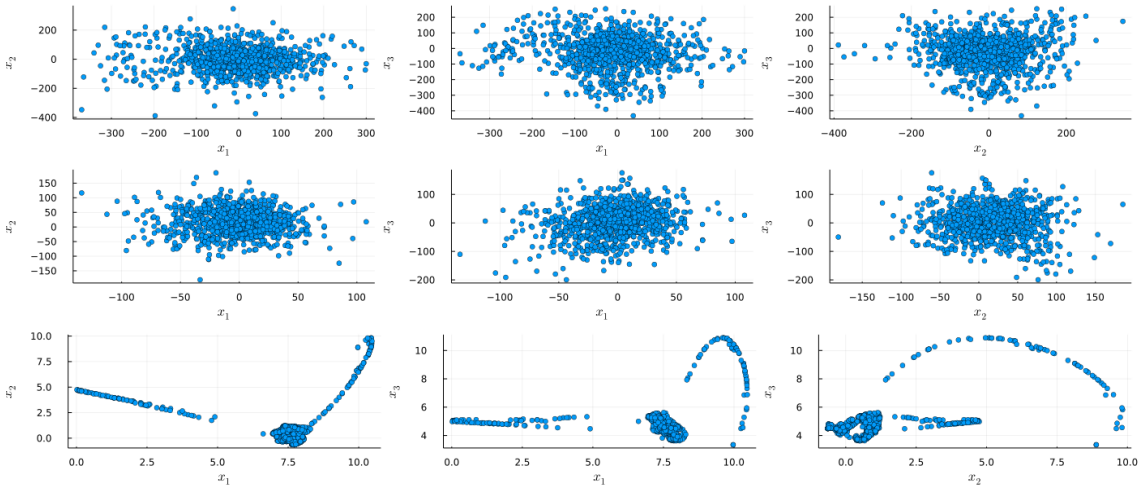


Figure 4.15: From left to right: Scatter plots of features for each of their first three components. Form Top to Bottom: PCA features, GMW-MLE features, UMAP features.

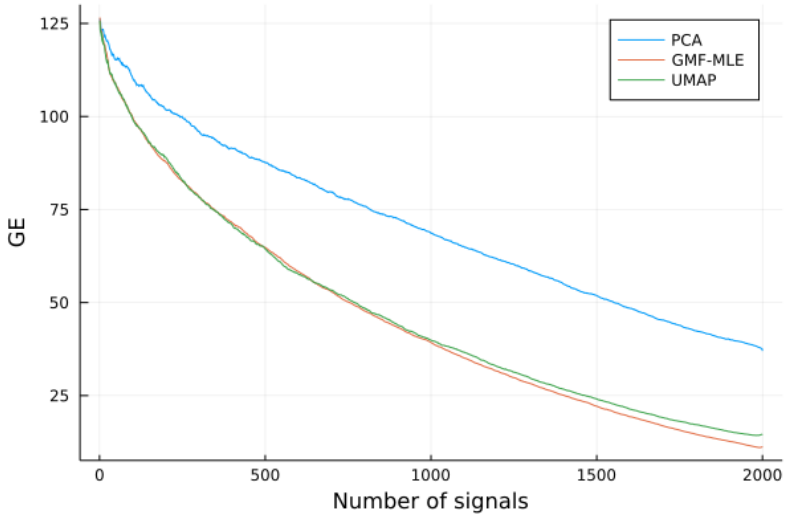


Figure 4.16: Evolution of the guessing entropy (GE) of the true key with increasing number of signals and for each compression method. The faster it decreases the better.

4.4 Wavelet Scattering Transform and Ensemble Methods for Side-Channel Analysis

In previous sections, we relied on the realignment of signals to perform side-channel attacks. Here, we explore a wavelet-based preprocessing method that do not necessarily require a realignment of signals. We study the use of the Wavelet Scattering Transform, proposed by S. Mallat in [81] to produce robust representations against time and frequency translations.

Also, we saw in Sec. 3.3.4, that the leakage model of sensitive variables is generally unknown. In the literature, we often admit a leakage model and evaluate its relevance on a dataset of signals through the evaluation of the Guessing Entropy. We propose in this section an ensemble method that combines the prediction of different leakage models to form a more robust and general model.

4.4.1 Translation invariance and stability under diffeomorphism

As explained in the beginning of this chapter, signals in side-channel analysis are generally desynchronized by jitter countermeasures. Additionally, we remark that patterns are sometimes distorted resulting in some displacements of the frequency content in the Fourier domain. Thus, to facilitate the learning of statistical models in side-channel analysis, a good representation Φs of signals s with an operator Φ should be stable to some extent against translation and deformation.

Let x_1, x_2 be two acquired signals, we say that x_1 is a *deformed* version of x_2 if there exists a diffeomorphism $\tau(t)$ such that $x_1(t) = x_2(\tau(t))$.

A practical example in SCA is given Figure 4.17 where two patterns from EM signals are plotted. Although both signals contain the same cryptographic information, we notice translations in the time and frequency domain. We presented in Sec. 1.3.4 and in particular in Fig. 1.1 the time-frequency domains covered by the STFT and wavelet transforms bases. We remark that in the case of the STFT we can adapt the size of the window to allow some robustness against time translation, while for the wavelet transform (WT) the representation will be unstable against time translation at high frequency. Alternatively, in the frequency space, WT representations are robust against small frequency shifts and STFT representations are unstable, we illustrated this in Fig. 4.18.

In [81], the problem is tackled by searching for an operator Φ that limits the distance between a signal and its deformed version, in the sense that $\Phi x \approx \Phi L_\tau x$ where L_τ denotes the deformation operator induced by the diffeomorphism τ. According to [81], the operator Φ should be designed with respect to the two following

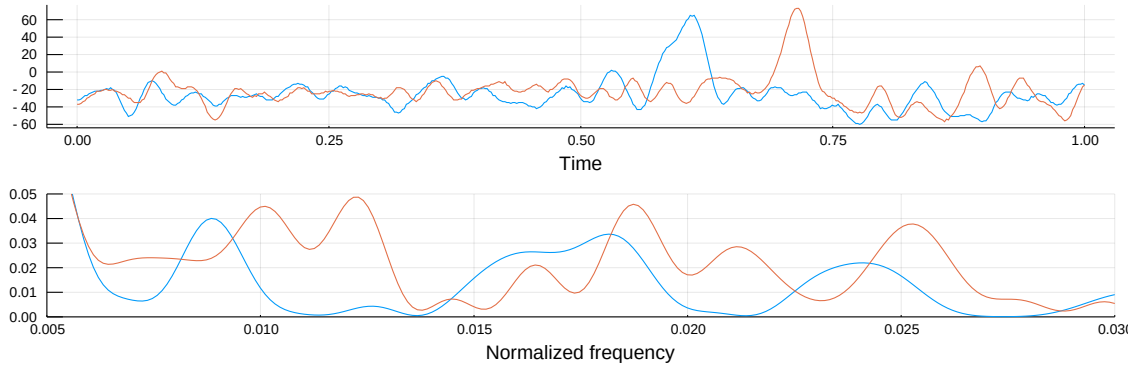


Figure 4.17: Jitter effect and deformation taken from JIT signals (see Section 4.1.1). Two temporal patterns are plotted at the top with their associated Fourier Transform at the bottom. The deformation between these patterns is characterized here by a frequency shift of some components (e.g. at frequency=0.026) in the Fourier spectrum.

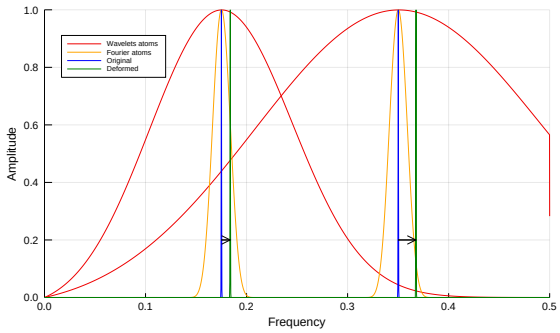


Figure 4.18: Effect of deformations in frequency for STFT and WT. We generate a modulated Gaussian pulse $x_1(t)$ at a low and high frequency along with its deformed version $x_2(t) = x_1(1.05t)$. We plot two elements of the basis of a STFT and WT basis such that it captures the original signal. We remark that under deformation the WT still capture the deformed signal.

properties:

- Φ is *translation invariant*, i.e. for $c \in \mathbb{R}$ and $L_c x(t) = x(t-c)$:

$$\Phi x = \Phi L_c x$$

- Moreover, Φ is *stable by diffeomorphism*, i.e. it is Lipschitz continuous to the action of a C^2-diffeomorphism τ. For $\tau \in \mathcal{C}^2(\mathbb{R})$, $L_\tau x(t) = x(t - \tau(t))$ and $C \in \mathbb{R}^+$:

$$\|\Phi x - \Phi L_\tau x\| \leq C\|x\|(\|\frac{\partial \tau}{\partial t}\|_\infty + \|\frac{\partial^2 \tau}{\partial t^2}\|_\infty) \tag{4.39}$$

The scattering transform proposed in [81] respects these properties and is presented hereafter.

4.4.2 The Wavelet Scattering Transform

In order to have such properties, Mallat proposes in [81, 7] cascading continuous wavelet transforms. Let s a signal and ψ a mother wavelet, we here write the continuous wavelet transform at scale a:

$$W[a]s = s * \psi_a = \int s(t) \frac{1}{\sqrt{a}} \psi \left(\frac{u-t}{a} \right) dt \tag{4.40}$$

Each convolution with a wavelet ψ_a is followed by the absolute value $|\,.\,|$ and averaged on a time domain of 2^J samples with $A_J x = x * \phi_{2^J}$, with J the maximum number of dyadic scales and ϕ_{2^J} a low pass filter. This procedure is then repeated along multiple paths of scales $p = (a_1, ..., a_m)$ with $a_i > 2^{-J}$. For a path p, we have:

$$\begin{aligned} S[p]x &= |||x * \psi_{a_1}| * \psi_{a_2}|... * \psi_{a_m}| * \phi_{2^J} \\ &= |W[a_m] \ ... \ |W[a_2] \ |W[a_1]x||| * \phi_{2^J} \\ &= A_J|W[a_m] \ ... \ |W[a_2] \ |W[a_1]x||| \\ &= A_J U[a_m] \ ... \ U[a_2]U[a_1]x \end{aligned} \tag{4.41}$$

with $U[a]x = |W[a]x| = |x * \psi_a|$ and $A_J x = x * \phi_{2^J}$. In practice the scattering transform is calculated on a path subset $\Omega_{J,m}$ for which a maximum length m of paths $p \in \Omega_{J,m}$ is set and with scales $a > 2^{-J}$. We illustrate Fig. 4.19 the scattering network.

While wavelet transforms provide stability under the action of small frequency translation, the nonlinear operation and the integration over time guarantees translation invariance. Cascading wavelet transforms allows recovering high frequencies lost after averaging the absolute value of the continuous wavelets transforms of lower levels. We refer to [81] for further details.

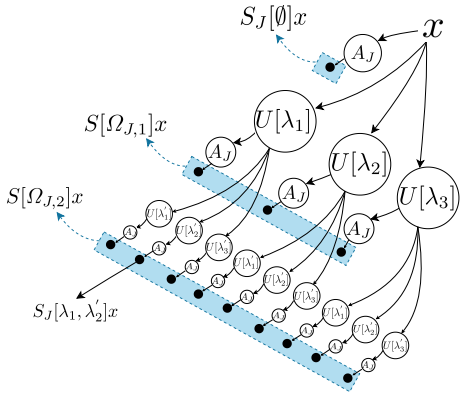


Figure 4.19: A two-level wavelet scattering transform. The scale here is noted λ instead of a.

We can adapt the number of scales depending on the spectral richness of signals, we have $a_j = 2^{-j/Q}, 0 \leq j < JQ$ where J is the number of dyadic scales and Q defines the number of inter-scales. The whole transform is characterized by three parameters: J, Q and the number of levels $m \in \{1, 2\}$.

To tune the parameters of the scattering transform, we propose the following rules of thumb: choose J proportionally to the amount of translation (i.e. jitter) present in signals and Q according to the desired discrimination in frequency. If J is set too high, a second level $m=2$ is required to retrieve the information lost by the low-pass filter.

The scattering transform is computed using the python implementation of [8]. In our case, Morlet wavelets are used for the continuous wavelet transform, we recall here their definition:

$$\psi(t) = c_\sigma e^{-\frac{1}{2}\left(\frac{t}{\sigma}\right)^2}(e^{iwt} - b_0) \tag{4.42}$$

with b_0 such that ψ is made admissible, see Sec. 1.3.2, c_σ is a normalisation constant such that $\|\psi\| = 1$. σ and w are fixed parameters chosen such that the mother wavelet covers the highest frequency band desired.

Before presenting some results on the use of the scattering transform, we present our combination procedure for approximating the leakage model.

4.4.3 A Combination Procedure for Ensemble Methods in SCA

For the task of classification in SCA, one label is usually considered to provide an estimation of a sensitive variable Z. Here we focus on the space of targeted class values with multiple classifiers trained on L different labelings $\{C_l\}_{1 \leq l \leq L}$, each labeling giving clues on the sensitive variable z with a probability $p(Z = z | C_l = c_l)$.

Classification of the sensitive variables considered in SCA lends itself well to

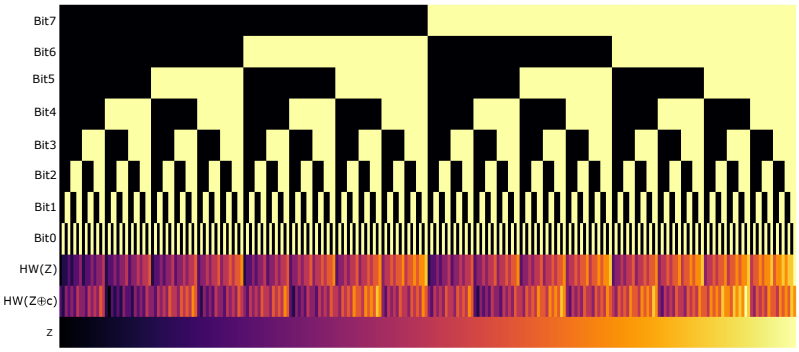


Figure 4.20: Example of partition functions for the approximation of the leakage model. The first eight rows are the partitioning according to single bit values. The second last-one is the Hamming weight partitioning and finally we have the identity at the last row. The x-axis is the value of the byte from 0 to 255.

partition our target space $Z \in \boldsymbol{Z}$ in complementary regions. We denote β_l the partition function that associates each z to a label $c_l \in \mathcal{C}_l$, such that $\beta_l(z) = c_l$ and $\boldsymbol{\beta}(z) = (c_1, ..., c_L) = c \in \mathcal{C}$. For example, if z is the byte 0x12 and $\boldsymbol{\beta}$ is composed of labelings respectively: identity over $\mathbb{Z}_8$, Hamming weight and the first big-endian bit value; then $\boldsymbol{\beta}(0\text{x}12) = (0\text{x}12, 2, 0)$. We represent Fig. 4.20, an example of partition functions.

Here we impose the labelings C_l to be conditionally independent such that

$$p(C = c|S = s) = \prod_l p(C_l = c_l|S = s)\,.$$

We also assume here that $\boldsymbol{\beta}$ is made bijective. Given a signal s, an estimation for z is given by:

$$\log(p(Z = z|S = s)) = \log(p(C = \boldsymbol{\beta}(z)|S = s)) \tag{4.43}$$

$$= \sum_l \log p(C_l = \beta_l(z)|S = s)) \tag{4.44}$$

A set of L classifiers $\{y_1, \ldots, y_L\}$ is trained accordingly to partitions β_l and gives predictions $p(C_l = \beta_l(z)|S = s)$. Once each classifier is trained, their predictions can be naively summed, in which case a *soft voting* (SV) is performed; or a classifier-specific weight can be applied to each classifier depending on its performance, that is a *weighted soft voting* (WSV). Remark that SV is a particular case of WSV where weights are all equal. If we note $y_l(z, s) = \log(p(C_l = \beta_l(z)|S = s))$ the vote accorded to the classifier l for the value z of Z, and $y(z, s) = \sum_l w_l y_l(z, s)$ the weighted vote with $w_l \in \mathbb{R}$. We can iteratively find a weight vector $w \in \mathbb{R}^L, \sum_{i=1}^{L} w_i = 1$, such

that the following cross-entropy loss is minimized:

$$L_{\text{wsv}}(X,Z) = -\frac{1}{N_t}\sum_{(s_i,z_i)\in\mathcal{D}_t} p(Z=z_i)y(z_i,s_i) \tag{4.45}$$

$$= -\frac{1}{N_t}\sum_{(s_i,z_i)\in\mathcal{D}_t}\sum_{l=1}^{L} w_l p(Z=z_i)\log(p(C_l=\beta_l(z_i)|s=s_i)) \tag{4.46}$$

where $\mathcal{D}_t = \{(s_i, z_i)\}$ is the training dataset.

To illustrate our approach, we consider the case where signals s are Gaussian distributed with the same covariance matrix. This is equivalent to choosing Linear Discriminant Analysis as classifiers [49], we have:

$$y_l(z,s) = \log\left(\frac{1}{R}e^{(s-\mu_l(z))^t\Sigma(s-\mu_l(z))}\right)$$

With R the normalization factor, $\mu_l(z)$ the mean value of signals for the labeling l and the label value z, and Σ the inverse covariance matrix. We assume a balanced dataset, i.e. $p(Z=z_i)=p$ is constant, and constraint weights such that $\sum_l w_l = 1$, we get:

$$L_{\text{wsv}}(X,Z) = -\frac{p}{N_t}\sum_{(s_i,z_i)\in\mathcal{D}_t}\sum_{l} w_l\left((x_i-\mu_l(z_i))^t\Sigma(x_i-\mu_l(z_i))-\log(R)\right) \tag{4.47}$$

$$= -\frac{p}{N_t}\sum_{(s_i,z_i)\in\mathcal{D}_t}\left((x_i-\mu^*(z_i))^t\Sigma(x_i-\mu^*(z_i))+c_\mu(z_i)-\log(R)\right) \tag{4.48}$$

$$\propto \log(\prod_{(s_i,z_i)\in\mathcal{D}_t}\frac{1}{R}e^{(-(x_i-\mu^*(z_i))^t\Sigma(x_i-\mu^*(z_i)))}) \tag{4.49}$$

Where $\mu^* = \sum_l w_l\mu_l$ and $c_\mu = \sum_l w_l\mu_l^t\Sigma\mu_l - \sum_{l,k} w_l w_k\mu_l^t\Sigma\mu_k$ that depends on estimated means μ_l, on weights w_l and on the inverse covariance matrix Σ. In the Gaussian distributed case with a fixed covariance matrix, we can see that the minimization of $L_{\text{wsv}}(X,Z)$ is equivalent to minimizing $(x_i-\sum_l w_l\mu_l(z_i))^t\Sigma(x_i-\sum_l w_l\mu_l(z_i))$ which is a simple linear regression with parameters $\boldsymbol{w}$.

Our combination procedure can be seen as a generalization of the Linear Regression Analysis of *Schindler et al* [106] where no assumption is made on the linearity of the leakage model. Arbitrary complex classifiers can be used to draw relations between signals and labels and the relevance of such relation can be evaluated by minimizing the cross-entropy criterion, i.e. classifiers with the highest weights are the most relevant. To obtain the overall estimation, log probabilities are linearly summed according to a simple Bayes rule, in case classifiers output scores, a logistic regression layer [49] can be added and trained to get probabilities.

As remarked Zhou in [125, Chap 4.3.5.2] the global score obtained after minimiza-

tion can be worse than considering the best classifier in the model. This procedure is interesting when no knowledge about the leakage model is available and can be iteratively improved by removing bad classifiers, i.e. when their weights are too low.

In practice, classifiers are individually trained on their associated labeling C_l and their predictions are combined after minimizing (4.46) with the weight vector $\boldsymbol{w}$. To automatize this process and to ensure reproducibility of the experiments, a specific library featuring automatic labelling and in parallel training of classifiers has been implemented in Julia.

4.4.4 Experiments

In this section, we integrate the two previous methods presented Sections 4.4.1 and 4.4.3 to perform attacks on desynchronized signals from the JIT and ASCAD datasets presented in Sec. 4.1.1. Attack results are compared with other types of preprocessing: raw temporal signals and STFT of signals. We also study the effect of optimizing the weights of the combination procedure (4.46) on attack results.

4.4.4.1 Overall model

We propose the method displayed on Figure 4.21. First, signals are preprocessed with the scattering transform (WST), then a PCA is applied to reduce the dimension and finally QDA classifiers trained on predefined labelings C_l outputs predictions which are merged with a Weighted Soft Voting (WSV) (4.46).

The set of classifiers is trained on canonical partitions, i.e. identity on z, Hamming weight and bit values:

$$\{\mathrm{Id} : z \to z,\ \mathrm{HW} : z \to \mathrm{HW}(z),\ \mathrm{Bit}_i : z \to (z \gg i)\ \&\ 1\ \forall i \in \{0, 1, \ldots, 7\}\}$$

where $\gg$ is the shift right operation for bit-vectors and & the bitwise `AND` operation.

4.4.4.2 Choosing the Parameters

Hyperparameters for the preprocessing with scattering transforms and STFT are chosen accordingly to the dataset and attack results.

For ASCAD, we used $54,000$ signals for the training set and $6,000$ signals for the attack set. For the scattering transform, signals are first upsampled to $1,024$ points, we fixed Q=1 since a fine resolution between high frequency bands is not required. We obtained good results with time scales J=3 and J=7, and limited the scattering transform to one layer m=1. For STFT preprocessing, signals are also upsampled to $1,024$. The best result in terms of guessing entropy is obtained with a

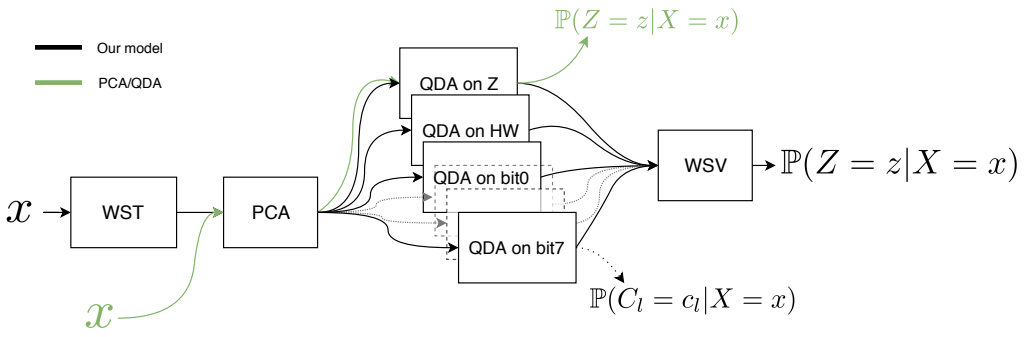


Figure 4.21: Illustration of the global method in black with the Wavelet Scattering Transform (WST) and the Weighted Soft Voting (WSV) from Sections 4.19 and 4.4.3. We also depicted in green a standard Template Attack (QDA) with PCA. We replace the WST with the modulus of a Short-Time Fourier Transform when comparing with STFT preprocessing.

sliding window of 128 points which corresponds to a time scale of 88 in the original signals, the overlap was set to 64.

For JIT, we considered a restrained dataset of $75,000$ signals since STFT and raw representation had too many features to fit the whole dataset in memory and to perform the PCA based dimension reduction. We managed to fit signals preprocessed with WST in memory when considering the whole training set of size $150,000$. For WST, we expected the JIT dataset to have a strong jitter so we set the following parameters J=10, Q=8, m=2 which gave preprocessed signals of size $2,992$. For STFT, we used a sliding window of size $1,024$ with an overlap of 512 which gave STFT of $7,680$ features.

For each dataset we limited the PCA to 50 components which corresponds to the number of components used for SoA template attack combined with a PCA on aligned temporal signals. When minimizing the loss function (4.46), we stopped the gradient descent after 200 iterations.

4.4.5 Results

In order to evaluate our model, we performed our attack on three folds. For each fold an intermediate guessing entropy (GE) measure [83] is calculated by averaging 100 rank curves obtained by shuffling the order of signals in equation (3.8). The final guessing entropy is obtained by averaging the guessing entropy of the three folds.

In the following we use the following notations: SV and WSV (4.46) when respectively a soft voting and weighted soft voting is applied with all the classifiers, SumBits a soft voting with the classifiers on bits, Z when considering only the classification on the byte and HW with the hamming weight. "Temp","Spec" and "Scat" respectively denote the raw temporal representation, the STFT preprocess-

ing and the Wavelet Scattering Transform. Attack results on SumBits, Z, HW and SV are used to characterize the performance of each preprocessing. The rank gap between SV and WSV indicates the efficiency of the combination procedure (4.46) for merging prediction of differently performing classifiers. We displayed on Table 4.1, the weights obtained after optimizing the WSV and the number of attack signals required to have a guessing entropy of 40 (NGE_{40}) when considering classifier individually (Z and HW), with SumBits, SV and WSV.

Results for ASCAD are displayed Figure 4.22 and on Table 4.1. When no desynchronization is present, preprocessing with a small time scale of analysis performs the best: attack results on SumBits are almost identical when considering WST with J=3, spectrograms and raw temporal signals; the same WST performs slightly better for Z and SV. Intriguingly the effect of desynchronization on attack results in ASCAD_{100} strongly varies with labelings. The large scale WST with J=7 performs the best on Z and SV and shows its robustness to desynchronization; the attack on SumBits is better with spectograms and might be due to the overlap between frames of analysis. The combination procedure resulted differently: it decreased the rank of SV of $2,000$ with spectograms preprocessing and of only 5 with WST. Globally, as expected the WSV is better than SV and makes all models converge to rank 1 except for temporal attacks on ASCAD_{100}.

In presence of a strong jitter and deformations in JIT, spectrogram and temporal attacks fail for any classifier while preprocessing with WST provides better attack results and becomes possible on SumBits (see Figure 4.23 and Table 4.1). On JIT, the WSV performed well and decreased the rank of SV for WST of approximately $1,600$.

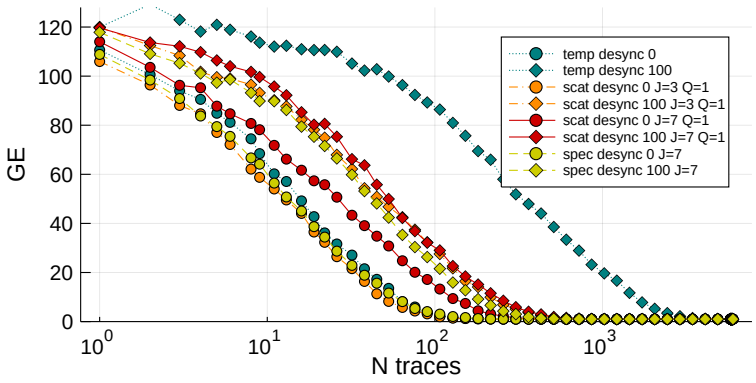
The weights of the WSV seem to be correlated with the guessing entropy of classifiers, e.g. when considering temporal attacks we see that weights on bits are higher than weights on H or Z. On ASCAD, the weights for the WST seem to be more distributed among classifiers and could explain why the weighted soft voting did not converged as well as for STFT preprocessing where the classifier over Z was heavily penalized. In other words, the iterative optimization of WSV seems to be facilitated with classifiers of unbalanced performance. We also notice the fact that bits are leaking dissymmetrically as proposed by *Suzuki et al.* in [114], e.g. on ASCAD the classifier on bit0 has a higher weight than the average on bits (SumBits), while on JIT the weight on bit7 is higher when considering successful models (Scattering with JIT 75k and 150k).

From our results on these datasets and given QDAs as classification models, Z and HW leakage models are globally disadvantaged when looking at the guessing entropy and the weights associated. The WSV has approximated a leakage model that relies more on individual bits. The difference of performance between Sumbits,

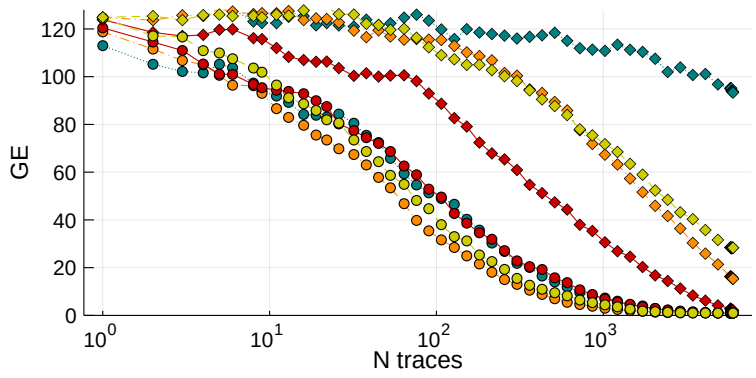
Dataset	Preprocessing		Z	H	Bit0	Bit4	Bit7	SumBits	SV	WSV
ASCAD_{100}	Temp	$\boldsymbol{w}$	<0.01	<0.01	0.29	0.18	0.19	0.19		
		NGE_{40}	∞	∞	-	-	-	485	∞	1465
	Spec	$\boldsymbol{w}$	0.02	0.22	0.39	0.31	0.32	0.31		
		NGE_{40}	3527	3392	-	-	-	**57**	2126	**242**
	Scat	$\boldsymbol{w}$	0.29	0.15	0.32	0.23	0.21	0.20		
		NGE_{40}	**676**	**675**	-	-	-	70	**428**	423
JIT 75k	Temp	$\boldsymbol{w}$	<0.01	0.15	0.34	0.43	0.34	0.39		
		NGE_{40}	∞	∞	-	-	-	∞	∞	∞
	Spec	$\boldsymbol{w}$	0.08	0.17	0.20	0.27	0.25	0.24		
		NGE_{40}	∞	∞	-	-	-	∞	∞	∞
	Scat	$\boldsymbol{w}$	0.19	0.18	0.42	0.48	0.48	0.47		
		NGE_{40}	9371	8851	-	-	-	1561	6102	4513
JIT 150k	Scat	$\boldsymbol{w}$	0.11	0.12	0.41	0.41	0.48	0.45		
		NGE_{40}	**7837**	**8023**	-	-	-	**884**	**3770**	**2149**

Table 4.1: For each preprocessing (Temp for raw temporal signal, Spec for STFT and Scat for WST): the required number of signals to get a guessing entropy of 40 (NGE_{40}) when considering individual classifiers with labeling over Z and H, with a soft voting over bits noted SumBits, with the overall Soft Voting SV and finally with the Weighted Soft Voting. We also indicated the weights of the classifiers obtained after optimizing the WSV for classifiers over Z, H, some individual bits and their average for SumBits. For ASCAD_{100}: we displayed the results obtained with a WST with J=7 and Q=1. For JIT: results with training on $75,000$ and $150,000$ signals.

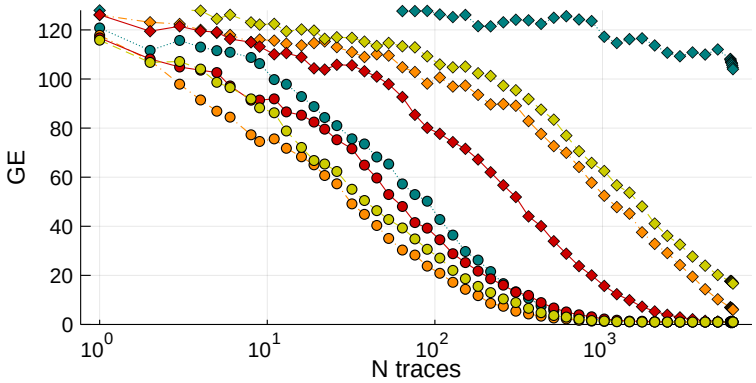
Z and HW is also explained by the number of samples required to estimate the parameters of QDAs, which makes attacks on individual bits more stable since less parameters are required. Thus a trade-off has to be made on the number of components for the PCA: while a high number of components increases the number of parameters to estimate, the attack results can be improved by selecting more eigenvectors with lower eigenvalues and better discriminating power.



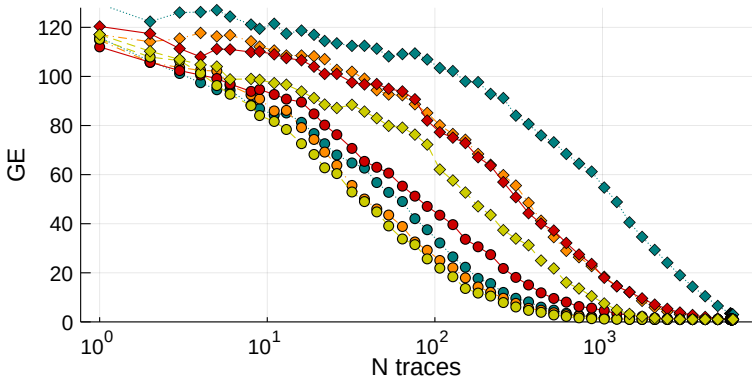
(a) GE SumBits



(b) GE Z

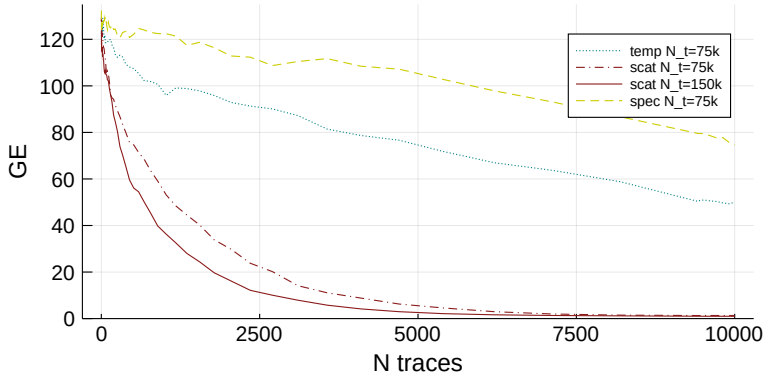


(c) GE SV

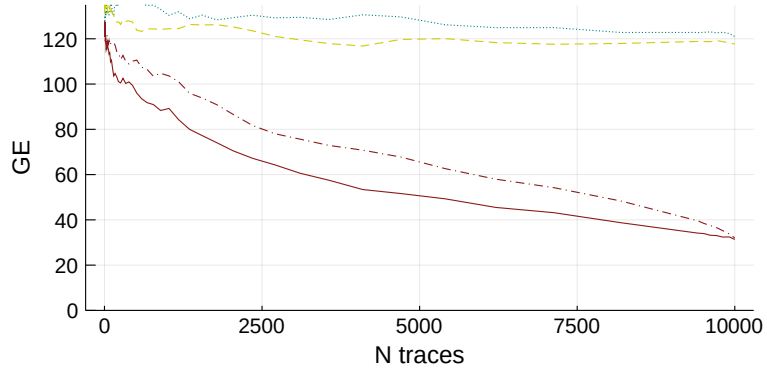


(d) GE WSV

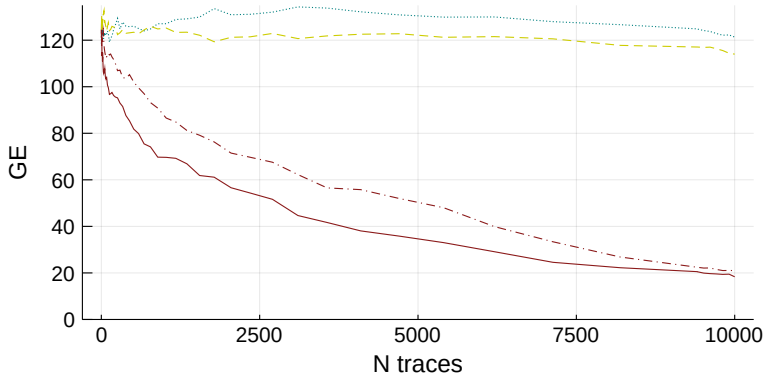
Figure 4.22: Guessing entropy as a function of the number of attack signals on ASCAD with classifiers trained on Sumbits, Z, SV and WSV, after different preprocessings (Temp for raw temporal signal, Spec for STFT and Scat for WST)



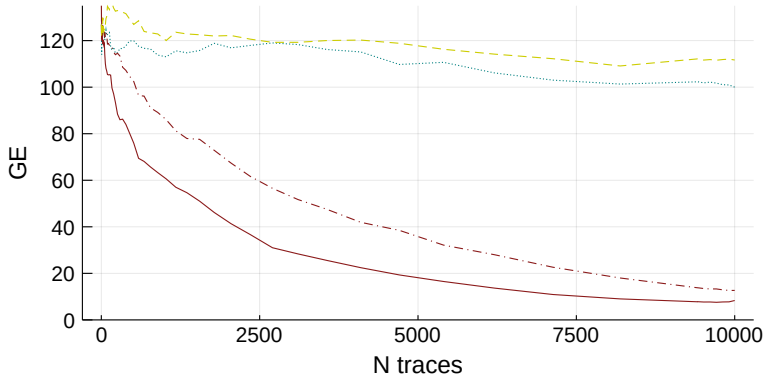
(a) GE SumBits



(b) GE Z

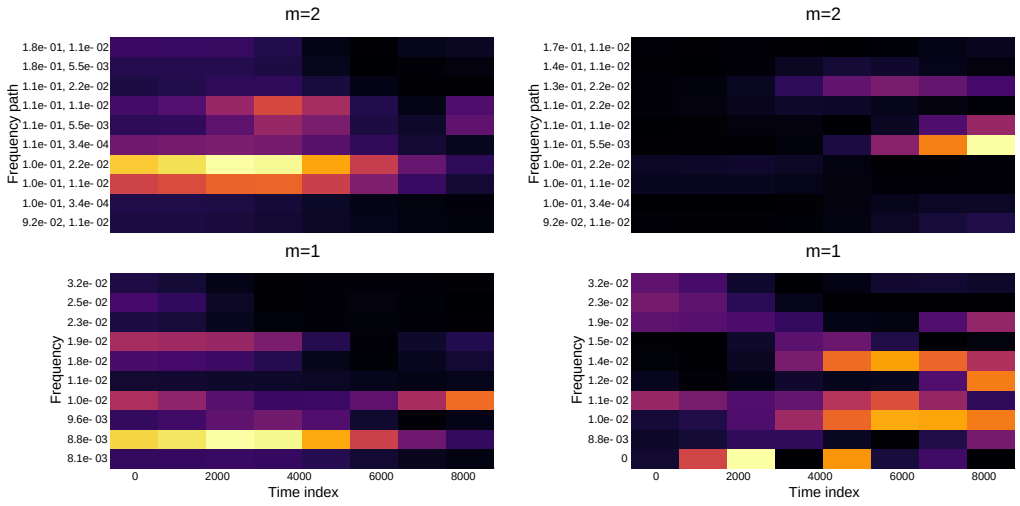


(c) GE SV



(d) GE WSV

Figure 4.23: Guessing entropy as a function of the number of attack signals on JIT with classifiers for Z, SumBits, with naive combination of prediction (SV) and with WSV. N_t is the number of signals used for training.



(a) Scat SumBits (b) Scat Z

Figure 4.24: Leakage visualization on Jit. On top the second level of the WST. Below the first level of the WST. We selected the top 10 frequency bands (and frequency paths for the second level) that contains the highest values of SNR. Amplitudes are scaled between 0 and 1.

4.4.5.1 Visualizing leakages

We previously showed results in terms of guessing entropy. We propose here an easy computation of a SNR score on the preprocessed signals by taking into account the covariances and means estimated during training. For each classifier l we compute a SNR score in the subspace induced by the PCA with a projection $P \in \mathbb{R}^{d \times p}$, where p is the number of components chosen for the PCA and d is the original dimension.[3] Each QDA classifier is defined by means $\mu_{l,i} \in \mathbb{R}^p$ and covariances matrices $\Sigma_{l,i} \in \mathbb{R}^{p \times p}$ for each label values $c_{l,i}$, $\forall i$. We note $\mathrm{SNR}_l^s \in \mathbb{R}^p$ and $\mathrm{SNR}_l^o \in \mathbb{R}^d$ respectively the SNR in the subspace and in the original space before the PCA, we have:

$$\mathrm{SNR}_l^s[r] = \frac{\mathrm{Var}_i\left[\mu_{l,i}[r]\right]}{\mathbb{E}_i\left[\mathrm{Diag}(\Sigma_{l,i})[r]\right]} \quad , \; r = 1, \ldots, p$$
$$\mathrm{SNR}_l^o = (P\,\mathrm{SNR}_l^s).\hat{}\,2 \tag{4.50}$$

Where $.\hat{}$ defines the entry-wise power. This score (4.50) gives some indication on the temporal and frequency aspects of the leakage. We computed some visualizations of this score for attacks on JIT respectively in Figure 4.24. Remark that these analyzes can be perturbed by the subspace induced by the PCA's eigenvectors. When the SNR is high we suppose that it gives some indication about how signals are leaking information. For SumBits we summed the SNR scores.

[3] After preprocessing, wavelet scattering transform and spectrogram representations are vectorized before the PCA.

On Fig. 4.24, the SNR visualization with the scattering transform positions the leakage around time index 2,000 when considering SumBits and Z. The two-level scattering transform has proven useful, the SNR score indicates for bits that the frequency band 1.0e-01 is leaking. For Z the 1.1e0-1 frequency path gives clues about a leakage around time index 8,000, which is also shown but more discreetly at the first level for z or for both level with SumBits.

4.4.6 Conclusion

Independently of choosing a classification model, we proposed two ways of injecting prior information in preprocessing and classification in order to increase the performance of Side-Channel attacks.

In this section, we adressed the problem of desynchronization and deformation encountered in side-channel analysis by using the scattering transform as a preprocessing step. In contrast with other time-frequency representations, such as STFT and Wavelet Transforms, the scattering transform provides robust representations that prooves useful against jitter protected signals.

Secondly, based on the fact that in general the leakage model is an unknown function of the sensitive variable, we proposed an approximation method by considering various labelings of the sensitive variable. For that, we train classifiers on different partitions of the sensitive variable‘s values and combine their predictions. Our combination method involves finding a weight vector which assesses the contribution of each classifier in the global prediction. To this end, the weights are found by iteratively minimizing a cross-entropy criterion.

These two propositions have been evaluated by integrating them in a new attack method, which successfully increased the performance of Template Attacks on artificially desynchronized signals and signals from a jitter-protected implementation. The wavelet scattering transform improves the performance of Template Attacks when jitter effects and distortion are present in signals. Although, we restricted ourself to Template Attacks as classification models, this preprocessing could be particularly interesting when followed by more complex classifiers, e.g. a convolutional neural network. We argue that it could reduce the amount of data required to normally make any classifier robust under small translation and deformations. The experimental results showed that the combination procedure makes attacks successful as long as some classifiers manage to get information from partitions of the sensitive variable. While specifying a fixed leakage model constraints the classifier to a given goal, the proposed combination procedure allows an evaluator to test various leakage models and quickly evaluate which ones he should focus on.

Chapter 5

Generative model for Side-Channel Analysis

In order to tackle jitter countermeasures in side-channels, we derived in the previous chapter a technique to realign signals with patterns extracted from wavelet representations in Sec. 4.2.4 and a preprocessing to compensate desynchronisation in side-channel signals in Sec. 4.4.

In this section, we adopt a different point of view, we will propose a generative model for side-channel signals in order to model the algorithm, the jitter countermeasure and the generation of patterns. We will present a method to simulate side-channel signals and preliminary results of a method to efficiently recover the time of occurrence of operations. We will see that the use of wavelet frames presents interesting capabilities to generate patterns and to initialize parameters for the estimation of the times of occurrence.

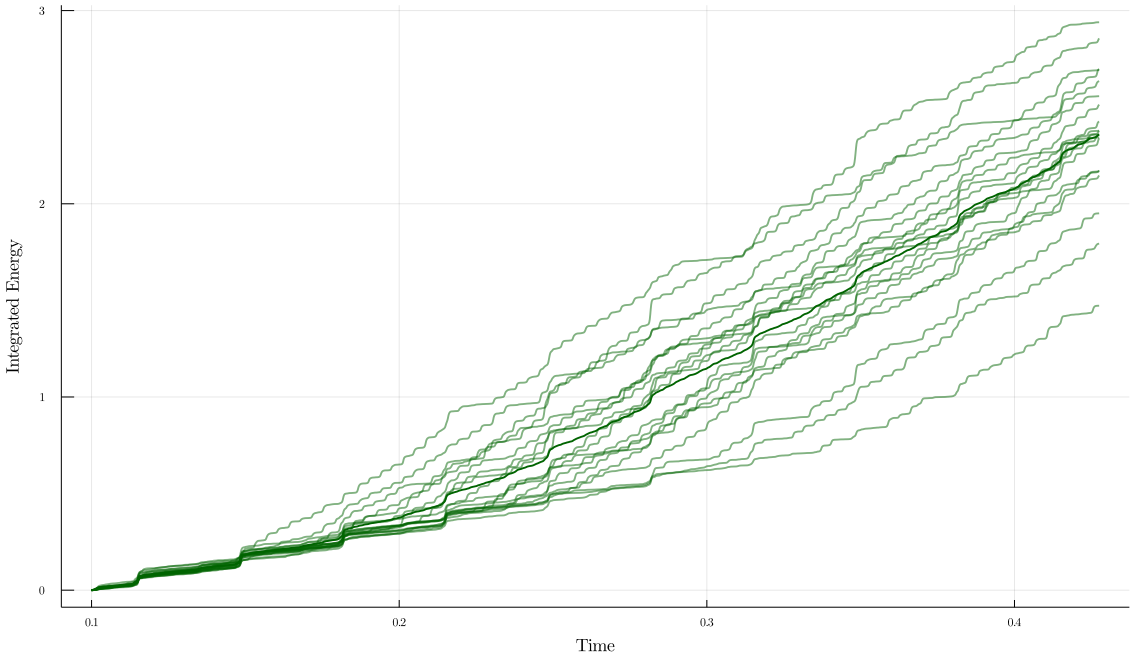


Figure 5.1: Evolution of the energy consumption integrated over time for 20 side-channel signals with jitter countermeasure. The jitter countermeasure appears around time 0.2 for the AES encryption algorithm. In bold, the average integrated energy consumption. We clearly see the effect of the jitter countermeasure as the trajectories evolve: the time between "jumps" of energy is clearly random. We denoised the original signals with a wavelet transform to highlight the jumps of energy.

5.1 Motivation

The patterns present in side-channel signals are related to physical processes on the chip that are paced by the clock. Each switch of the clock triggers a flux of electrons on the chip that induces a relatively high amount of energy consumed over a short period of time and creates a pattern in the signal. To illustrate this, we show Fig. 5.1 the evolution of the energy integrated over time for jitter protected signals in the JIT dataset. We remark that the trajectories evolve randomly and almost step by step. If the increase of energy was instantaneous and constant each time an operation is executed, the trajectories would look like those of point processes in the field of stochastic processes. Point process models are used to model random time arrivals in queues, and has been employed to estimate heart rate variability in electrocardiogram signals in [11].

In this chapter, we take inspiration from the field of stochastic processes and in particular of point processes to model the jitter. The use of a model for the jitter to locate more efficiently the patterns in side-channel signals has yet not been proposed in the literature, we think that it may be a good direction to improve side-channel attacks, as it allow the evaluator to incorporate information about the

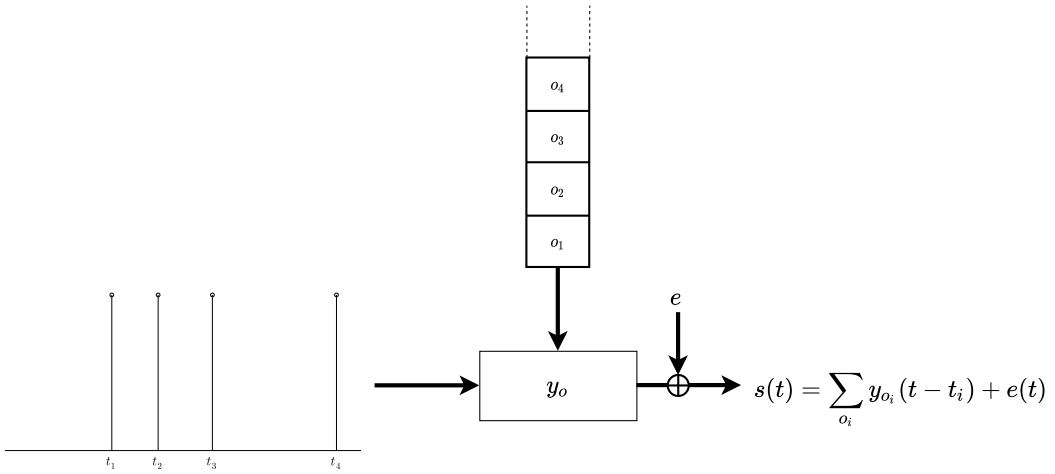


Figure 5.2: A filter-based model for side-channel analysis. We assume that a sequence of machine states $\{o_i\}$ and of time points $\{t_i\}$ forming an impulse train are chosen randomly. At each impulse passing through the system y_o, a filter y_{o_i} is chosen according to the first value o_i in the ordered sequence $\{o_i\}_i$ representing the successive operations that occur on the device.

jitter. Furthermore it could allow an analytical study of the entropy of the jitter on the security of the device.

5.1.1 Filter model for SCA

We introduce here the general model before going into some details. We assume that a side-channel signal s observed over a time range $[0, T_0]$ satisfy:

$$s(t) = \sum_{i \geq 1} y_{o_i}(t - t_i) + e(t)\,, \forall t \in [0, T_0] \tag{5.1}$$

where $\{o_i\}_i$ is an ordered sequence of operations, $\{t_i\}_i$ the times of occurrence, $\{y_{o_i}\}_i$ the patterns produced by each operations seen here as functions of the time, and e a statistical noise.

We present a filter-based model in Fig. 5.2. The jitter is represented by a shot noise, a Dirac impulse train, entering a filter that changes at each operation. Each time a Dirac impulse enters the system, the filter outputs an impulse response corresponding to the first operation in the remaining ordered list of operations, called the stack. The jitter is represented by the random delay between each Dirac impulse in the shot noise. If the delay is constant, no jitter is present.

In this model, the times of occurrence and the machine events play the role of hidden variables that control the dynamic of the signal. In the following, we will specify a model for the operations and the jitter. For the generation of patterns, we will extend the model based on wavelet frames of Sec. 4.3.

5.2 A model for the algorithm

An operation designates either a legitimate operation from the developer or a fake one randomly injected by a countermeasure installed by the fabricator of the hardware. The execution of an operation can be thought as a state in which the machine enters. The state of the machine is characterized by the type of operations and by the values of the variables being manipulated.

We assume a finite amount of possible machine operations, say $N_o \in \mathbb{N}$, such that $|\boldsymbol{O}| = N_o$ with $\boldsymbol{O}$ the space of operations. The operation at step t is noted o_t. We assume that random machine events O have a simple categorical distribution:

$$O \sim \text{Cat}(\alpha) \text{ with } \alpha \in \mathbb{R}^{+N_o}, \sum_{i=1}^{N_o} \alpha_i = 1 \,, p(O = i) = \alpha_i \tag{5.2}$$

The global dynamic of a side-channel signal is characterized by the transitions between different operations. One approach is to model the transitions with Markov Chains [63, 41, 40], thus if we note $M \in \mathbb{R}^{N_o \times N_o}$ the transition matrix of an algorithm, $M_{i,j}$ contains the probability to go from $o_{t-1} = i$ to $o_t = j$. To form ordered sequences, we introduce the space of k finite length sequence

$$\boldsymbol{A}^k = \{a = (o_1, \ldots, o_k) | o_i \in \boldsymbol{O}\}$$

. With this model, the probability of the k-length sequence $a \in \boldsymbol{A}^k$ is:

$$p_A(A = a) = p(o_1) \prod_{i=2}^{k} p(o_i | o_{i-1}) = \alpha_1 \prod_{i=2} M_{i,i-1} \tag{5.3}$$

A natural extension of the Markov Chain model is to directly consider the probabilistic graph of the cryptographic algorithm, this approach is presented in [117, 70].

While this model allows to take into account the type of operations executed at each cycle of the clock, the jitter model introduced in the next section models the delay between each operations.

5.3 Jitter model

We now present a model for the jitter encountered in side-channel signals. We distinguish continuous and discrete models. Continuous models can be used to represent jitter countermeasures with unstable clock, while discrete models will represent software based jitter countermeasures with the introduction of dummy (`nop`) operations with a fixed delay. Here we will focus on the continuous models, and we will consider

the jitter independent of the sequence of operations.

5.3.1 Jitter as a Poisson point process model

We start by a simple model with exponential delay. This model introduces some quantities that will be used in a subsequent model. With $\tau_i = t_i - t_{i-1}$ the delay between operations o_i and o_{i-1}, and ΔT_i its random variable, the model is given by:

$$\begin{aligned} \lambda > 0,\ \forall i \in \mathbb{N}, \Delta T_i \sim \text{Exp}(\lambda), \\ \text{i.e. } p_{\Delta T_i}(\Delta T_i = \tau) = \mathbf{1}_{\tau>0}\lambda e^{-\lambda\tau} \end{aligned} \tag{5.4}$$

With $t_0 = 0$ and T_i the random times of occurrence such that

$$T_i = \sum_{j=1}^{i} \Delta T_j$$

and we note:

$$p_{T_i|T_{i-1}}(T_i = t_i | T_{i-1} = t_{i-1}) = p_{\Delta T_i}(\Delta T_i = t_i - t_{i-1}) \tag{5.5}$$

In practice, we will observe signals over a bounded region of time. We are then interested in the random event *exactly k operations occurred before t* noted $N_t = k$. Thus, we introduce the space $\boldsymbol{T}_t^k$ of times of occurrence

$$\boldsymbol{T}_t^k = \left\{ (t_1, \dots, t_{k+1}) \in \mathbb{R}^{k+1} \,\middle|\, 0 < t_1 < \cdots < t_k \leq t < t_{k+1} \right\} \tag{5.6}$$

which can equivalently be written

$$\boldsymbol{T}_t^k = \left\{ (\tau_1, \dots, \tau_{k+1}) \in \mathbb{R}^{k+1} \,\middle|\, \sum_{i=1}^{k} \tau_i < t,\ \tau_{k+1} > t - \sum_{i=1}^{k} \tau_i \right\}. \tag{5.7}$$

The probability of $N_t = k$ is given by

$$p(N_t = k) = p(t_k \leq t,\, t_{k+1} > t) \tag{5.8}$$

$$= \int_{\boldsymbol{T}_t^k} \prod_{i=1}^{k+1} p(t_i | t_{i-1}) dt_1 \dots dt_{k+1} \tag{5.9}$$

$$= \frac{1}{k!}(\lambda t)^k e^{-\lambda t} \tag{5.10}$$

The last identity can be found in various textbooks on Poisson processes [97, 34] or by direct calculation using the convolution property for sums of independent variables, i.e. $p(\sum_{i=1}^{k} \Delta T_i = t) = (\circledast_{i=1}^{k} p_{\Delta T_i})(t)$.

Additionally, the probability of the event *at least k operations occurred before t* noted $N_t \geq k$ is

$$p(N_t \geq k) = \frac{1}{(k-1)!}\gamma(k, \lambda t)\,, \tag{5.11}$$

where $\gamma(a,t) = \int_0^t u^{a-1}e^{-u}du$, $a > 0$ is the lower incomplete gamma function. It will be useful in the next model. We give further details in Appendix. A.2.

Under this model, the average number of operations before a time t is:

$$\mathbb{E}\left[N_t\right] = t\lambda\,. \tag{5.12}$$

The model presented here however does not characterise well a clock jitter as the probability that two operations get arbitrary close in time is non-negligible. As these events are very unlikely to happen in reality, we consider a natural extension of this model in the following.

5.3.2 Gamma point process model

The previous model can be extended to construct the Gamma point process model. We will say that a Gamma step occurs if k Poisson steps has occurred. Thus, a Gamma distributed delay ΔT_i, with parameters $k \in \mathbb{N}, \lambda \in \mathbb{R}_*^+$ follows [98, Chap.5, p.174]:

$$p(\Delta T_i = \tau) = \mathbf{1}_{\tau>0}\frac{\lambda}{(k-1)!}(\lambda\tau)^{k-1}e^{-\lambda\tau}\,. \tag{5.13}$$

With k integer, we interpret τ as the time it took for k elementary Poisson steps to occur between two Gamma steps.

Let $N_t \geq n$ the event *at least n operations occurred before t*, its probability is given by:

$$p(N_t \geq n) = p(N_t^p \geq nk) \tag{5.14}$$

$$= \frac{1}{(nk-1)!}\gamma(nk, \lambda t) \tag{5.15}$$

where $N_t^p \geq nk$ is the event *at least nk Poisson steps occurred before t* given by (5.11). In other words, nk elementary Poisson steps are needed before t to account for n Gamma steps.

We can now derive the probability for the event *exactly n operations before t*, we

have:

$$p(N_t = n) = p(N_t \geq n) - p(N_t \geq n+1) \tag{5.16}$$
$$= p(N_t^p \geq nk) - p(N_t^p \geq (n+1)k) \tag{5.17}$$
$$= \sum_{m=nk}^{(n+1)k-1} p(N_t^p = m) \tag{5.18}$$
$$= \sum_{m=nk}^{(n+1)k-1} \frac{1}{m!}(\lambda t)^m e^{-\lambda t} \tag{5.19}$$

where N_t^p is the Poisson event, with the use of (5.10) in (5.18).

To compute $p(N_t = n)$ in practice we will use the implementation of some mathematical libraries of the lower incomplete gamma function as with (5.15) we can write $p(N_t = n)$ as:

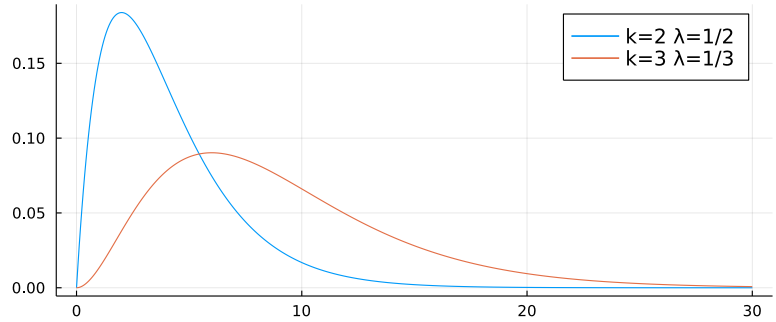
$$p(N_t = n) = \frac{1}{(nk-1)!}\gamma(nk, \lambda t) - \frac{1}{((n+1)k-1)!}\gamma((n+1)k, \lambda t)\,. \tag{5.20}$$

We do not formally derive here the average number of Gamma steps, but since a Gamma step corresponds to the realisation of k Poisson steps, we evaluate the average number of Gamma events before t to be $\lambda t/k$.

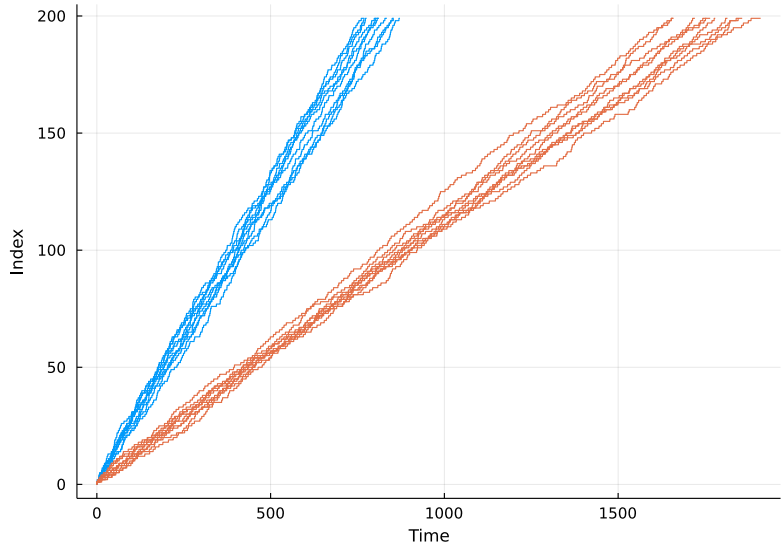
For simplicity here, we presented the probability of $N_t \geq n$ and $N_t = n$ with k integer, but by continuity of $\gamma(nk, \lambda t)$ the model can be used with $k \in \mathbb{R}_*^+$.

We present on Fig. 5.3 some simulations with this model. We also verify on Fig. 5.3c that the probability of $N_t = n$ in (5.19) is correct. In particular, we remark similarities between the integrated energy of signals from Fig. 5.1 and the evolution of point processes of Fig. 5.3.

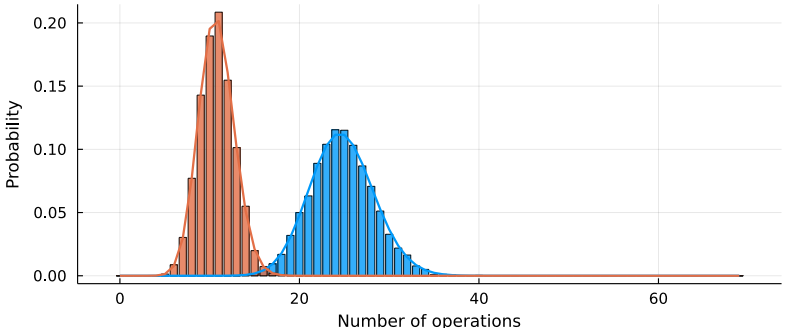
In the following we will use the Gamma point process to represent continuous jitter countermeasures.



(a) Probability distribution of a Gamma distributed delay.



(b) Trajectories of Gamma point processes.



(c) Probability of *exactly k operations before T=100*.

Figure 5.3: Jitter countermeasure as a Gamma point process. We compare two Gamma point processes with parameters $k = 2, \lambda = 1/2$ and $k = 3, \lambda = 1/3$.

5.4 Gaussian Mixture Model for patterns with GMW factorized covariances

We employ in this section the model presented in Sec. 4.3 to generate patterns with a frame of wavelets. It allows the generation of smooth signals and the inclusion of prior information on the time-frequency properties of patterns.

We will extend this model by assuming a Gaussian Mixture Model (GMM) on the synthesis coefficients of the patterns. With the GMM, the synthesis coefficients will be conditioned on the operations. It permits the generation of specific patterns given the value of some operations.

For simplicity here, we will work with continuous signals and consider the generation of patterns in $\mathcal{L}^2(\mathbb{R})$. Let Ψ_Ξ the frame operator (Sec. 1.3.3) with a basis of real Generalized Morse Wavelets (GMW) $\{\psi_\xi\}_{\xi\in\Xi}$ with a finite size index set $|\Xi| = m$. To generate patterns in $\mathcal{L}^2(\mathbb{R})$, we will use the GMW in the real domain, i.e. we take the real part of the complex GMW with parameters ξ.

Let $o \in \boldsymbol{O}$ an operation, noting its synthesis coefficients $x_o \in \mathbb{R}^m$, its pattern $y_o \in \mathcal{L}^2(\mathbb{R})$ is generated using the frame Ψ^*_Ξ by:

$$y_o = \Psi^*_\Xi x_o \,, \tag{5.21}$$

with Ψ^*_Ξ the adjoint of Ψ_Ξ.

At $t \in \mathbb{R}$, we can express the pattern in function of each wavelet $\psi_\xi, \xi \in \Xi$ of the frame, we have:

$$y_o(t) = \sum_{\xi\in\Xi} \psi_\xi(t) x_o[\xi] \,, \tag{5.22}$$

where $x_o[\xi] \in \mathbb{R}$ is the synthesis wavelet coefficient for the wavelet ψ_ξ and for the machine event o.

To introduce the influence of the operations on the synthesis coefficients, we note X the random variable for the synthesis coefficients, and assume that it follows a Gaussian Mixture Model (Sec. 2.5):

$$p_X(X = x) = \sum_{o\in\boldsymbol{O}} p(o)\mathcal{N}(x|\mu_o, \Sigma_o) \,. \tag{5.23}$$

Each operation $o \in \boldsymbol{O}$ is associated to a mean $\mu_o \in \mathbb{R}^m$ and a covariance $\Sigma_o \in \mathbb{R}^{m\times m}$.

We can now deduce the probability distribution of the random pattern $Y(t)$ at $t \in \mathbb{R}$. Since Ψ_Ξ is a linear operator, $Y(t)$ also follows a Gaussian Mixture model:

$$p_{Y(t)}(Y(t) = y(t)) = \sum_{o\in\boldsymbol{O}} p(o)\mathcal{N}(y(t)|(\Psi^*_\Xi\mu_o)(t), (\Psi^*_\Xi\Sigma_o\Psi_\Xi)(t,t)) \,, \tag{5.24}$$

with respectively mean and variance given by

$$\mu_{y,o}(t) = (\Psi_{\Xi}^{*}\mu_o)(t) = \sum_{\xi \in \Xi} \mu_o[\xi]\psi_{\xi}(t) \tag{5.25}$$

$$\Sigma_{y,o}(t,t') = (\Psi_{\Xi}^{*}\Sigma_o\Psi_{\Xi})(t,t') = \sum_{\xi_1 \in \Xi}\sum_{\xi_2 \in \Xi} \Sigma_o[\xi_1,\xi_2]\psi_{\xi_1}(t)\psi_{\xi_2}(t')\,, \tag{5.26}$$

where $\Sigma_o[\xi_1,\xi_2] \in \mathbb{R}$ is the covariance coefficient between wavelets ψ_{ξ_1} and ψ_{ξ_2}.

Finally, we recall that with the GMMs on Y and X, and given the value of an operation o, the probability distribution of the synthesis coefficients follows

$$p(X = x|O = o) = \mathcal{N}(x|\mu_o, \Sigma_o)\,, \tag{5.27}$$

and the probability distribution of the pattern at $t \in \mathbb{R}$ is

$$p(Y(t) = y(t)|O = o) = \mathcal{N}(y(t)|(\Psi_{\Xi}^{*}\mu_o)(t), (\Psi_{\Xi}^{*}\Sigma_o\Psi_{\Xi})(t,t))\,. \tag{5.28}$$

We presented the probability distribution of $Y(t)$ for a single value of time t. However, in practice, as our patterns are structured in the time domain, we will get a more meaningful estimation of the probability of an observed pattern if we sample it n times, e.g. at time indices $u_1, \ldots, u_n$, thus forming a vector of sampled values $[Y(u_1), \ldots, Y(u_n)]$.

The probability distribution of $Y(t)$ can naturally be extended to a vector of random values $Y(u_1, \ldots, u_n) = [Y(u_1), \ldots, Y(u_n)]^T \in \mathbb{R}^n$. Given an operation $o \in \boldsymbol{O}$, the random vector $Y(u_1, \ldots, u_n)$ follows a multivariate Gaussian distribution with mean and covariance given by

$$\mu_{y,o}^{u_1:u_n} = [\mu_{y,o}(u_1), \ldots, \mu_{y,o}(u_n)]^T \in \mathbb{R}^n \tag{5.29}$$

$$\Sigma_{y,o}^{u_1:u_n} = \begin{bmatrix} \Sigma_{y,o}(u_1,u_1) & \cdots & \Sigma_{y,o}(u_1,u_n) \\ \vdots & \ddots & \vdots \\ \Sigma_{y,o}(u_n,u_1) & \cdots & \Sigma_{y,o}(u_n,u_n) \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{5.30}$$

We have now all the elements to build our generative model for side-channel signals. But before, we present next an algorithm to simulate signals.

5.5 Simulation of side-channel signals

With the model of the algorithm, of the jitter and of the patterns presented in previous sections, we propose the simulation algorithm in Algorithm 2 to generate artificial signals.

Its functioning is as follow: at $t_0 = 0$ an initial dummy operation o_0 is randomly drawn for which no pattern is generated, it serves as a starting operation for the Markov Chain; at iteration i it samples the current operation o_i given o_{i-1} and the delay of the next clock cycle τ_{i+1}. A writing buffer of size $[0, \ldots, T_0]$ is updated with a pattern positioned at t_i and associated with the current operation o_i. The simulation stops and no pattern is generated when the next clock cycle is out of the region of simulation, i.e. when $t_i > T_0$.

The simulation is parametrized by a set of parameters. We explain below their use and how we initialize them:

- The transition matrix $M \in \mathbb{R}^{N_o \times N_o}$ and the vector $\alpha \in \mathbb{R}^{N_o}$ are used to characterize the transitions between operations in the model of Sec. 5.2. We choose them using

$$\alpha \sim \mathrm{Dir}(r_0),\ M[o] \sim \mathrm{Dir}(r_0),\ o \in \boldsymbol{O},\ r_0 \in \mathbb{R}^{+N_o},$$

  with Dir the Dirichlet distribution (Sec. 2.4) and where we noted $M[o]$ the row of probability for the operation o.

- The parameters $k, \lambda \in \mathbb{R}_*^+$ of the Gamma distributed delay of Sec. 5.3.2 are chosen explicitly to study the influence of the jitter.

- The means and covariances $\mu_o, \Sigma_o, o \in \boldsymbol{O}$ for the GMM on the synthesis coefficients are randomly chosen for each operations $o \in \boldsymbol{O}$ using

$$\begin{aligned} &\Sigma_o \sim \mathcal{W}(m, A),\ A \in \mathbb{R}^{m \times m} \text{ pos. def.} \\ &\mu_o \sim \mathcal{N}(0, \epsilon\Sigma_o^{-1})\, \epsilon \in \mathbb{R}^+ \end{aligned}$$

  with A positive definite and where $\mathcal{W}$ is the Wishart distribution (Sec. 2.4) for sampling covariance matrices and m the number of wavelets. We choose the identity for A and epsilon low typically 0.01. It is a common practice to use these distributions to sample the parameters of a Gaussian distribution. The combination of these prior distributions forms the Normal-Wishart distribution.

- And finally, the frame of Generalized Morse Wavelets $\Psi_\Xi \in \mathbb{R}^{n \times m}$ with $\xi = (a, u, \beta, \gamma) \in \Xi$ are initialized by fixing β, γ and by sampling translation and

scaling parameters with:

$$a \sim \mathcal{G}(\alpha_a, k_a),\ u \sim \mathcal{N}(\mu_u, \sigma_u)\,,$$

with $\alpha_a, k_a, \mu_u, \sigma_u$ appropriately chosen in $\mathbb{R}^+$. Additionally, some (a, u) parameters can be rejected if they do not respect some predefined criteria or if they do not led to well defined wavelets. As an example, since our wavelets are normalized to 1 we can check their norm and reject bad (a, u) samples if the norm of $\psi_{a,u,\beta,\gamma}$ is not equal to 1. We choose a Gamma distribution (see Sec. 2.4) to sample the scaling because it is distributed on $\mathbb{R}^+$ and that we can modify its mean and variance easily with α_a, k_a. For the translation parameter, we simply choose a Normal distribution centered in the time region of the pattern, and with a low variance to reduce the chance of sampling translation parameters that will position the wavelets on the border of the time region.

Algorithm 2: Algorithm for the generation of side-channel signals

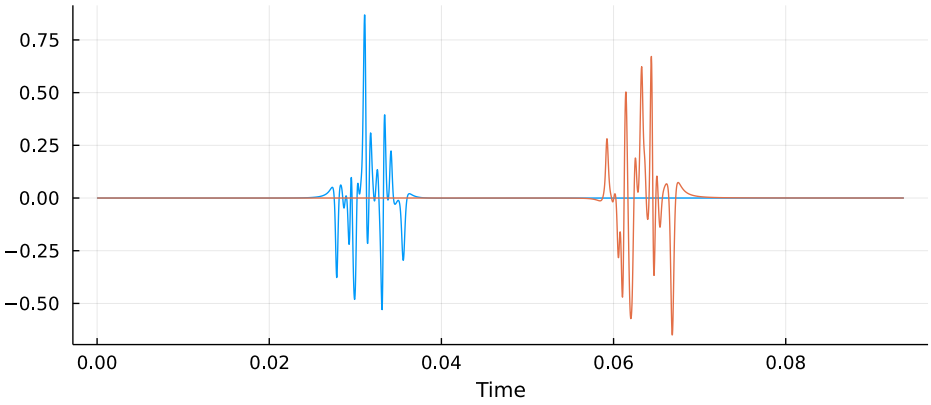
Input: Ψ_Ξ,$\{\mu_i\}_i$,$\{\Sigma_i\}_i$,Σ_e,α,M,k,λ
$Y \leftarrow [0 \dots 0]$ //Size T_0;
$S \leftarrow [0 \dots 0]$ //Size T_0;
$t_0 \leftarrow 0$;
$o_0 \sim \mathrm{Cat}(\alpha)$;
while *True* **do**
  $\tau \sim \mathcal{G}(k, \lambda)$;
  $t_i \leftarrow \tau + t_{i-1}$;
  if $t_i > T_0$ **then**
    break;
  end
  $o_i \sim \mathrm{Cat}(M[o_{i-1}])$;
  $x_i \sim \mathcal{N}(\mu_{o_i}, \Sigma_{o_i})$;
  $Y \leftarrow \Psi_\Xi^\dagger x_i$;
  $Y \leftarrow \mathrm{shift}(Y, t_i)$;
  $S \leftarrow S + Y$;
end
$E \sim \mathcal{N}(0, \Sigma_e)$;
$S \leftarrow S + E$;
return S,$\{t_i\}_i$,$\{o_i\}_i$;

This procedure may be used to simulate side-channel signals and test beforehand the ability of some side-channel methods in the recovery of the states of a Markov Chain. We can think of the states as the value of a sequence of algorithmic operations. The goal of a side-channel attack in this context will be to recover all or part of the states of the Markov Chain to get information on some sensitive variables.

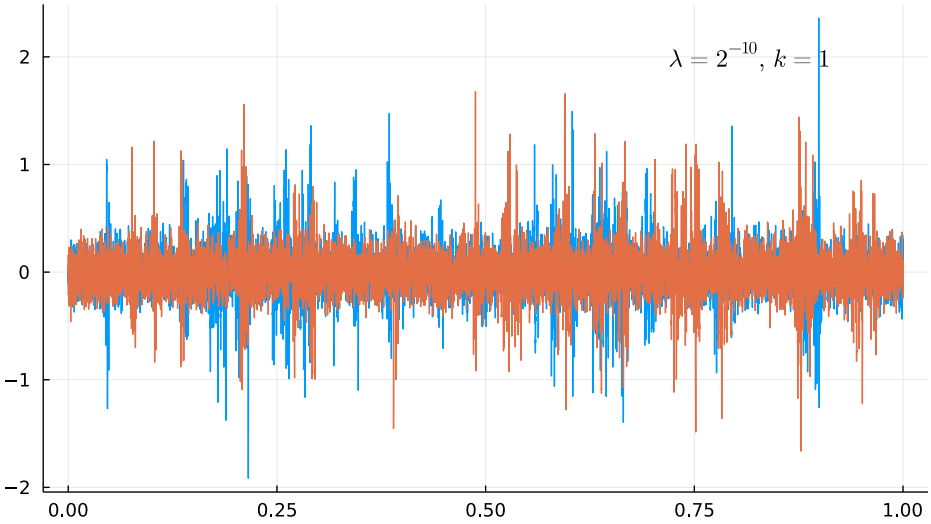
By varying the variance of the Gamma delay or the noise we may test the limit of some side-channel methods against jitter countermeasures.

The use of wavelet frames in this context allows the generation of "smooth" signals and with particular time-frequency properties that can be chosen via the parameters of the frame of Generalized Morse Wavelets. Here, we presented a random initialisation of the translation and scaling parameters of the frame, but β, γ parameters could also be randomly initialized.

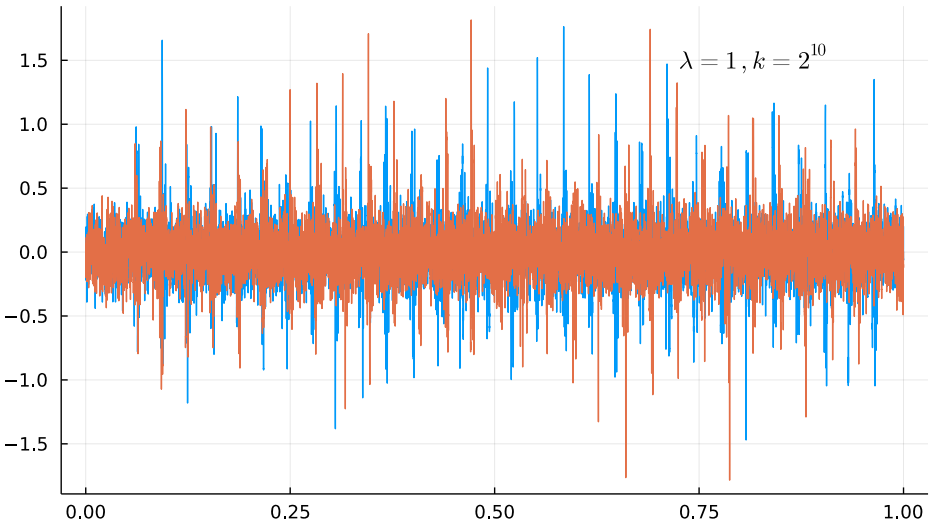
We present on Fig. 5.4 some examples of artificial side-channel signals. The range of simulation is fixed to $T_0 = 2^{15}$ and the hyper-parameters for initialising the parameters of the model are $r_0 = [1, \ldots, 1], A = I, \epsilon = 0.01, \alpha_a = 2.5, k_a = 1.6, \mu_u = 2^10, \sigma_u = 2^5$. The patterns y_o are generated with size 2048. By modifying the variance of the Gamma point process through the hyper-parameters (k, λ), we modify the influence of the jitter. In these simulations, the average delay is constant equal to 2^{10} and we modify the variance by a factor 2^{10}.



(a) Two examples of generated patterns without noise.



(b) Two generated side-channel signals with high jitter. The variance of the random delay is $k/\lambda^2 = 2^{20}$.



(c) Two generated side-channel signals with low jitter. The variance of the random delay is $k/\lambda^2 = 2^{10}$.

Figure 5.4: Simulation of side-channel signals.

5.6 Learning the parameters of the generative model

With the prior models introduced in previous sections on the times of occurrence $\{t_i\}_i$, on the algorithm operations $\{o_i\}$ and on the patterns, we can now construct the full generative model on the signals. The full model will be parametrized by a frame of wavelets Ξ, the means and covariances of the GMM on the synthesis coefficients $\{\mu_o, \Sigma_o\}_{o \in \boldsymbol{O}}$, the parameters of the Gamma point process model for the jitter k, λ, and finally the transition matrix and the prior distribution M, α of the Markov Chain model for characterizing sequences of operations .

Given a dataset of signals $\{s_i\}_i$, the goal is to fit the model to the data and learn the parameters $\Xi, \{\mu_o, \Sigma_o\}_o, k, \lambda, \alpha$ and M. The fitness of the model will be measured during testing by its ability to recover the true times of occurrence $\{t_i\}_i$ and operations $\{o_i\}_i$.

We assume that the parameters of the jitter k and λ are fixed a priori as we do not know yet how to learn those parameters. For the remaining ones, i.e. $\Xi, \{\mu_o, \Sigma_o\}_o, \alpha$ and M, we propose to modify a Hidden Markov Model [121] learning algorithm based on the Expectation Maximization algorithm of Sec. 2.6 that would take into account the gradient descent method for learning a frame proposed in Sec. 4.3.

To be able to recover the times of occurrence $\{t_i\}_i$ and operations $\{o_i\}_i$, we need first to have a look at the prior distribution $p(t_{1:l+1}, o_{1:l})$ and the likelihood $p(s(t)|t_{1:l+1}, o_{1:l})$ to see if they are readily usable.

We will suppose that the number of operations occuring on $[0, T_0]$ is known equal to $l \in \mathbb{N}$. The joint prior distribution of the algorithm and the jitter models is $p(t_{1:l+1}, o_{1:l})$, and by assuming that the jitter is independent of the algorithm, it can be expressed:

$$p(t_{1:l+1}, o_{1:l}) = p(o_1)p(t_1)\prod_{i=2}^{l} p(o_i|o_{i-1})p(t_i|t_{i-1})p(t_{k+1}|t_k)\,, \tag{5.31}$$

while the likelihood of $s(t)$, with (5.1) is expressed:

$$p(s(t)|o_{1:l}, t_{1:l+1}) = p(\sum_{i=1}^{l} y_{o_i}(t - t_i) + e(t)|o_{1:l}, t_{1:l+1}) \tag{5.32}$$

$$= \left(p_{Y(t-t_1)|o_1} \circledast \cdots \circledast p_{Y(t-t_l)|o_l} \circledast p_{E(t)}\right)(s(t)) \tag{5.33}$$

with $p_{Y(t-t_i)}$ given by (5.24) and $p_{E(t)}$ the probability of the additive noise E at time $t \in \mathbb{R}$.

We notice that all patterns involved follow Gaussian distributions, by assum-

ing that the noise $E(t)$ is also Gaussian, we get that $S(t)$ is also Gaussian as the convolution of Gaussian distributions stay Gaussian. Given $\{o_i, t_i\}_i$, its mean and covariance are given by:

$$\mu_s(t) = \sum_{i=1}^{l} (\Psi^* \mu_{o_i})(t - t_i) \tag{5.34}$$

$$\Sigma_s(t,t) = \sum_{i=1}^{l} (\Psi^* \Sigma_{o_i} \Psi)(t - t_i, t - t_i) + C(t,t) \tag{5.35}$$

with $C(t,t)$ the covariance of the noise e. For the case $l = 0$, i.e. when no operation occurs before T_0, we have $s(t) = e(t)$. We also remark that the likelihood does not depend on the last time of occurrence t_{l+1} occuring after T_0, thus we have $p(s(t)|o_{1:l}, t_{1:l+1}) = p(s(t)|o_{1:l}, t_{1:l})$.

Now, we consider that to be able to properly estimate the operations $o_{1:l}$, we first need to estimate the most likely times of occurrence $t_{1:l}$ given a signal s. We are then interested in the posterior $p(t_{1:l+1}|s(t))$ given by:

$$p(t_{1:l+1}|s(t)) = \frac{p(s(t)|t_{1:l})p(t_{1:l+1})}{\int_{\boldsymbol{T}_{T_0}^l} p(s(t), t_{1:l+1}) dt_1 \dots dt_{l+1}} \tag{5.36}$$

but we remark that both the numerator and denominator are intractable. Thus, we need to come up with a sampling strategy to get estimation of the times of occurrence $t_{1:l}$. This will be the goal of Sec. 5.7.

But before, we resume next the learning strategy for estimating the parameters $\Xi, \{\mu_o, \Sigma_o\}_o, \alpha$ and M of the model.

5.6.1 Learning strategy

Assuming for simplicity here that the number of operations is fixed, we consider the following strategy to learn the model given a dataset of signals $\{s_i\}_i$ acquired over a range $[0, T_0]$.

1. Initialize the parameters of the generative model using the same initialisation procedure as in Sec. 5.5, i.e. $\Xi, \{\mu_o, \Sigma_o\}_o, \alpha, M$. The parameters of the jitter are assumed known k, λ.

2. For each signal s_i, get samples of the times of occurrence $(t_1, \dots, t_l)$ and extract a sequence of patterns $(y_1^i, \dots, y_l^i)$.

3. Given $\{(y_1^i, \dots, y_l^i)\}_i$, update $\Xi, \{\mu_o, \Sigma_o\}_o, \alpha, M$ with a modified Hidden Markov Model learning algorithm based on the Expectation Maximization (EM) algorithm of Sec. 2.6, and with the learning method for the frame of Sec. 4.3. At it-

eration j, The goal is to minimize the cross-entropy $C(\tilde{p}_{Y_{1:l}} p_{O_{1:l}|Y_{1:l},\theta_{j-1}}, p_{Y_{1:l},O_{1:l}|\theta})$, with $Y_{1:l}$ the random sequence of patterns, $O_{1:l}$ the random sequence of operations, θ the parameters of the model, i.e. $\Xi, \{\mu_o, \Sigma_o\}_o, \alpha, M$, and θ_{j-1} the estimate of the parameters at previous iteration.

4. Repeat 2-3 until convergence of the loss of the EM algorithm.

5.7 Time occurrence estimation

For simplicity here, we will assume that we already know the number of operations that occurred before T_0. Given the signal s acquired over a period $[0, T_0]$, we propose the following proposal for the posterior distribution to successively sample the times of occurrence $t_{1:l}$:

$$q_{T_i|S,T_{i-1}}(t|s, t_{i-1}) \propto \sum_{o_i} p(o_i) p(Y(0) + E(t) = s(t)|o_i) p_{T_i|T_{i-1}}(t|t_{i-1})\,. \tag{5.37}$$

with $t_0 = 0$, the times of occurrence can be successively sampled, starting with t_1, then t_2 given t_1, and so on until t_k.

This proposal distribution makes the underlying assumption that the patterns are far enough such that $S(t) \sim Y(t - t_i) + E(t)$ when t is close enough to t_i. Under this assumption, we should get a maximum with our proposal distribution when $t = t_i$.

We verify if the sampling strategy is viable by generating side-channel signals with the simulation algorithm of previous section and proceed to the sampling of the proposal (5.37) parametrized with the true parameters of the model on the patterns, i.e. $\Xi, \{\mu_{y,o}, \Sigma_{y,o}\}_{o \in \boldsymbol{O}}$. Then, in the blind case, we discuss on the initialisation of those parameters for learning the model.

In practice, the probability (5.37) is at a time t with a vector of values $[s(t - u(n-1)/2), \ldots, s(t + un/2)]^T \in \mathbb{R}^n, u > 0$. In that case the proposal distribution is simply extended using the ending remark of Sec. 5.4.

5.7.1 Estimation using a Metropolis Hasting algorithm with true generation parameters

Using the Metropolis Hasting algorithm of Sec. 2.7, we get estimations of $\{t_i\}_{i=1}^l$ by successively sampling (5.37) and by taking the candidate with the highest relative probability. The estimates are noted $t_1^*, t_2^*, \ldots, t_l^*$. The performance of the method is evaluated with an increasing jitter. With the Gamma point process of Sec. 5.3.2 with parameters k and λ, we fix the mean of the delay to $\mu_0 = k/\lambda$ and increase its

$\log_2(\sigma/\mu_0)$	2	4	8	10
$\lvert t - t^* \rvert / \mu_0$	1.02e-2±8.2e-3	1.32e-2±1.5e-2	6.80e-1±5.1e-1	1.53±1.5

Table 5.1: Errors in the estimation of the times of occurrence. We keep the mean μ_0 of the Gamma distributed delay constant and increase its variance σ.

variance $\sigma = k/\lambda^2$. In our experiments, μ_0 is fixed to 2^{11} and we vary λ from 1 to 2^{-10}. We use a Normal distribution with a variance of 500 for the sampling prior distribution in the Metropolis Hastings algorithm. We simulate 100 signals of size $T_0 = 2^{15}$. With the average delay being $\mu_0 = 2^{11}$, it gives 16 patterns in average by signal, and thus to approximately 1600 times of occurrence to estimate.

If we note t^* the estimation of the time of occurrence t, the error is evaluated using

$$\frac{|t - t^*|}{\mu_0}, \tag{5.38}$$

it measures the error of estimation in proportion to the average delay μ_0.

The frequency operation of the system is $1/\mu_0$, an error of 1 in (5.38) thus means that we completely missed a cycle of operation.

We show results in Tab. 5.1 and on Fig. 5.6. We remark that the method is robust up to a jitter of normalized variance $\sigma/\mu_0 = 2^4$. After this, the error quickly diverges. This is explained by the fact that with a high jitter, two operations may get close enough and the proposal distribution is no more valid. In that case, the proposal should take into account colliding patterns. However, in the context of side-channel analysis this method presents promising results as patterns should be far enough.

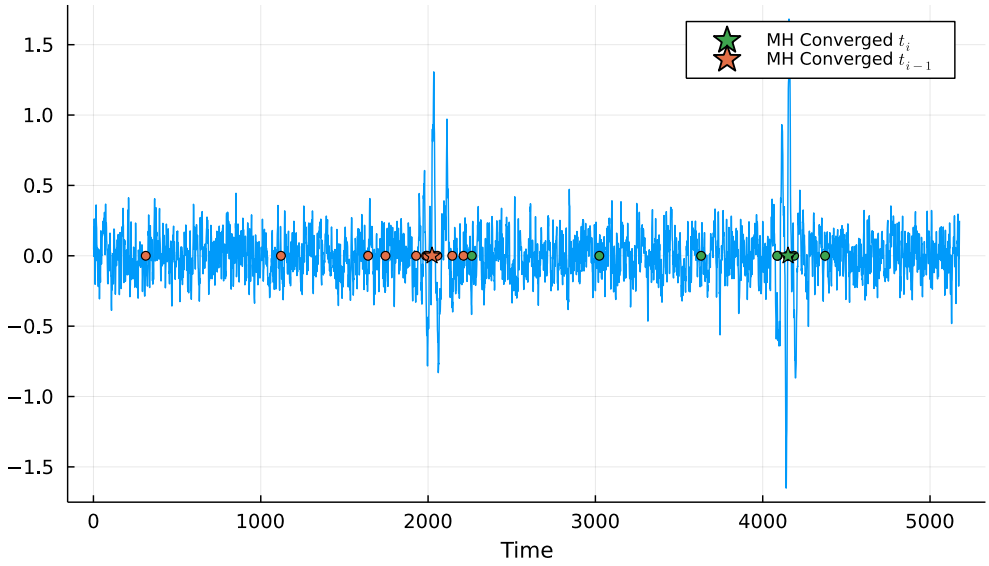
We illustrate on Fig.5.5 how the Metropolis Hastings algorithm converges to estimates of times of occurrence.

The proposal was parametrized with the true parameters used for generating the signals. We see that the times of occurrence can be recovered using the method proposed when the jitter is not too high. We study next in which condition this procedure can be employed when the parameters are unknown.

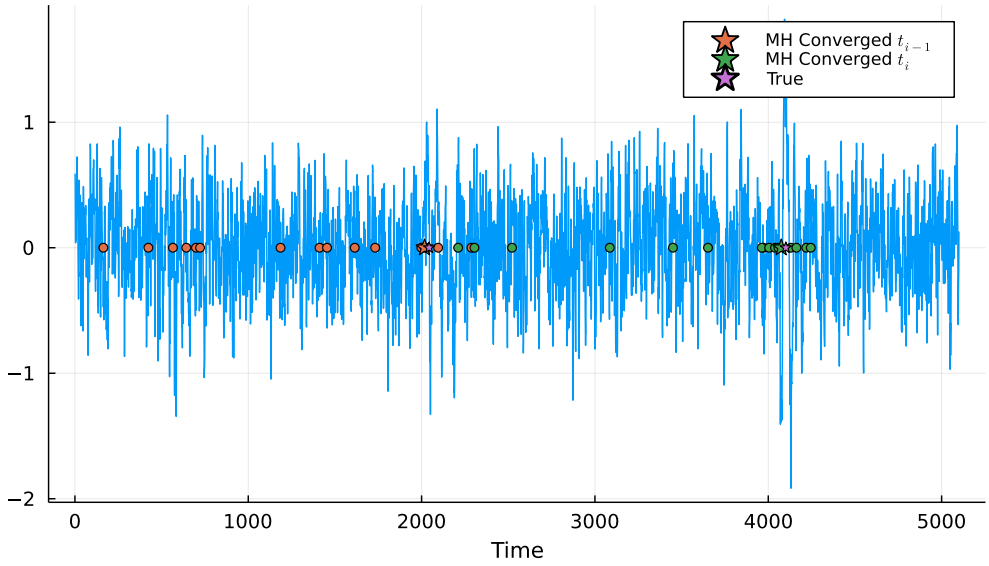
5.7.2 Estimation with unknown parameters

We discuss here how to initialize the parameters of the proposal in order to get good samples when using the learning strategy presented in previous section. The proposal (5.37) depends on the parameters of the GMM $\{\mu_{y,o}, \Sigma_{y,o}\}_{o \in \boldsymbol{O}}$ and how they are initialized. We show in Fig. 5.7 the evolution of

$$\log q(t_1|s) - \log p(t_1|t_0) = \log \left[\sum_{o_i} p(o_i) p(Y(0) + E(t_1) = s(t_1)|o_i) \right]$$



(a) Noise at −5dB .



(b) Noise at −1dB.

Figure 5.5: Sampling the times of occurrence of operations with Metropolis Hastings.

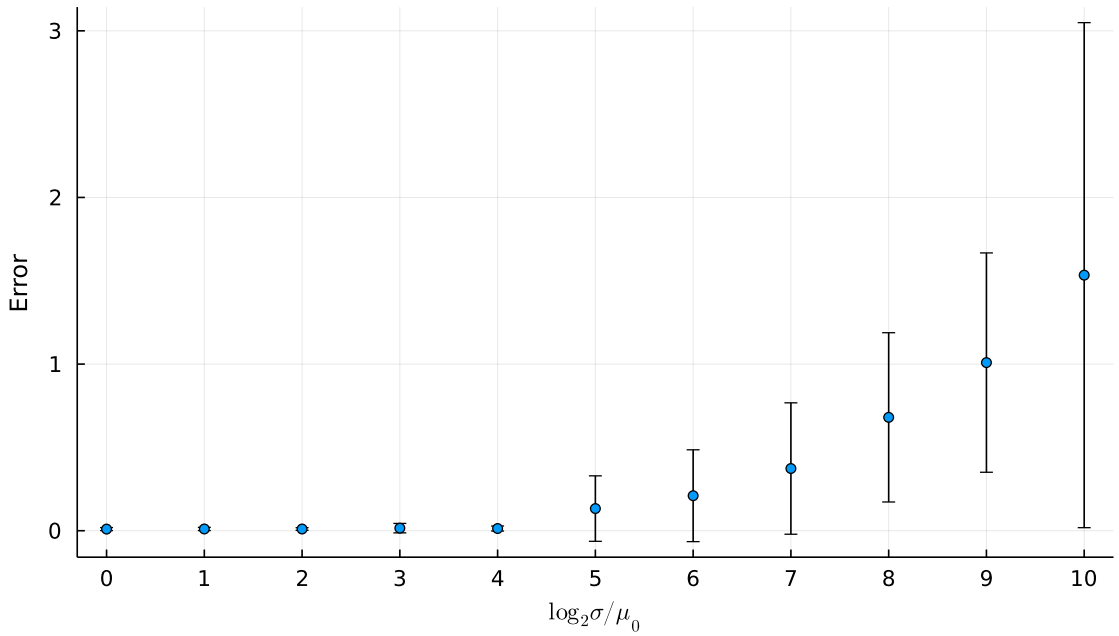


Figure 5.6: Evolution of the error of estimation as a function of the variance of the jitter.

for two different initialisations of the parameters.

We can adopt a naive approach and initialize the covariances of the GMM with a Wishart distributions or we can use the initialisation strategy of Sec. 5.5 employing wavelet frames to factorize the covariance matrices. For both approaches, the means are fixed to null vectors, as in practice we do not assume beforehand a specific polarity to signals.

We remark that with the naive approach we do not detect any bumps corresponding to the presence of patterns while the initialisation with wavelet frames allows a better chance of sampling "good" times of occurrence using the Metropolis Hastings algorithm. It suggests first that the learning strategy may be viable if the sampling succeeds to get correct time of occurrence values. Also, for the fitting of this model with real side-channel signals it indicates how to initialize those parameters to ensure the success of the learning strategy. This approach is reserved for further work, we aim at studying the strategy exposed here for the analysis of real side-channel signals.

We assume that these preliminary results on artificial signals can be transposed on real side-channel signals. The next important work is to verify in the blind case if the learning strategy is correct on artificial signals, and then to test it with real signals.

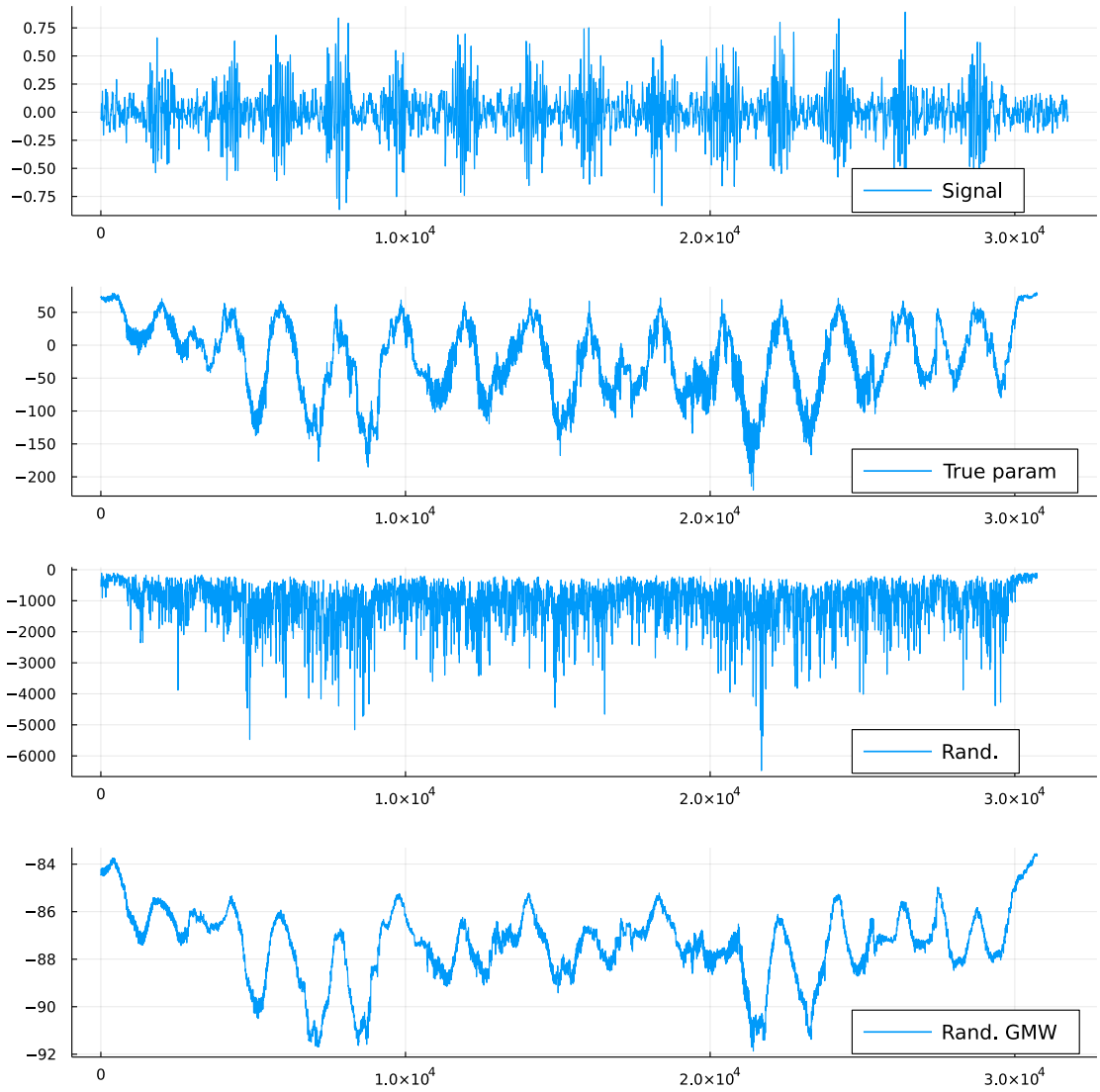


Figure 5.7: Evaluation of the log proposal for the localization of patterns. We removed the weight of the prior distribution on τ to observe the log probability of the GMM model. At the top a simulated signal using the method presented Sec. 5.5. Below the log probability of the GMM using the true parameters. Third row, the log probability with a GMM with randomly initialized covariance matrices using Wishart distributions. At the bottom, the log probability with of a GMM with randomly initialized covariance matrices factorized with random frames of Generalized Morse Wavelets.

5.8 Conclusion

We presented a generative model for side-channel signals that includes a model of the algorithm in the form of a Hidden Markov Model, and a model of the jitter as a Gamma point process. For generating patterns we use the model presented in Sec. 4.3 which employs wavelet frames to encode particular time-frequency properties, and with the synthesis coefficients modeled using a Gaussian Mixture Model.

We apply this model first for simulating side-channel signals. We are able to simulate artificial signals with controllable properties such as; the parameters of the Markov Chain to change the most likely sequences of operations; the jitter to perturb the localization of information; and the time-frequency properties of the patterns. We envisage the simulation of side-channel signals to test the limit of state-of-the-art side-channel attacks, and anticipate new countermeasures without requiring an actual implementation.

We presented a learning strategy for the model in which an estimation of the times of occurrence is required. Thus, we derived a method with a Metropolis Hastings algorithm to sample the times of occurrence of the operations (states of the Markov Chain). Given the true parameters of the GMM, the method is efficient to recover the times of occurrence up to a very high variance of the jitter. In the blind case preliminary results suggest that for good parameter initialisation with frames of wavelets the method could also be used. We plan in a further work to test the learning strategy in the blind case with artificial signals and then implement this strategy to fit the model to real side-channel signals. After convergence of the model on a training set of side-channels signals, it could be used to estimate the times of occurrence and the value of the operations.

Conclusion

In this thesis we presented different uses of wavelet analysis for improving side-channel attacks.

From a practical point of view first, the visualization of side-channel signals through wavelet transforms allows a better understanding of the signal structure. It provides a multiresolution time-frequency map that can be used to distinguish the patterns related to the algorithmic operations. The evaluator can use these new representations to see the signal from another perspective and identify the different operations related to the cryptographic algorithm. In Sec. 4.2, we presented a method for allowing an evaluator to extract time-frequency patterns from scalograms, perform a denoising, and resynthesize them into adapted filters. By intercorrelating the signal with those patterns, the evaluator is able to recover in the signal similar patterns in the time domain. This simple approach allows the evaluator to perform an on-the-fly study and quickly realign signals before using more advanced methods.

A wavelet basis is often chosen such as to grasp most of any type of signals. Thus, a large part of the basis is generally not used. The study of methods for the estimation of adapted frames for the analysis of signals is an ongoing topic that is well worth investigating. We have studied this approach in the context of side-channel analysis, and presented in Sec. 4.3 a method for estimating a frame of Generalized Morse Wavelets adapted to a dataset of patterns. We formulated our problem as factor analysis problem and solved it via a Maximum Likelihood approach. By iteratively optimising the likelihood loss through gradient descent, we are able to continuously learn an adapted frame of wavelets. We next applied this frame as a dimension reduction technique for compressing patterns before canonical template attacks. This work has been published in [37].

Next, we presented an attack method with a more direct approach for tackling jitter countermeasures in signals. Instead of performing realignment, we presented an attack method using the scattering transform [81] for mapping signals to representations stable under small translation and deformation. This way, we may use simple attack methods such as template attacks against jitter protected devices. This technique has no particular cost of implementation and preserves the informa-

tion. It can be used as a preprocessing step for any type of classification method. In a secondary work, we also presented an ensemble method for approximating the leakage model of a sensitive variable. Instead of considering one particular leakage model, such as the Hamming weight of a sensitive variable, a set of classifiers are trained in parallel with new labelings of the sensitive variable, and we proposed a method to merge their results to produce a global a posteriori probability. This technique allows a better understanding of what part of the sensitive variable is leaking. The use of the scattering transform and the ensemble method has been published in [38].

Finally, in an opening chapter we presented a novel model for side-channel signals. The main motivation behind this is to provide an end-to-end framework for assessing the security of the device and identify the source of the leak. The model is built around three main parts; a model of the algorithm, a model of the jitter, and a generating model for patterns related to algorithmic operations. In particular, we propose to model the jitter as a Gamma point process, it allows to represent a continuous random delay between operations. We reemployed the model used for the estimation of frames of wavelets from Chap. 4. The patterns are generated through a frame of wavelets and with synthesis coefficients following a Gaussian Mixture Model. The overall model allows the simulation of side-channel signals that may be used to test the limit of some side-channel attacks or to design countermeasures without requiring an actual implementation. The use of wavelet frames in this context allows the generation of patterns with specific time-frequency properties. Finally, in a last part, we study the fitting of the model to real data. We envisage a learning strategy based on Expectation Maximization that requires a proper estimation of the times of occurrence beforehand. For that reason, we focused in a last part on a sampling strategy for the recovery of the times of occurrence of the operations.

Perspectives

The proposed work on the estimation of frames of Generalized Morse Wavelets may be improved by studying the Fisher information matrix. The study of the sensibility of the Maximum likelihood loss according to the parameters of the frame could provide ways to improve the gradient updates during the optimisation of the loss. We also envisage in a further work to transpose the problem in the time domain to take into account the non-continuity at the border of cropped patterns in side-channel signals. The direct control we have over the parameters of the frame allows us to easily add regularisation terms in the loss. This approach could be studied to incorporate additional prior information on the time-frequency properties of patterns.

The scattering transform has been used with deep neural networks in hybrid architectures in [95] for visual recognition task. A similar approach could be envisaged in side-channel attacks as a preprocessing step for convolutional neural networks. Also, an extension of the scattering transform, namely the Joint Time-Frequency scattering transform, has been proposed in [5] for audio classification tasks, its study could be proposed for the analysis of side-channel signals. Finally, our work on the estimation of a frame could be coupled with the scattering transform. Indeed, since the scattering transform is based on a fixed basis, we could use our work on Generalized Morse Wavelet frame estimation to learn an adapted frame of wavelet for the scattering transform. This approach could be compared with traditional convolutional neural network to see if it allows us to gain some performance or learning stability.

The generative model of last chapter is still an important ongoing work. It is at the crossroad of wavelet analysis, state-space models and stochastic processes. Further readings on the latter field may bring us new ideas to model the jitter and drive a better estimation method of the times of occurrence. We are aware that other types of point process model exist in the literature on stochastic process. The underlying methods to study point processes and their models wait to be applied in the side-channel analysis context. As a direct improvement of the generative model, a study of the dependence between the cryptographic algorithm and the jitter could be carried. In practice, some jitter countermeasures only activate upon the execution of a critical part of the algorithm. Thus, to properly understand a side-channel signal as a whole we have to take this dependence into account. This may require to drop the Hidden Markov Model to represent the algorithm and use its Factor Graph representation. Also, in our work, we proposed a model for a continuous jitter. A discrete type of jitter with a fixed delay for representing dummy operations could be considered. Finally, we think that this model could be further improved to test the limit of state-of-the-art side-channel methods by providing quantitative measures of their performance against the generation parameters of the model, with the aim that new software or hardware countermeasures may be developed to anticipate attackers.

Bibliography

[1] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Vol. 55. US Government printing office, 1948.

[2] Paul S Addison. "Wavelet transforms and the ECG: a review". In: *Physiological measurement* 26.5 (2005), R155.

[3] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. "The EM Side-Channel(s)". In: *Cryptographic Hardware and Embedded Systems - CHES 2002*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 29–45.

[4] Shun-ichi Amari. *Information geometry and its applications*. Vol. 194. Springer, 2016.

[5] Joakim Andén, Vincent Lostanlen, and Stéphane Mallat. "Joint Time-Frequency Scattering for Audio Classification". In: *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)* (Sept. 2015). arXiv: 1512.02125.

[6] Joakim Andén and Stéphane Mallat. "Deep Scattering Spectrum". In: *IEEE Transactions on Signal Processing* 62.16 (2014), pp. 4114–4128.

[7] Joakim Andén and Stéphane Mallat. "Deep scattering spectrum". In: *IEEE Transactions on Signal Processing* 62.16 (2014), pp. 4114–4128.

[8] Mathieu Andreux, Tomás Angles, Georgios Exarchakis, Roberto Leonarduzzi, Gaspar Rochette, Louis Thiry, John Zarka, Stéphane Mallat, Joakim Andén, Eugene Belilovsky, Joan Bruna, Vincent Lostanlen, Matthew J. Hirn, Edouard Oyallon, Sixhin Zhang, Carmine-Emanuele Cella, and Michael Eickenberg. "Kymatio: Scattering Transforms in Python". In: *CoRR* abs/1812.11214 (2018). arXiv: 1812.11214.

[9] R. Jacob Baker. *CMOS: Circuit Design, Layout and Simulation*. John Wiley & Sons, 2019.

[10] P. Balazs, M. Dörfler, F. Jaillet, N. Holighaus, and G. Velasco. “Theory, Implementation and Applications of Nonstationary Gabor Frames”. In: *Journal of Computational and Applied Mathematics* 236.6 (Oct. 15, 2011).

[11] Riccardo Barbieri, Eric Matten, Abdulrasheed Alabi, and Emery Brown. “A point-process model of human heartbeat intervals: New definitions of heart rate and heart rate variability”. In: *American journal of physiology. Heart and circulatory physiology* 288 (Feb. 2005), H424–35.

[12] Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. “Mutual information analysis: a comprehensive study”. In: *Journal of Cryptology* 24.2 (2011), pp. 269–291.

[13] Pierre Belgarric, Shivam Bhasin, Nicolas Bruneau, Jean-Luc Danger, Nicolas Debande, Sylvain Guilley, Annelie Heuser, Zakaria Najm, and Olivier Rioul. “Time-Frequency Analysis for Second-Order Attacks”. In: *Smart Card Research and Advanced Applications*. Ed. by Aurélien Francillon and Pankaj Rohatgi. Springer International Publishing, 2014, pp. 108–122.

[14] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. “Mutual information neural estimation”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 531–540.

[15] Ryad Benadjila, Emmanuel Prouff, Rémi Strullu, Eleonora Cagli, and Cécile Dumas. “Deep learning for side-channel analysis and introduction to ASCAD database”. In: *Journal of Cryptographic Engineering* 10.2 (2020), pp. 163–188.

[16] L. Benini, E. Omerbegovic, A. Macii, M. Poncino, E. Macii, and F. Pro. “Energy-aware design techniques for differential power analysis protection”. In: *Proceedings 2003. Design Automation Conference IEEE*. 2003, pp. 36–41.

[17] Pierre Brémaud. *Mathematical principles of signal processing: Fourier and wavelet analysis*. Springer, 2002.

[18] Eric Brier, Christophe Clavier, and Francis Olivier. “Correlation Power Analysis with a Leakage Model”. In: *Cryptographic Hardware and Embedded Systems - CHES 2004*. Ed. by Marc Joye and Jean-Jacques Quisquater. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 16–29.

[19] Eleonora Cagli. “Feature Extraction for Side-Channel Attacks”. PhD thesis. Sorbonne Université, Dec. 2018.

[20] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. “Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures”. In: *Cryptographic Hardware and Embedded Systems – CHES 2017*. Ed. by Wieland Fischer and Naofumi Homma. Springer International Publishing, 2017, pp. 45–68.

[21] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. “Template Attacks”. In: *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*. 2002, pp. 13–28.

[22] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. “Template Attacks”. In: *Cryptographic Hardware and Embedded Systems - CHES 2002* (Berlin, Heidelberg). Springer Berlin Heidelberg, 2003.

[23] Xavier Charvet and Herve Pelletier. “Improving the DPA attack using Wavelet transform”. In: *NIST Physical Security Testing Workshop*. Vol. 46. 2005.

[24] Omar Choudary and Markus G Kuhn. “Efficient template attacks”. In: *International Conference on Smart Card Research and Advanced Applications*. Springer. 2013, pp. 253–270.

[25] Ole Christensen. *An Introduction to Frames and Riesz Bases*. Applied and Numerical Harmonic Analysis. Birkhauser,Cham, 2016.

[26] Christophe Clavier, Jean-Sébastien Coron, and Nora Dabbous. “Differential Power Analysis in the Presence of Hardware Countermeasures”. In: *Cryptographic Hardware and Embedded Systems — CHES 2000*. Ed. by Çetin K. Koç and Christof Paar. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 252–263.

[27] Jean-Sébastien Coron and Ilya Kizhvatov. “Analysis and improvement of the random delay countermeasure of CHES 2009”. In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2010, pp. 95–109.

[28] Valence Cristiani, Maxime Lecomte, and Philippe Maurine. “Leakage assessment through neural estimation of the mutual information”. In: *International Conference on Applied Cryptography and Network Security*. Springer. 2020, pp. 144–162.

[29] Cryptome. *NSA TEMPEST Documents*. https://cryptome.org/nsa-tempest.htm. 2014.

[30] Joan Daemen and Vincent Rijmen. “The Block Cipher Rijndael”. In: *Smart Card Research and Applications*. Ed. by Jean-Jacques Quisquater and Bruce Schneier. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 277–284.

[31] Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.

[32] Ingrid Daubechies. "Time-frequency localization operators: a geometric phase space approach". In: *IEEE Transactions on Information Theory* 34.4 (1988), pp. 605–612.

[33] Ingrid Daubechies and Thierry Paul. "Time-frequency localisation operators-a geometric phase space approach: II. The use of dilations". In: *Inverse problems* 4.3 (1988), p. 661.

[34] Mark HA Davis. *Markov models & optimization*. Routledge, 1993.

[35] Nicolas Debande, Y. Souissi, M. A. E. Aabid, S. Guilley, and J. Danger. "Wavelet transform based pre-processing for side channel analysis". In: *2012 45th Annual IEEE/ACM International Symposium on Microarchitecture Workshops*. 2012, pp. 32–38.

[36] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1 (1977), pp. 1–38.

[37] Gabriel Destouet, Cecile Dumas, Anne Frassati, and Valerie Perrier. "Generalized Morse Wavelet frame Estimation applied to Side-Channel Analysis". In: *2021 6th International Conference on Frontiers of Signal Processing*. 2021.

[38] Gabriel Destouet, Cécile Dumas, Anne Frassati, and Valérie Perrier. "Wavelet Scattering Transform and Ensemble Methods for Side-Channel Analysis". In: *Constructive Side-Channel Analysis and Secure Design*. Ed. by Guido Marco Bertoni and Francesco Regazzoni. Cham: Springer International Publishing, 2021, pp. 71–89.

[39] R. J. Duffin and A. C. Schaeffer. "A Class of Nonharmonic Fourier Series". In: *Transactions of the American Mathematical Society* 72.2 (1952), pp. 341–366.

[40] François Durvaux, Mathieu Renauld, François-Xavier Standaert, Loic van Oldeneel tot Oldenzeel, and Nicolas Veyrat-Charvillon. "Efficient Removal of Random Delays from Embedded Software Implementations Using Hidden Markov Models". In: *Smart Card Research and Advanced Applications*. Ed. by Stefan Mangard. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 123–140.

[41] Thomas Eisenbarth, Christof Paar, and Björn Weghenkel. “Building a Side Channel Based Disassembler”. In: *Transactions on Computational Science X: Special Issue on Security in Computing, Part I*. Ed. by Marina L. Gavrilova, C. J. Kenneth Tan, and Edward David Moreno. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 78–99.

[42] Conal Elliott. *The simple essence of automatic differentiation*. 2018. eprint: 1804.00746.

[43] Dennis Gabor. “Theory of communication. Part 1: The analysis of information”. In: *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering* 93.26 (1946), pp. 429–441.

[44] Catherine H Gebotys, Simon Ho, and Chin Chi Tiu. “EM analysis of rijndael and ECC on a wireless java-based PDA”. In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2005, pp. 250–264.

[45] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Deep sparse rectifier neural networks”. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 315–323.

[46] N. R. Goodman. “Statistical Analysis Based on a Certain Multivariate Complex Gaussian Distribution (An Introduction)”. In: *Annals of Mathematical Statistics* 34.1 (Mar. 1963), pp. 152–177.

[47] A. Grossmann and J. Morlet. “Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape”. In: *SIAM Journal on Mathematical Analysis* 15.4 (1984), pp. 723–736.

[48] B.B Gupta and Megha Quamara. *Smart Card Security. Applications, Attacks and Countermeasures*. Taylor and Francis Group, 2020.

[49] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*. Springer, 2009.

[50] Annelie Heuser, Olivier Rioul, and Sylvain Guilley. “Good Is Not Good Enough”. In: *Cryptographic Hardware and Embedded Systems – CHES 2014*. Ed. by Lejla Batina and Matthew Robshaw. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 55–74.

[51] A. Hjorungnes and D. Gesbert. “Complex-Valued Matrix Differentiation: Techniques and Key Results”. In: *IEEE Transactions on Signal Processing* 55.6 (2007), pp. 2740–2746.

[52] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. “Stochastic variational inference”. In: *The Journal of Machine Learning Research* 14.1 (2013), pp. 1303–1347.

[53] Naofumi Homma, Sei Nagashima, Yuichi Imai, Takafumi Aoki, and Akashi Satoh. “High-Resolution Side-Channel Attack Using Phase-Based Waveform Matching”. In: *Cryptographic Hardware and Embedded Systems - CHES 2006*. Ed. by Louis Goubin and Mitsuru Matsui. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 187–200.

[54] Gabriel Hospodar, Benedikt Gierlichs, Elke De Mulder, Ingrid Verbauwhede, and Joos Vandewalle. “Machine learning in side-channel analysis: a first study”. In: *Journal of Cryptographic Engineering* 1.4 (2011), p. 293.

[55] Michael Innes. *Don’t Unroll Adjoint: Differentiating SSA-Form Programs*. 2019. eprint: `1810.07951`.

[56] Michael Innes, Elliot Saba, Keno Fischer, Dhairya Gandhi, Marco Concetto Rudilosso, Neethu Mariya Joy, Tejan Karmali, Avik Pal, and Viral Shah. “Fashionable Modelling with Flux”. In: *CoRR* abs/1811.01457 (2018). eprint: `1811.01457`.

[57] J. Irwin, D. Page, and N.P. Smart. “Instruction stream mutation for non-deterministic processors”. In: Feb. 2002, pp. 286–295.

[58] Yuval Ishai, Amit Sahai, and David Wagner. “Private circuits: Securing hardware against probing attacks”. In: *Annual International Cryptology Conference*. Springer. 2003, pp. 463–481.

[59] David J.Thomson. “Spectrum Estimation and Harmonic Analysis”. In: *Proceedings of the IEEE* (1982).

[60] Ian T Jolliffe. “Principal components in regression analysis”. In: *Principal component analysis*. Springer, 1986, pp. 129–155.

[61] *JuliaLinearAlgebra/IterativeSolvers.jl: v0.9.1*. Version v0.9.1. 2021.

[62] E.F. Kaasschieter. “Preconditioned conjugate gradients for solving singular systems”. In: *Journal of Computational and Applied Mathematics* 24.1 (1988), pp. 265–275.

[63] Chris Karlof and David Wagner. “Hidden Markov Model Cryptanalysis”. In: *Cryptographic Hardware and Embedded Systems - CHES 2003*. Ed. by Colin D. Walter, Çetin K. Koç, and Christof Paar. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 17–34.

[64] John Kelsey, Bruce Schneier, David Wagner, and Chris Hall. “Side channel cryptanalysis of product ciphers”. In: *Computer Security — ESORICS 98*. Ed. by Jean-Jacques Quisquater, Yves Deswarte, Catherine Meadows, and Dieter Gollmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 97–110.

[65] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. “Spectre Attacks: Exploiting Speculative Execution”. In: *40th IEEE Symposium on Security and Privacy (S&P'19)*. 2019.

[66] Paul Kocher, Joshua Jaffe, and Benjamin Jun. “Differential power analysis”. In: *Annual international cryptology conference*. Springer. 1999, pp. 388–397.

[67] Paul C. Kocher. “Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems”. In: *Advances in Cryptology — CRYPTO '96*. Ed. by Neal Koblitz. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 104–113.

[68] Steffen L Lauritzen. *Graphical models*. Vol. 17. Clarendon Press, 1996.

[69] Thanh-Ha Le. “Analyses et mesures avancées du rayonnement électromagnétique d'un circuit intégré”. PhD thesis. 2007.

[70] Hélène Le Bouder, Ronan Lashermes, Yanis Linge, Gaël Thomas, and Jean-Yves Zie. “A multi-round side channel attack on aes using belief propagation”. In: *International Symposium on Foundations and Practice of Security*. Springer. 2016, pp. 199–213.

[71] Liran Lerman, Romain Poussier, Gianluca Bontempi, Olivier Markowitch, and François-Xavier Standaert. “Template attacks vs. machine learning revisited (and the curse of dimensionality in side-channel analysis)”. In: *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer. 2015, pp. 20–33.

[72] J.M. Lilly and S.C. Olhede. “Higher-Order Properties of Analytic Wavelets”. In: *IEEE Transactions on Signal Processing* 57.1 (2009), pp. 146–160.

[73] Jonathan M Lilly and Sofia C Olhede. “Generalized Morse wavelets as a superfamily of analytic wavelets”. In: *IEEE Transactions on Signal Processing* 60.11 (2012), pp. 6036–6041.

[74] Yanis Linge. “Etudes cryptographiques et statistiques de signaux compromettants”. PhD thesis. Université de Grenoble, Nov. 2013.

[75] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg. "Meltdown: Reading Kernel Memory from User Space". In: *27th USENIX Security Symposium (USENIX Security 18)*. 2018.

[76] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1999.

[77] David JC MacKay and David JC Mac Kay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[78] Houssem Maghrebi, Thibault Portigliatti, and Emmanuel Prouff. "Breaking Cryptographic Implementations Using Deep Learning Techniques". In: *Security, Privacy, and Applied Cryptography Engineering - 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings*. 2016, pp. 3–26.

[79] Stéphane Mallat. "A theory for multiresolution signal decomposition: the wavelet representation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11.7 (1989), pp. 674–693.

[80] Stéphane Mallat. *A wavelet tour of signal processing*. Third. Elsevier, 2009.

[81] Stéphane Mallat. "Group invariant scattering". In: *Communications on Pure and Applied Mathematics* 65.10 (2012), pp. 1331–1398.

[82] Zdenek Martinasek, Jan Hajny, and Lukas Malina. "Optimization of Power Analysis Using Neural Network". In: *Smart Card Research and Advanced Applications*. Ed. by Aurélien Francillon and Pankaj Rohatgi. Cham: Springer International Publishing, 2014, pp. 94–107.

[83] James L Massey. "Guessing and entropy". In: *Proceedings of 1994 IEEE International Symposium on Information Theory*. IEEE. 1994, p. 204.

[84] Loic Masure. "Towards a Better Comprehension of Deep Learning for Side-Channel Analysis". PhD thesis. Sorbonne University, France, 2020.

[85] Loïc Masure, Cécile Dumas, and Emmanuel Prouff. "A Comprehensive Study of Deep Learning for Side-Channel Analysis". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020.1 (Nov. 2019), pp. 348–375.

[86] Mitsuru Matsui. "Linear cryptanalysis method for DES cipher". In: *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer. 1993, pp. 386–397.

[87] David May, Henk Muller, and Nigel Smart. "Non-deterministic Processors". In: vol. 2119. July 2001, pp. 115–129.

[88] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2020. eprint: 1802.03426.

[89] S. Moore, R. Anderson, P. Cunningham, R. Mullins, and G. Taylor. "Improving smart card security using self-timed circuits". In: *Proceedings Eighth International Symposium on Asynchronous Circuits and Systems*. 2002, pp. 211–218.

[90] Simon Moore, Ross Anderson, Robert Mullins, George Taylor, and Jacques JA Fournier. "Balanced self-checking asynchronous logic for smart card applications". In: *Microprocessors and Microsystems* 27.9 (2003), pp. 421–430.

[91] Ruben A Muijrers, Jasper GJ van Woudenberg, and Lejla Batina. "RAM: Rapid alignment method". In: *International Conference on Smart Card Research and Advanced Applications*. Springer. 2011, pp. 266–282.

[92] Robb J Muirhead. *Aspects of multivariate statistical theory*. Vol. 197. John Wiley & Sons, 2009.

[93] Maxime Nassar, Youssef Souissi, Sylvain Guilley, and Jean-Luc Danger. "RSM: A small and fast countermeasure for AES, secure against 1st and 2nd-order zero-offset SCAs". In: *2012 Design, Automation & Test in Europe Conference & Exhibition, DATE 2012, Dresden, Germany, March 12-16, 2012*. Ed. by Wolfgang Rosenstiel and Lothar Thiele. IEEE, 2012, pp. 1173–1178.

[94] Sofia C Olhede and Andrew T Walden. "Generalized morse wavelets". In: *IEEE Transactions on Signal Processing* 50.11 (2002), pp. 2661–2670.

[95] Edouard Oyallon, Eugene Belilovsky, and Sergey Zagoruyko. "Scaling the scattering transform: Deep hybrid networks". In: *IEEE International conference on computer vision*. 2017.

[96] Athanasios Papoulis. *Foundations of signal processing*. McGraw-Hill, 1977.

[97] Athanasios Papoulis and H Saunders. "Probability, random variables and stochastic processes". In: (2002).

[98] Emanuel Parzen. *Stochastic processes*. SIAM, 1999.

[99] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[100] Thomas Plos, Michael Hutter, and Martin Feldhofer. "Evaluation of side-channel preprocessing techniques on cryptographic-enabled HF and UHF RFID-tag prototypes". In: *Workshop on RFID Security*. 2008, pp. 114–127.

[101] S. James Press and K. Shigemasu. “Bayesian Inference in Factor Analysis”. In: *Contributions to Probability and Statistics: Essays in Honor of Ingram Olkin*. Ed. by Leon Jay Gleser, Michael D. Perlman, S. James Press, and Allan R. Sampson. New York, NY: Springer New York, 1989, pp. 271–287.

[102] Emmanuel Prouff and Matthieu Rivain. “Masking against Side-Channel Attacks: A Formal Security Proof”. In: *Advances in Cryptology – EUROCRYPT 2013*. Ed. by Thomas Johansson and Phong Q. Nguyen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 142–159.

[103] Nilson Report. *Card Fraud WorldWide*. `https://nilsonreport.com/publication_chart_and_graphs_archive.php`. 2015.

[104] Walter Rudin. *Real and Complex Analysis, 3rd Ed.* USA: McGraw-Hill, Inc., 1987.

[105] Masa-aki Sato. “Online Model Selection Based on the Variational Bayes”. In: *Neural Computation* 13.7 (July 2001), pp. 1649–1681.

[106] Werner Schindler, Kerstin Lemke, and Christof Paar. “A Stochastic Model for Differential Side Channel Cryptanalysis”. In: *Cryptographic Hardware and Embedded Systems – CHES 2005*. Ed. by Josyula R. Rao and Berk Sunar. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 30–46.

[107] K. M. Shelfer and J. D. Procaccino. “Smart Card Evolution”. In: *Communcation of the ACM* 45.7 (1999), pp. 83–88.

[108] David Slepian. “Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - V: The Discrete Case”. In: *The Bell System Technical Journal* (1978).

[109] David Slepian. “Some comments on fourier analysis, uncertainty and modeling”. In: *SIAM Review* (1983).

[110] Vaclav Smidl and Anthony Quinn. *The variational Bayes method in signal processing*. Springer Science & Business Media, 2006.

[111] E. W. Stacy. “A Generalization of the Gamma Distribution”. In: *Ann. Math. Statist.* 33.3 (Sept. 1962), pp. 1187–1192.

[112] François-Xavier Standaert. “How (Not) to Use Welch’s T-Test in Side-Channel Security Evaluations”. In: *Smart Card Research and Advanced Applications*. Ed. by Begül Bilgin and Jean-Bernard Fischer. Cham: Springer International Publishing, 2019, pp. 65–79.

[113] Douglas R. Stinson. *Cryptography: theory and practice*. 3rd ed. The CRC Press series on discrete mathematics and its applications. Chapman & Hall/CRC, 2006. ISBN: 9781584885085.

[114] Daisuke Suzuki, Minoru Saeki, and Tetsuya Ichikawa. “DPA leakage models for CMOS logic circuits”. In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2005, pp. 366–382.

[115] Harald Uhlig. “On Singular Wishart and Singular Multivariate Beta Distributions”. In: *Ann. Statist.* 22.1 (Mar. 1994), pp. 395–405.

[116] Martin Vetterli, Jelena Kovacevic, and Vivek K Goyal. *Foundations of signal processing*. Cambridge University Press, 2014.

[117] Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. “Soft analytical side-channel attacks”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2014, pp. 282–296.

[118] Martin J Wainwright, Michael I Jordan, et al. “Graphical models, exponential families, and variational inference”. In: *Foundations and Trends in Machine Learning* 1.1–2 (2008), pp. 1–305.

[119] Lyndon White et al. *JuliaDiff/ChainRules.jl: v0.8.2*. Version v0.8.2. June 2021.

[120] Peter Wright and Paul Greengrass. *The Spycatcher*. Heinemann, 1987.

[121] Guorong Xuan, Wei Zhang, and Peiqi Chai. “EM algorithms of Gaussian mixture model and hidden Markov model”. In: *Proceedings 2001 International Conference on Image Processing*. Vol. 1. 2001, 145–148 vol.1.

[122] Guang Yang, Huizhong Li, Jingdian Ming, and Yongbin Zhou. “Convolutional Neural Network Based Side-Channel Attacks in Time-Frequency Representations”. In: *Smart Card Research and Advanced Applications*. Ed. by Begül Bilgin and Jean-Bernard Fischer. Springer International Publishing, 2019, pp. 1–17.

[123] Meyer Yves. *Ondelettes et opérateurs . I, Ondelettes*. Actualités mathématiques. Hermann, 1989.

[124] Xinping Zhou, Carolyn Whitnall, Elisabeth Oswald, Degang Sun, and Zhu Wang. “A Novel Use of Kernel Discriminant Analysis as a Higher-Order Side-Channel Distinguisher”. In: *Smart Card Research and Advanced Applications*. Ed. by Thomas Eisenbarth and Yannick Teglia. Cham: Springer International Publishing, 2018, pp. 70–87.

[125] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. 1st. Chapman & Hall/CRC, 2012.

Appendix

A.1 ML for the estimation of Generalized Morse Wavelets

We recall the loss and the optimisation problem of Sec. 4.3. Given a set of data $D = \{y_1, \dots, y_L\}$ of size L, we aim at optimising the following likelihood loss:

$$\mathcal{L}_{\text{MLE}}(\mu, \Xi, \Lambda, \Sigma) = -\log \prod_{i=1}^{L} p(y_i|\mu, \Xi, \Lambda, \Sigma)\,. \tag{39}$$

We search for $\mu_*, \Xi_*, \Lambda_*, \Sigma_*$ minimizing $\mathcal{L}_{\text{MLE}}$:

$$(\mu_*, \Xi_*, \Lambda_*, \Sigma_*) = \underset{\mu,\Xi,\Lambda,\Sigma}{\arg\min}\, \mathcal{L}_{\text{MLE}}(\mu, \Xi, \Lambda, \Sigma) \tag{40}$$

The differential form is given by:

$$\begin{aligned}\frac{1}{L} d\mathcal{L}_{\text{MLE}} = &\operatorname{tr}\left(h[\Sigma_y, S] d\Sigma_y\right) \\ &+ 2\Re\left((\mu - \mu_e)^\dagger {\Sigma_y}^{-1} d\mu\right)\,.\end{aligned} \tag{41}$$

with

$$h[A, B] = A^{-1}(A - B)A^{-1}\ ,\ S = \frac{1}{L}\sum_{j=1}^{L}(y_j - \mu)(y_j - \mu)^\dagger\,.$$

Starting from:

$$\begin{aligned}\frac{1}{L} d\mathcal{L}_{\text{MLE}} = &\operatorname{tr}\left(h[\Sigma_y, S] d\Sigma_y\right) \\ &+ (\mu - \mu_e)^\dagger \Sigma_y^{-1} d\mu + (\mu - \mu_e)^T \overline{\Sigma_y^{-1} d\mu}\end{aligned} \tag{42}$$

For μ:

$$\nabla_\mu \mathcal{L}_{\text{MLE}} = \overline{(\nabla_{\bar{\mu}} \mathcal{L}_{\text{MLE}})} \tag{43}$$

$$= (\mu - \mu_e)^\dagger \Sigma_y^{-1} \tag{44}$$

Now developing $\mathrm{tr}\,(h[\Sigma_y, S]d\Sigma_y)$ using (4.25) with $\Sigma = CC^\dagger$ and $\Psi^\lhd = C^\dagger\Psi$:

$$\begin{aligned}
\mathrm{tr}\,(h[\Sigma_y, S]d\Sigma_y) &= \mathrm{tr}\,(h[\Sigma_y, S]d\Lambda) + \mathrm{tr}\left(h[\Sigma_y, S]\Psi_\Xi^\dagger dC\Psi_\Xi^\lhd\right) \\
&\quad + \mathrm{tr}\left(h[\Sigma_y, S]\Psi_\Xi^{\lhd\dagger} dC^\dagger\Psi_\Xi\right) + \mathrm{tr}\left(h[\Sigma_y, S]\Psi_\Xi^{\lhd\dagger} C^\dagger d\Psi_\Xi\right) \qquad (45)\\
&\quad + \mathrm{tr}\left(h[\Sigma_y, S]d\Psi_\Xi^\dagger C\Psi_\Xi^\lhd\right) \\
&= \mathrm{tr}\,(h[\Sigma_y, S]d\Lambda) + \mathrm{tr}\left((\Psi_\Xi h[\Sigma_y, S]\Psi_\Xi^{\lhd\dagger})^\dagger dC\right) \\
&\quad + \mathrm{tr}\left((\Psi_\Xi h[\Sigma_y, S]\Psi_\Xi^{\lhd\dagger})^T d\overline{C}\right) + \mathrm{tr}\left((C\Psi_\Xi^\lhd h[\Sigma_y, S])^\dagger d\Psi_\Xi\right) \qquad (46)\\
&\quad + \mathrm{tr}\left((C\Psi_\Xi^\lhd h[\Sigma_y, S])^T d\overline{\Psi_\Xi}\right) \\
&= \mathrm{vec}\,(h[\Sigma_y, S])^\dagger \mathrm{vec}\,(d\Lambda) + \mathrm{vec}\left(\Psi_\Xi h[\Sigma_y, S]\Psi_\Xi^{\lhd\dagger}\right)^\dagger \mathrm{vec}\,(dC) \\
&\quad + \mathrm{vec}\left(\Psi_\Xi h[\Sigma_y, S]\Psi_\Xi^{\lhd\dagger}\right)^T \mathrm{vec}\left(d\overline{C}\right) \\
&\quad + \mathrm{vec}\,(C\Psi_\Xi^\lhd h[\Sigma_y, S])^\dagger \mathrm{vec}\,(d\Psi_\Xi) \qquad (47)\\
&\quad + \mathrm{vec}\,(C\Psi_\Xi^\lhd h[\Sigma_y, S])^T \mathrm{vec}\left(d\overline{\Psi_\Xi}\right)
\end{aligned}$$

We recognize:

$$\begin{aligned}
\nabla_{\Psi_\Xi}\mathcal{L}_{\mathrm{MLE}} &= \overline{(\nabla_{\overline{\Psi_\Xi}}\mathcal{L}_{\mathrm{MLE}})} \\
&= \mathrm{vec}\,((C\Psi_\Xi^\lhd h[\Sigma_y, S]))^\dagger \qquad (48)\\
\nabla_C\mathcal{L}_{\mathrm{MLE}} &= \overline{(\nabla_{\overline{C}}\mathcal{L}_{\mathrm{MLE}})} \\
&= \mathrm{vec}\left((\Psi_\Xi h[\Sigma_y, S]\Psi_\Xi^{\lhd\dagger})\right)^\dagger \qquad (49)\\
\nabla_\Lambda\mathcal{L}_{\mathrm{MLE}} &= \mathrm{vec}\,(h[\Sigma_y, S])^\dagger \qquad (50)
\end{aligned}$$

The chain rules for $\xi \in \Xi$ give the jacobian:

$$\begin{aligned}
\nabla_\xi\mathcal{L}_{\mathrm{MLE}} &= \nabla_{\Psi_\Xi}\mathcal{L}_{\mathrm{MLE}}\,\nabla_{\widehat{\psi}_\xi}\Psi_\Xi\,\nabla_\xi\widehat{\psi}_\xi \\
&\quad + \nabla_{\overline{\Psi_\Xi}}\mathcal{L}_{\mathrm{MLE}}\,\nabla_{\widehat{\psi}_\xi}\overline{\Psi_\Xi}\,\nabla_\xi\overline{\widehat{\psi}_\xi} \qquad (51)\\
&= 2\Re\left(\nabla_{\Psi_\Xi}\mathcal{L}_{\mathrm{MLE}}\,\nabla_{\widehat{\psi}_\xi}\Psi_\Xi\,\nabla_\xi\widehat{\psi}_\xi\right) \qquad (52)
\end{aligned}$$

Where we used that $\nabla_{\Psi_\Xi}\mathcal{L}_{\mathrm{MLE}} = \overline{(\nabla_{\overline{\Psi_\Xi}}\mathcal{L}_{\mathrm{MLE}})}$,$\nabla_{\widehat{\psi}_\xi}\Psi_\Xi = \nabla_{\overline{\widehat{\psi}_\xi}}\overline{\Psi_\Xi}$ and $\nabla_\xi\widehat{\psi}_\xi = \overline{(\nabla_\xi\overline{\widehat{\psi}_\xi})}$

Recalling the expression of ψ_ξ with $\xi = (a, u, \beta, \gamma)$

$$\widehat{\psi}_{a,u;\beta,\gamma} = \sqrt{a}c_{\beta,\gamma}(aw)^\beta e^{-(aw)^\gamma}e^{-iwu}\,. \qquad (53)$$

with $c_{\beta,\gamma}^2 = \pi\gamma\ 2^r/\Gamma(r)$ and $r = (2\beta+1)/\gamma$. The jacobian of $\nabla_\xi\widehat{\psi}_\xi$ writes

$$\nabla_\xi\widehat{\psi}_\xi = \left[\frac{\partial\widehat{\psi}_\xi}{\partial a} \quad \frac{\partial\widehat{\psi}_\xi}{\partial u} \quad \frac{\partial\widehat{\psi}_\xi}{\partial\beta} \quad \frac{\partial\widehat{\psi}_\xi}{\partial\gamma}\right], \qquad (54)$$

with:

$$\frac{\partial\widehat{\psi}_\xi}{\partial a}(w) = \frac{\gamma}{2}\left(\frac{r}{2} - e^{\gamma\log(aw)}\right)\widehat{\psi}_\xi(w) \tag{55}$$

$$\frac{\partial\widehat{\psi}_\xi}{\partial u}(w) = iw\widehat{\psi}_\xi(w) \tag{56}$$

$$\frac{\partial\widehat{\psi}_\xi}{\partial \beta}(w) = \left(\frac{1}{\gamma}(\log(2) - \mathrm{dig}(r)) + \log(aw)\right)\widehat{\psi}_\xi(w) \tag{57}$$

$$\begin{aligned}\frac{\partial\widehat{\psi}_\xi}{\partial \gamma}(w) = &\left(\frac{1}{2\gamma}(1 - r(\log(2) - \mathrm{dig}(r))\right.\\ &\left. - \log(aw)(aw)^\gamma\right)\widehat{\psi}_\xi(w)\end{aligned} \tag{58}$$

where $\mathrm{dig}(x) = \frac{\partial \log\Gamma(z)}{\partial z}|_{z=x}$ is the digamma function.

A.2 Poisson point process

For a Poisson point process with rate $\lambda \in \mathbb{R}^+$, the times of occurrence $t_i, i \in \mathbb{N}$ follow the relation

$$\begin{gathered} t_{i+1} - t_i = \tau_i \\ \text{with } \tau_i \sim \mathrm{Exp}(\lambda)\end{gathered} \tag{59}$$

The random variables are noted T_i for the times of occurrence and $\Delta T_i = T_i - T_{i-1}$ for the delays.

Let $N_t \geq k$ be the event *at least k instructions are executed before t*, for $k \geq 1$ with $t_0 = 0$ we have:

$$p(N_T \geq k) = p(T_k < t) \tag{60}$$

$$= p(\sum_{i=1}^{k} \Delta T_i < t) \tag{61}$$

$$= \int_0^t p(\sum_{i=1}^{k} \Delta T_i = u)du \tag{62}$$

For $p(\sum_{i=1}^{k} \Delta T_i = t), \forall t \in \mathbb{R}^+$, we get:

$$p(\sum_{i=1}^{k} \Delta T_i = u) = \int_{\sum_{i=1}^{k} \Delta T_i = u} \prod_{i=1}^{k} p(\tau_i) d\tau_1 \dots d\tau_k \tag{63}$$

$$= \int_{\tau_1=0}^{t} p(\tau_1) \underbrace{\int_{\sum_{i=2}^{k} \Delta T_i = t-\tau_1} \prod_{i=2}^{k} p(\tau_i) d\tau_2 \dots d\tau_k}_{p(\sum_{i=2}^{k} \Delta T_i = t-\tau_1)} d\tau_1 \tag{64}$$

$$= \int_{0}^{t} p(\tau_1) p(\sum_{i=2}^{k} \Delta T_i = t - \tau_1) d\tau_1 \tag{65}$$

Since $p(\tau_1 = t) = p(\sum_{i=2}^{k} \Delta T_i = t) = 0$ for $t < 0$ we get

$$= \int_{-\infty}^{+\infty} p(\tau_1) p(\sum_{i=2}^{k} \Delta T_i = t - \tau_1) d\tau_1 \tag{66}$$

$$= \left(p_{\Delta T_1} * p_{\sum_{i=2}^{k} \Delta T_i} \right)(t) \tag{67}$$

$$= \underbrace{p_{\Delta T_1} * \dots * p_{\Delta T_k}}_{k}(t) \tag{68}$$

$$= \mathcal{F}^{-1} \left(\prod_{i=1}^{k} \widehat{p_{\Delta T_i}} \right)(t) \tag{69}$$

$$= \mathcal{F}^{-1} \left(\widehat{p_{\Delta T}}^{k} \right)(t) \tag{70}$$

with $p_{\Delta T}$ the general probability distribution for the delay, $\mathcal{F}^{-1}$ the inverse Fourier Transform and $\widehat{p_{\Delta T}}$ the Fourier Transform of $p_{\Delta T}$.

In our case, the delay follows an exponential distribution, $p_{\Delta T}(\tau) = \mathbf{1}_{\tau>0} \lambda e^{-\lambda \tau}$, we thus get the result:

$$\widehat{p(\tau)}(w) = \int_{-\infty}^{+\infty} \mathbf{1}_{\tau>0} \lambda e^{-\lambda \tau} e^{-i\omega\tau} d\tau = \frac{\lambda}{i\omega + \lambda}, \, w \in \mathbb{R} \tag{71}$$

Now, with

$$I_k(t) = \mathcal{F}^{-1} \left(\widehat{p_{\Delta T}}^{k} \right)(t) \tag{72}$$

$$I_1(t) = \mathcal{F}^{-1} \left(\widehat{p_{\Delta T}} \right)(t) = \mathbf{1}_{t>0} \lambda e^{-\lambda t} \tag{73}$$

We get the recurrence relation:

$$I_k(t) = \frac{\lambda t}{(k-1)} I_{k-1}(t) \,, \tag{74}$$

giving

$$I_k(t) = \mathbf{1}_{t>0} \frac{\lambda^k}{(k-1)!} t^{k-1} e^{-\lambda t}\,, \tag{75}$$

and thus

$$p(\sum_{i=1}^{k} \Delta T_i = t) = I_k(t) = \mathbf{1}_{t>0} \frac{\lambda^k}{(k-1)!} t^{k-1} e^{-\lambda t}\,. \tag{76}$$

We recognize a Gamma distribution with shape and rate parameters k and λ. Using (76) in (62), the probability distribution of the event $N_t \geq k$ is given by:

$$\begin{aligned} p(N_t \geq k) &= \int_0^t p(\sum_{i=1}^{k} \Delta T_i = t) d\tau_1 \dots d\tau_k && (77)\\ &= \int_0^t \mathbf{1}_{u>0} \frac{\lambda^k}{(k-1)!} u^{k-1} e^{-\lambda u} du && (78)\\ &= \frac{1}{\Gamma(k)} \gamma(k, \lambda t) && (79) \end{aligned}$$

where $\gamma(a,t) = \int_0^t u^{a-1} e^{-u} du$, $a > 0$ is the lower incomplete gamma function, converging to the gamma function $\Gamma(k) = \int_0^{+\infty} u^{a-1} e^{-u} du$ when $t \to \infty$.

We retrieve the intuition that for $t \to \infty$ it is almost certain that at least k operations will occur

$$\lim_{t\to\infty} p(N_t \geq k) = \frac{1}{\Gamma(k)} \lim_{t\to\infty} \gamma(k, \lambda t) = \Gamma(k)/\Gamma(k) = 1\,.$$

Now, we can derive the probability of the event *exactly k operations are executed before t* noted $N_t = k$, using (79) we have:

$$\begin{aligned} p(N_t = k) &= p(N_t \geq k) - p(N_t \geq k+1) && (80)\\ &= \int_0^{\lambda t} \frac{1}{\Gamma(k)} u^{k-1} e^{-u} - \frac{1}{\Gamma(k+1)} u^k e^{-u} du && (81)\\ &= \int_0^{\lambda t} \frac{d}{du} \left(\frac{1}{\Gamma(k+1)} u^k e^{-u} \right) du && (82)\\ &= \frac{1}{\Gamma(k+1)} (\lambda t)^k e^{-\lambda t} && (83) \end{aligned}$$

we recognize a Poisson distribution with rate λt.

By introducing the survival distribution S_X of a continuous univariate random variable X defined by

$$S_X(t) = \int_t^{+\infty} p_X(u) du\,,$$

, we also remark that $p(N_t = k)$ can be written as a convolution

$$p(N_t = k) = p(\sum_{i=1}^{k} \Delta T_i < t, \Delta T_{k+1} > t - \sum_{i=1}^{k} \Delta T_i) \tag{84}$$

$$= \int_0^t p(\sum_{i=1}^{k} \Delta T_i = u) \int_{t-u}^{+\infty} p(\Delta T_{k+1} = x) dx du \tag{85}$$

$$= p_{\sum_{i=1}^{k} \Delta T_i} * S_{\Delta T_{k+1}}(t) \tag{86}$$

$$= p_{\Delta T_1} * \ldots * p_{\Delta T_k} * S_{\Delta T_{k+1}}(t) \tag{87}$$

and gives the same result as (83).