Sorbonne Université

Ecole Doctorale n° 130
Informatique, Télécommunications, Électronique de Paris

Laboratoire d'Informatique de Paris 6

Résumé de thèse en Français

Extraction de caractéristiques pour les attaques par canaux auxiliaires

Feature Extraction for Side-Channel Attacks

Par Eleonora Cagli
Thèse de Doctorat en Informatique

Dirigée par Emmanuel PROUFF
Encadrée par Cécile DUMAS

Table des matières

1 Contexte **1**

1.1 Le CESTI 1

1.2 Les attaques par canaux auxiliaires 1

2 Objectifs et contributions **1**

2.1 L'avant-propos de cette thèse : la recherche des points d'intérêt 2

2.2 Approche par réduction de dimension 2

2.3 Vers l'apprentissage profond 3

3 Résultats principaux **3**

3.1 Notations 3

3.2 Techniques linéaires de réduction de dimension 4

3.2.1 Analyse aux composantes principales, classique et le profilée 4

3.2.2 L'enjeu du choix des composantes 4

3.2.3 La méthode par variance expliquée locale 5

3.2.4 LDA et le problème de l'échantillon de petite taille 6

3.2.5 Résultats et conclusions 6

3.3 Analyse discriminante par noyau 7

3.3.1 Analyse et résultats 9

3.4 Réseaux Neuronaux Convolutifs 9

3.4.1 Architecture et apprentissage 10

3.4.2 Augmentation de données 11

3.4.3 Résultats expérimentaux 12

4 Conclusions et Perspectives **14**

1 Contexte

1.1 Le CESTI

Les présents travaux de doctorat ont été réalisés au sein du laboratoire CESTI (Centre d'Évaluation de la Sécurité des Systèmes d'Information) du CEA de Grenoble. La mission d'un CESTI est d'évaluer les aspects sécuritaires des produits qui nécessitent l'obtention d'un certificat pour pouvoir être commercialisés sur certains marchés sensibles. Les cartes à puce sont un exemple notable de tels types de dispositifs. Dans le schéma de certification français, c'est l'ANSSI (Agence National de la Sécurité des Systèmes d'Information) qui délivre le certificat, après consultation d'un rapport issu d'un des laboratoires CESTI agréés.

Un dispositif sécurisé permet, dans la grande majorité des cas, d'exécuter des algorithmes cryptographiques, pour offrir des garanties de confidentialité, authenticité, non-répudiation et intégrité des données pour les protocoles d'interface avec ce même dispositif. Quand un algorithme cryptographique est implémenté sur un support matériel, il devient potentiellement vulnérable à des attaques autres que celles considérées en cryptanalyse classique. En effet, en plus de la faiblesse mathématique théorique de l'algorithme, il existe des faiblesses matérielles liées à l'implémentation. Ces attaques matérielles sont à prendre en compte dans l'évaluation sécuritaire d'un produit sécurisé. Notamment, une partie du processus d'évaluation consiste à mener des attaques par canaux auxiliaires (ou *Side-Channel Attacks* en anglais, d'où l'acronyme SCA), qui font l'objet de cette thèse, et qui exploitent des fuites d'information issues de *canaux auxiliaires*, c'est-à-dire autres que les interfaces I/O du composant.

1.2 Les attaques par canaux auxiliaires

Introduites en 1996 par Paul Kocher [11], les attaques par canaux auxiliaires sont basées sur l'observation des variations de certaines quantités physiques du composant, comme la consommation de puissance ou le rayonnement électromagnétique, pendant l'exécution des algorithmes cryptographiques. En effet, en observant ces comportements physiques involontaires, qui sont mesurés sous forme de signaux, des déductions sur les variables internes de l'algorithme peuvent être faites. L'attaquant choisit ensuite, selon l'algorithme attaqué, les variables internes, appelées *variables sensibles*, qui seront suffisantes pour inférer la clé secrète.

2 Objectifs et contributions

Dans le contexte d'évaluation d'un dispositif sécurisé, les évaluateurs peuvent avoir accès à un ou plusieurs exemplaires du dispositif *ouverts*, ou *à secrets connus*. Ces dispositifs donnent droit à l'évaluateur de choisir ou connaître la clé secrète, de fixer d'autres variables, de désactiver des contremesures, ou de charger du logiciel. Cette possibilité est exploitée pour lancer des exécutions dans lesquelles l'attaquant aurait la connaissance complète du flux d'exécution, y compris les opérations, les variables internes manipulées, les accès aux registres, les aléas tirés en interne, etc. De cette manière il est capable de comprendre et caractériser les relations entre le comportement interne du composant et les observations physiques, avant de lancer l'attaque sur un autre dispositif, fermé, à clé inconnue. Quand une phase de caractérisation est disponible, on parle d'attaques par profilage, qui ont un rôle très important dans l'évaluation d'un dispositif, permettant de tester celui-ci dans le scénario le plus favorable pour l'attaquant. Cette thèse se concentre principalement sur cette typologie d'attaques. En effet, nous traitons les problèmes qu'un évaluateur rencontre quand, dans un scénario si favorable, il veut exploiter de façon optimale la phase de caractérisation, afin d'extraire ensuite un maximum d'information des signaux acquis dans la phase propre d'attaque. La stratégie plus répandue pour mener une attaque par profilage est l'attaque *template* [7] : dans cette attaque, la phase de caractérisation consiste à construire un profil statistique des fuites, pour chaque valeur (ou classe) de la variable sensible, à l'aide des traces de profilage, acquises à secrets connus et étiquetées avec la valeur correcte de la variable sensible. Ces profils sont des estimations de la densité de probabilité d'observer un signal (vu comme un vecteur aléatoire, assumé suivre une loi gaussienne multivariée) sachant la valeur de la donnée ciblée (sa classe). Lors de la phase d'attaque, les traces acquises à clé inconnue sont comparées aux profils, et un calcul de maximum de probabilité *a-posteriori* amène au choix du candidat parmi les hypothèses de clé. Comme détaillé dans la suite, un des enjeux de cette

approche est la sélection des dits *points d'intérêt* (*Points of Interest* en anglais, ou PoIs), problème strictement relié au problème plus général de la réduction de dimension.

2.1 L'avant-propos de cette thèse : la recherche des points d'intérêt

L'acquisition des signaux issus des canaux auxiliaires se fait habituellement à l'aide d'un oscilloscope numérique, qui effectue un échantillonnage des signaux analogiques et les transforme en séquences numériques discrètes. Ces séquences sont souvent appelés *traces*, et leurs composantes sont les *caractéristiques* temporelles, ou points temporels, du signal. Pour garantir une analyse poussée du dispositif, la fréquence d'échantillonnage peut être très élevée, ce qui provoque l'acquisition de traces de grande dimension. Cependant, en général, il semble qu'un nombre limité de points temporels soit suffisant pour mener une attaque. Ce sont les PoIs, c'est-à-dire les points qui dépendent statistiquement de la variable sensible ciblée par l'attaque. En littérature l'utilisation de certains tests d'hypothèse statistique est déployée pour effectuer une sélection des PoI comme phase préliminaire d'une attaque. Cette sélection permettrait de réduire la complexité de l'attaque, en terme de temps et de mémoire. L'objectif initial de cette thèse était de proposer de nouvelles méthodes pour chercher et caractériser les PoIs, afin d'améliorer et possiblement optimiser ce pré-traitement des traces par sélection de points.

2.2 Approche par réduction de dimension

Suite à l'utilisation de statistiques univariées pour identifier les PoIs, un axe de recherche différent s'est développé dans le contexte des SCAs, issu du domaine de l'apprentissage automatique (ou *Machine Learning*, ML). Des techniques plus générales ont été proposées pour réduire la dimension des données, passant d'une approche par sélection de caractéristiques à une approche par *extraction de caractéristiques*. Vers 2014, les méthodes linéaires d'extraction de caractéristiques ont attiré l'attention des chercheurs en proposant l'application de techniques telles que l'*Analyse aux Composantes Principales* (PCA), l'*Analyse Discriminante Linéaire* (LDA) ou les *Projection Pursuits* (PP). Ces méthodes exploitent des combinaisons linéaires avantageuses des points temporels, pour définir de nouvelles caractéristiques amenant à des attaques plus efficaces. La première contribution de cette thèse fait partie de cet axe de recherche : nous avons abordé deux problématiques concernant l'application des PCA et LDA dans le contexte SCA : le choix des composantes et le problème de l'échantillon de petite taille. Les résultats de cette étude, publiés en 2015 à CARDIS [3], sont résumés en section 3.2 et font l'objet du chapitre 4 de la thèse.

Aujourd'hui, tout dispositif sécurisé est équipé de contremesures spécifiques contre les SCAs. Un type de contremesure très efficace repose sur le *masquage*. Quand un masquage est implémenté correctement, toute variable interne du calcul originaire qui est sensible, est divisée en plusieurs morceaux, dont la majorité est tirée au sort pendant l'exécution (les *masques*). Ainsi tout sous-ensemble de morceaux est statistiquement indépendant de la variable sensible elle-même. Le calcul cryptographique est mené en accédant uniquement aux morceaux, et non pas à la variable sensible. Ceci oblige l'attaquant à analyser des distributions de probabilité conjointes des caractéristiques du signal, en étudiant conjointement son comportement aux instants temporels où chacun des morceaux est manipulé. Autrement dit, les statistiques univariées qui sont exploitables pour identifier les PoIs en absence de masquage deviennent inefficaces si un masquage est présent, car tout point temporel du signal est par lui-même indépendant de la variable sensible. En outre, les distributions jointes du signal doivent être analysées aux ordres statistiques supérieurs pour retrouver une dépendance statistique avec les données sensibles. Ceci implique que les méthodes linéaires d'extraction de caractéristiques sont aussi inefficaces dans ce contexte. Ainsi, la sélection ou l'extraction de caractéristiques dans des traces protégées par masquage présente des difficultés non-négligeables. Cette complexité est mitigée quand l'attaquant peut effectuer une phase de caractérisation pendant laquelle il peut accéder aux masques. En pratique, ceci n'est pas toujours possible. Dans cette thèse nous abordons ce sujet dans le cas où cette possibilité est nulle, en proposant l'exploitation de la technique de l'*analyse discriminante par noyau* (*Kernel Discriminant Analysis*, KDA). C'est une extension de la LDA qui permet d'extraire des caractéristiques de façon non-linéaire. Les résultats obtenus dans ce contexte ont été publiés à CARDIS 2016 [4] et sont résumés en section 3.3. Ils font l'objet du chapitre 5 de la thèse.

2.3 Vers l'apprentissage profond

En observant le chemin que nous avons suivi pendant les travaux de thèse, on remarque que nous sommes partis du problème de l'identification des PoIs d'un signal, ce qui est classiquement résolu par des outils statistiques classiques, et qu'ensuite nous avons élargi à la fois les objectifs et les méthodologies. En effet, on sait que la qualité de l'extraction d'information a une grande influence sur la réussite d'une attaque. Or pour extraire de l'information il faut approximer des distributions de probabilité afin de distinguer différentes valeurs secrètes. Les premières attaques par canaux auxiliaires proposées en littérature opéraient point par point ; elles nécessitaient donc d'analyser les distributions de données en seulement quelques instants temporels pris séparément. Dans ce contexte la sélection des PoIs jouait un rôle fondamental. Cependant, dès qu'on revient à l'objectif initial d'une attaque, et qu'on se demande comment approximer des distributions dissociables, le fait de rejeter complètement une grande partie des caractéristiques du signal, en n'en sélectionnant que quelques unes, paraît du gaspillage. Combiner toutes les caractéristiques en quelques points paraît une méthode plus appropriée pour obtenir des caractéristiques plus discriminantes. Afin de déterminer les meilleures combinaisons nous avons exploré les outils d'extraction de caractéristiques, que nous avons utilisés comme pré-traitement du signal. Dans un premier temps, nous avons considéré des outils linéaires, ensuite des généralisations non-linéaires pour traiter aussi les implémentations protégées par masquage.

Conscients du fait que ces outils sont à mi-chemin entre les statistiques multivariées classiques et le domaine de l'apprentissage automatique, nous avons commencé à explorer ce domaine, qui est aujourd'hui en grand développement. Le grand intérêt pour l'apprentissage automatique est justifié par sa capacité à capter et analyser des données de grande dimension dans une large variété de champs applicatifs, y compris les attaques par canaux auxiliaires. Pour cela, des modèles de plus en plus complexes sont mis en œuvre, trop complexes pour être traités dans un cadre de statistiques formelles. L'apprentissage automatique accepte des non-optimalités intrinsèques mais montre aujourd'hui d'excellents résultats.

L'étude des outils d'apprentissage automatique nous a menés à revenir aux objectifs fondamentaux d'une attaque : plutôt qu'optimiser des pré-traitements de données afin d'obtenir des caractéristiques montrant des distributions facilement distinguables, nous pouvons chercher des modèles pour approximer directement ces distributions à partir des données brutes. Cette approche relève d'une branche de l'apprentissage automatique, appelée *apprentissage profond*. En apprentissage profond, la phase de caractérisation des données est effectuée en un seul processus, qui intègre éventuellement les pré-traitements nécessaires. Ce bénéfice est obtenu grâce aux modèles multi-couches, notamment les *réseaux neuronaux* (*Neural Networks*, NN), sur lesquels nous nous concentrons dans la dernière partie de la thèse. Étant des modèles non-linéaires, les NN peuvent être utilisés pour compenser la contremesure de masquage. De plus, des architectures particulières de NN, les dits réseaux convolutifs (CNN), conçus originalement pour la reconnaissance d'images, s'adaptent aussi à d'autres types de contremesures : celles qui provoquent une désynchronisation des signaux. Nous avons étudié ce contexte, en proposant l'utilisation des CNNs comme solution, munie d'une autre stratégie classique dans le domaine de l'apprentissage automatique, l'*augmentation des données* (DA). Le chapitre 6 de la thèse est dédié à ce sujet. Les résultats obtenus ont été publiés à CHES 2017 [5] et sont résumés en section 3.4.

3 Résultats principaux

3.1 Notations

Dans la thèse, le symbole X désigne une variable aléatoire ($\vec{X}$ pour un vecteur colonne aléatoire) sur un ensemble $\mathcal{X}$, et x (respectivement $\vec{x}$) désigne une réalisation de X (respectivement $\vec{X}$). La i-ème coordonnée d'un vecteur $\vec{x}$ est représentée par $\vec{x}[i]$, et la transposée d'un vecteur $\vec{x}$ par $\vec{x}^\intercal$. Les matrices sont notées en gras majuscules (par exemple $\mathbf{A}$). Les traces acquises des canaux auxiliaires sont interprêtées comme réalisations $\vec{x}_1, \ldots, \vec{x}_N$ d'un vecteur aléatoire réel $\vec{X} \in \mathbb{R}^D$, où D est la longueur du signal. Quand une méthode de réduction de dimension est utilisée comme pré-traitement, celle-ci amène à la définition d'une fonction appelée *extracteur* et notée $\epsilon\colon \mathbb{R}^D \to \mathbb{R}^C$. La variable

sensible manipulée pendant l'acquisition des traces est notée Z. Celle-ci peut avoir différentes formes, mais souvent dans cette thèse elle est définie comme $Z = f(K, E)$, où E désigne une variable publique, par exemple une partie de message en clair, et K une partie d'une clé secrète que l'attaquant souhaite retrouver. Les valeurs acquises par la variable sensible sont vues comme réalisations de la variable aléatoire Z en $\mathcal{Z} = \{s_1, \ldots, s_{|\mathcal{Z}|}\}$. Les éléments de Z sont parfois encodés via le *one-hot-encoding* : à chaque élément s_j on associe un vecteur $\vec{s_j}$ de dimension $|\mathcal{Z}|$, avec toutes les entrées nulles, sauf la j-ème qui est égale à 1 : $s_j \to \vec{s_j} = (0, \ldots, 0, \underbrace{1}_{j}, 0, \ldots, 0)$. Un élément générique de $\mathcal{Z}$ sera noté s, si son indice i n'est pas nécessaire.

3.2 Techniques linéaires de réduction de dimension

Dans cette section sont décrites les études menées autour des méthodes linéaires d'extraction de caractéristiques, en particulier de l'Analyse aux Composantes Principales (PCA) et de l'Analyse Discriminante Linéaire (LDA).

3.2.1 Analyse aux composantes principales, classique et le profilée

L'extracteur linéaire $\epsilon^{\mathrm{PCA}}(\vec{x}) = \mathbf{A}\vec{x}$ se déduit des certains vecteurs propres $\vec{\alpha}_1, \ldots, \vec{\alpha}_C$, appelés *Composantes Principales* (PCs), dont les transposés sont arrangées en tant que lignes dans la matrice de projection **A**. Classiquement la PCA intervient sur des données non étiquetées $\vec{x}_1, \ldots, \vec{x}_N$, dont la moyenne est supposée nulle et rangées en colonnes dans une matrice **M** de dimension $D \times N$, de telle sorte que la matrice de covariance des données soit la suivante :

$$\mathbf{S} = \frac{1}{N}\mathbf{M}\mathbf{M}^{\intercal} \ . \tag{1}$$

Dans ce cas, les vecteur propres $\vec{\alpha}_1, \ldots, \vec{\alpha}_C$ correspondent aux vecteurs propres de la matrice **S** et leurs valeurs propres associées sont notées $\lambda_1, \ldots, \lambda_r$. La PCA est la projection qui maximise la variance globale des caractéristiques extraites. La variance étant liée à la quantité d'information des données, cette transformation est censée réduire la dimension des traces tout en renforçant l'information contenue. Une propriété remarquable de la PCA est que chaque λ_i correspond à la variance empirique des données projetées sur la PC correspondante $\vec{\alpha}_i$.

Dans un scénario d'attaque par profilage, cet outil classique est toutefois largement sous-optimal : il n'exploite aucune phase de caractérisation où l'attaquant est en possession d'un ensemble de données étiquetées $(\vec{x}_i, z_i)_{i=1..N_p}$. Dans la littérature SCA [1, 8, 9, 10, 14] une version *profilée* de la PCA a été introduite. En utilisant les moyennes empiriques par classe :

$$\vec{\mu}_s = \hat{\mathbb{E}}[\vec{X} \mid Z = s] = \frac{1}{N_s} \sum_{i:\ z_i = s} \vec{x}_i \ , \tag{2}$$

la PCA profilée utilise la matrice des *écarts inter-classes* suivante à la place de la matrice de covariance **S** :

$$\mathbf{S_B} = \sum_{s \in \mathcal{Z}} N_s (\vec{\mu}_s - \overline{\vec{x}})(\vec{\mu}_s - \overline{\vec{x}})^{\intercal} \ , \tag{3}$$

où $\overline{\vec{x}}$ est la moyenne empirique de toutes les données confondues. L'extracteur obtenu de cette manière garantit que les centroïdes par classes des données projetées sont les plus éloignés possible.

3.2.2 L'enjeu du choix des composantes

L'introduction de la PCA dans le contexte des SCA a levé les questions suivantes : *combien* de composantes et *lesquelles* sont suffisantes/nécessaires pour réduire la dimension des traces sans perdre de l'information discriminante importante ? Une première réponse a été donnée dans [9], en relation au concept de *variance expliquée* (ou *variance expliquée globale*, EGV) d'une PC $\vec{\alpha}_i$:

$$\mathrm{EGV}(\vec{\alpha}_i) = \frac{\lambda_i}{\sum_{k=1}^{r} \lambda_k} \ . \tag{4}$$

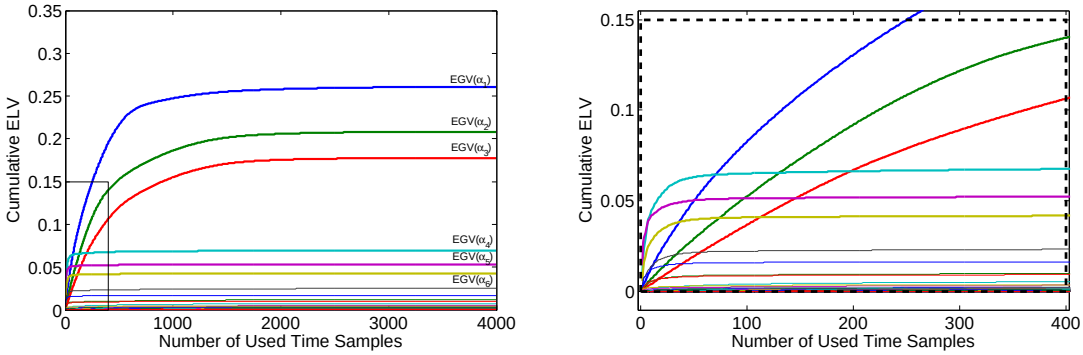

Figure 1 – Tendance cumulative de l' ELV des composantes principales. A droite un zoom de l'image de gauche. Campagne d'acquisition sur 8-bit AVR Atmega328P.

Par définition, la somme de toutes les EGV est égale à 1. Le choix des composantes basé sur l'EGV consiste à fixer un seuil β pour la *variance expliquée cumulative* et à garder C PCs distinctes, où C est l'entier minimum tel que :

$$\text{EGV}(\vec{\alpha}_1) + \text{EGV}(\vec{\alpha}_2) + \cdots + \text{EGV}(\vec{\alpha}_C) \geq \beta \; . \tag{5}$$

Si l'attaquant a une contrainte de dimension finale C, le choix par EGV consiste simplement à garder les C premières composantes, dans l'ordre naturel des valeurs propres associées. Dans le contexte des SCA, ce type de choix ne semble pas être systématiquement le meilleur : des articles déclarent que les plus fortes composantes contiennent la plus grande quantité d'information, alors que d'autres suggèrent d'écarter ces premières composantes [2, 13].

Dans le contexte des implémentations sécurisées, le but des développeurs est de minimiser les fuites, autrement dit, de réduire le nombre de points de fuite, d'où l'hypothèse suivante qui peut être raisonnablement faite :

Hypothèse 1 *L'information compromettante d'une trace acquise de canaux auxiliaires est localisée en peu de points.*

Sous cette hypothèse, les auteurs de [12] ont donné une deuxième réponse au problème du choix des composantes : ils utilisent le *ratio de participation inversé* (IPR) pour évaluer la *localisation* des vecteurs propres. Le IPR est défini ainsi :

$$\text{IPR}(\vec{\alpha}_i) = \sum_{j=1}^{D} \vec{\alpha}_i[j]^4 \; . \tag{6}$$

Les auteurs de [12] suggèrent de sélectionner les PCs en ordre inverse par rapport à leur IPR.

3.2.3 La méthode par variance expliquée locale

Les méthodes de sélection par EGV et IPR sont d'une certaine façon complémentaires : la première se base uniquement sur les valeurs propres associées aux PC et ne considère pas la forme des PC elles-mêmes ; la seconde ne considère pas l'information donnée pas les valeurs propres, mais seulement la distribution des coefficients des PCs. Dans ce panorama nous avons proposé une nouvelle méthode de sélection qui fait un pont entre l'EGV et l'IPR : nous avons introduit la *variance expliquée locale* (ELV) d'une PC $\vec{\alpha}_i$ en un point j, définie par :

$$\text{ELV}(\vec{\alpha}_i, j) = \frac{\lambda_i \vec{\alpha}_i[j]^2}{\sum_{k=1}^{r} \lambda_k} = \text{EGV}(\vec{\alpha}_i)\vec{\alpha}_i[j]^2 \; . \tag{7}$$

Soit $\mathcal{J} = \{j_1^i, j_2^i, \ldots, j_D^i\} \subset \{1, 2, \ldots, D\}$ un ensemble d'indices ordonnés de telle sorte que $\text{ELV}(\vec{\alpha}_i, j_1^i) \geq \text{ELV}(\vec{\alpha}_i, j_2^i) \geq \cdots \geq \text{ELV}(\vec{\alpha}_i, j_D^i)$. On observe que la somme sur tous les $\text{ELV}(\vec{\alpha}_i, j)$, pour tout

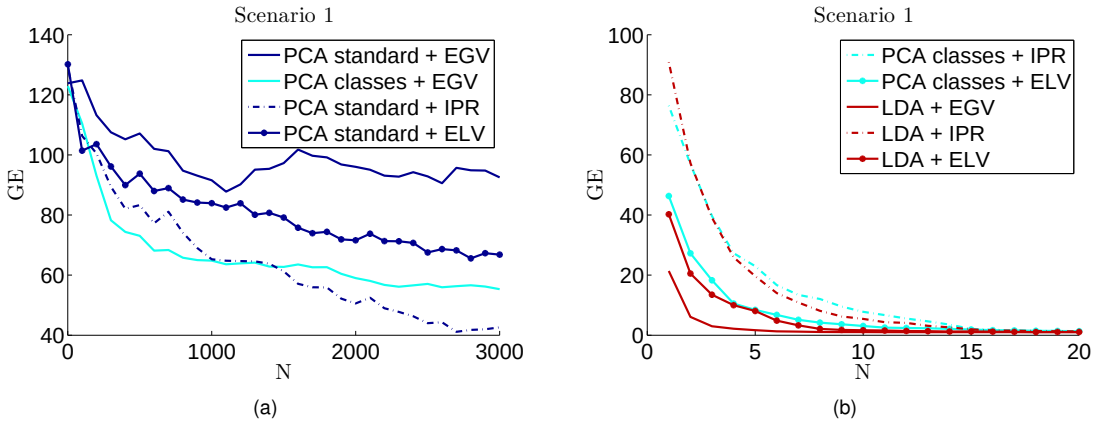

Figure 2 – Guessing Entropy (rang moyen de la bonne hypothèse de clé) en fonction du nombre de traces d'attaque, pour différentes méthodes d'extraction. Les valeurs sont estimées en moyennant le rang sur 100 expériences indépendantes.

$j \in [1, \ldots, D]$, est égale à $\mathrm{EGV}(\vec{\alpha}_i)$. Si on cumule cette somme en suivant l'ordre donné par l'ensemble ordonné $\mathcal{J}$, on obtient une description complète de la tendance suivie par la composante $\vec{\alpha}_i$ pour atteindre son EGV. Comme montré sur la figure 1, où ces ELV cumulatives sont représentées, les trois premières composantes atteignent leur EGV finale très lentement, alors que les $4^{\text{ème}}$, $5^{\text{ème}}$ et $6^{\text{ème}}$ atteignent une large partie de leur EGV très rapidement, c'est-à-dire en sommant les contributions ELV de peu de points. La sélection par ELV, par analogie avec l'EGV, demande de fixer la dimension réduite du signal C, ou un seuil β pour la ELV cumulative. Dans le premier cas, les valeurs maximales d'ELV de chaque PC sont comparées, et les C valeurs les plus élevées sont sélectionnées. Dans le second cas, tous les couples (PC, point temporel) sont ordonnés par ordre décroissant de leur ELV, et sommés jusqu'à atteindre le seuil β. Les PC qui contribuent à la somme sont sélectionnées.

3.2.4 LDA et le problème de l'échantillon de petite taille

L'extracteur ϵ^{LDA} est une réduction de dimension optimale, sous certaines conditions, pour résoudre un problème de classification, c'est-à-dire un problème d'apprentissage où l'on vise à assigner à une donnée (une trace dans notre cas) une étiquette (donnée ici par la valeur de la variable Z manipulée lors de l'acquisition). Il a pour but non seulement d'éloigner les centroïdes par classe, mais aussi de reprocher au mieux les données appartenant à une même classe. La forte analogie entre les SCA et la tâche de classification en apprentissage automatique rend la LDA beaucoup plus adaptée à notre contexte que la PCA, mais elle est plus coûteuse. Comme la PCA, la LDA construit une matrice de projection **A** en juxtaposant des vecteurs propres, appelés composantes linéaires discriminantes (LDC), qui sont ceux de la matrice $\mathbf{S_W}^{-1}\mathbf{S_B}$, où $\mathbf{S_B}$ est définie en (3) et $\mathbf{S_W}$ est la matrice des *écarts intra-classe* :

$$\mathbf{S_W} = \sum_{s \in \mathcal{Z}} \sum_{i\,:\,z_i = s} (\vec{x}_i - \vec{\mu}_s)(\vec{x}_i - \vec{\mu}_s)^\intercal. \tag{8}$$

Le désavantage principal de la LDA est appelé le *problème de l'échantillon de petite taille* et apparaît quand le nombre des traces N est inférieur ou égal à leur taille D. Ceci implique que la matrice $\mathbf{S_W}$ n'est pas inversible. Si la LDA a été introduite relativement tard dans la littérature SCA, la communauté en reconnaissance de formes et apprentissage automatique cherche une solution à ce problème depuis le début des années quatre-vingt-dix. Nous avons exploré les solutions proposées, et les avons testées sur des signaux compromettants.

3.2.5 Résultats et conclusions

Nous avons mené diverses analyses expérimentales pour évaluer la méthode de sélection de composantes par ELV et les techniques d'extension de la LDA en présence d'échantillon de petite taille.

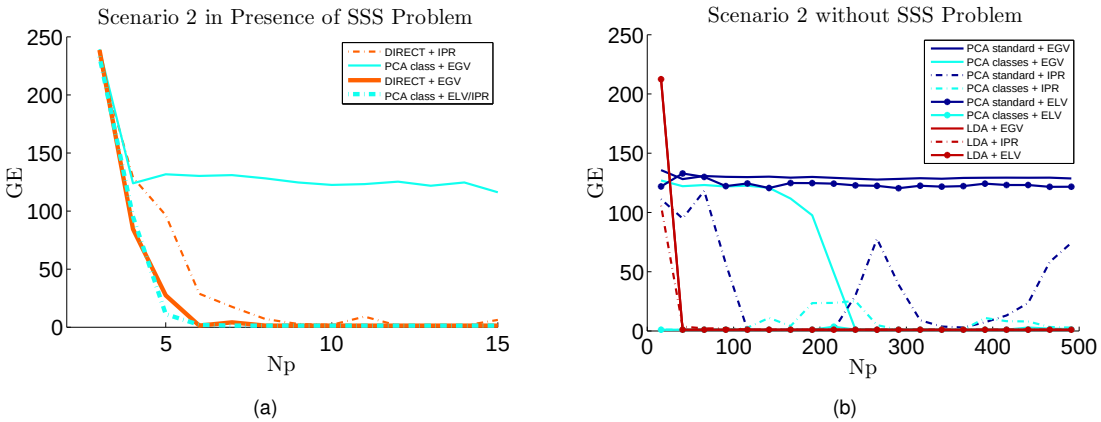

Figure 3 – Guessing Entropy en fonction du nombre de traces de profilage. Figure (a) : la meilleure extension de LDA et la PCA profilée en présence du problème de l'échantillon de petite taille ; Figure (b) : une comparaison en absence de problème de l'échantillon de petite taille.

Dans cette section nous reportons les plus significatifs, en les extrayant du chapitre 4 de la thèse. Nous considérons ici deux scénarios : dans le premier (Scenario 1) l'attaquant a acquis un grand nombre de traces de profilage (N_p) et veut minimiser le nombre de traces d'attaque (N_a) pour obtenir une attaque réussie. En observant la figure 2 nous pouvons tirer trois conclusions :

– nous confirmons que la PCA standard est bien sous-optimale par rapport à la PCA profilée
– nous confirmons que la LDA est plus adaptée que la PCA pour pré-traiter les traces par canaux auxiliaires
– la PCA profilée, munie de la méthode de sélection par ELV (au lieu de la classique EGV), donne des performances de l'attaque significativement meilleures qui se rapprochent de celles obtenues à travers la LDA.

Dans le Scenario 2, l'attaquant veut au contraire minimiser le nombre de traces de profilage nécessaires pour obtenir une bonne méthode de pré-traitement et de profilage. Dans ce scenario, on observe le problème de l'échantillon de petite taille. En observant la figure 3 nous concluons, encore une fois, en présence ou non d'échantillon de petite taille, que la PCA munie de la sélection par ELV a des performances proches de celles de la LDA ou de sa meilleure alternative en présence d'échantillon de petite taille.

3.3 Analyse discriminante par noyau

Quand une contremesure de masquage d'ordre $(d-1)$ est mise en place, la donnée ciblée Z est représentée par d morceaux M_i manipulés à différents instants temporels $t_1, \ldots, t_d$ et l'information sensible réside dans le moment $\mathbb{E}[\vec{X}[t_1]\vec{X}[t_2]\cdots\vec{X}[t_d]]$ d'ordre d, c'est-à-dire que la fonction $f(z) = \mathbb{E}[\vec{X}[t_1]\vec{X}[t_2]\cdots\vec{X}[t_d]|Z = z]$ n'est pas constante. Pour exploiter cette information il a été montré [6] qu'il est nécessaire de considérer une statistique des données qui, vue comme un polynôme multivarié dans les coordonnées de $\vec{X}$, contient le monôme de degré d $\prod_{i=1,\ldots,d}\vec{X}[t_i]$. Comme, en pratique, les points $t_1, \ldots, t_d$ sont inconnus de l'attaquant, une idée naïve pour obtenir un extracteur efficace est de le chercher parmi toutes les combinaisons possibles du produit de d points. Ceci revient à immerger les données à travers une fonction non-linéaire Φ dans un espace de grande dimension $\mathcal{F} = \mathbb{R}^{\binom{D}{d}}$, et d'ici à appliquer les méthodes d'extraction linéaires (LDA et PCA, par exemple). L'espace $\mathcal{F}$ est appelé *espace des caractéristiques*, car c'est l'espace qui contient les caractéristiques informatives des traces. La croissance combinatoire de la taille de $\mathcal{F}$ est un obstacle d'un point de vu calculatoire, car cela demande de manipuler et stocker des traces de taille $\binom{D}{d}$. L'algorithme KDA permet à un attaquant d'utiliser l'extracteur LDA dans $\mathcal{F}$ sans effectuer de calculs dans l'espace $\mathcal{F}$, comme schématisé dans la figure 4.

L'outil central de l'astuce du noyau est la *fonction noyau* $K\colon \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$, qui doit satisfaire la

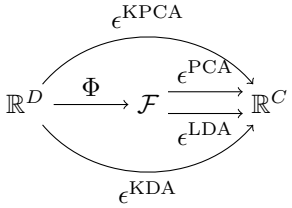

Figure 4 – Appliquer la KDA et la KPCA permet d'éviter des calculs dans l'espace $\mathcal{F}$.

propriété suivante, en rélation avec la fonction Φ :

$$K(\vec{x},\vec{y}) = \Phi(\vec{x}) \cdot \Phi(\vec{y}) \ , \tag{9}$$

pour toutes données $\vec{x}$ et $\vec{y}$, où $\cdot$ est le produit scalaire.

Toute fonction Φ a une fonction noyau associée, donnée par (9), pour un ensemble de données. Les fonctions noyau éligibles comme astuce du noyau sont celles qui sont calculables directement à partir des données brutes $\vec{x}_i$, sans évaluer la fonction Φ.
La fonction noyau *polynomiale de degré* d fait partie de ce type de fonctions. Elle est définie ainsi :

$$K(\vec{x}_i,\vec{x}_i) = (\vec{x}_i \cdot \vec{x}_j)^d \ , \tag{10}$$

et correspond à la fonction Φ qui associe à ses coordonnées tous les monômes de degré d, à quelques constantes près. Elle est donc, à quelques constantes près, la fonction Φ du schéma de la figure 4. L'algorithme pour obtenir ϵ^{KDA} est donné dans le paragraphe suivant.

KDA pour traces masquées à l'ordre d

Étant donné un ensemble de traces de profilage $(\vec{x}i, z_i)_{i=1,\dots,N_t}$ et la fonction noyau $K(\vec{x},\vec{y}) = (\vec{x}\cdot\vec{y})^d$:
1) Construire une matrice **M** (elle agit comme *matrice des écarts inter-classe*) :

$$\mathbf{M} = \sum_{s\in\mathcal{Z}} N_s(\vec{M}_s - \vec{M}_T)(\vec{M}_s - \vec{M}_T)^\intercal \ , \tag{11}$$

où $\vec{M}_s$ et $\vec{M}_T$ sont deux vecteurs colonne de taille N, avec des coordonnées données par :

$$\vec{M}_s[j] = \frac{1}{N_s} \sum_{i:z_i=s} K(\vec{x}_j,\vec{x}_i) \tag{12}$$

$$\vec{M}_T[j] = \frac{1}{N_t} \sum_{i=1}^{N_t} K(\vec{x}_j,\vec{x}_i) \ . \tag{13}$$

2) Construire une matrice **N** (elle agit comme *matrice des écarts intra-classe*) :

$$\mathbf{N} = \sum_{s\in\mathcal{Z}} \mathbf{K}_s(\mathbf{I} - \mathbf{I}_{N_s})\mathbf{K}_s^\intercal \ , \tag{14}$$

où **I** est la matrice identité de taille $N_s \times N_s$, $\mathbf{I}_{N_s}$ est la matrice de taille $N_s \times N_s$ avec tous les éléments égaux à $\frac{1}{N_s}$ et $\mathbf{K}_s$ est la sous-matrice de $\mathbf{K} = (K(\vec{x}_i,\vec{x}_j))_{\substack{i=1,\dots,N_t\\ j=1,\dots,N_t}}$ de taille $N_t \times N_s$ qui contient toutes les colonnes d'indice i tel que $z_i = s$.
3) Régulariser la matrice **N** pour garantir la stabilité calculatoire et gérer le *sur-apprentissage* :

$$\mathbf{N} = \mathbf{N} + \mu\mathbf{I} \tag{15}$$

4) Trouver les valeurs propres non nulles $\lambda_1,\dots,\lambda_Q$ de $\mathbf{N}^{-1}\mathbf{M}$ et les vecteurs propres correspondant $\vec{\nu}_1,\dots,\vec{\nu}_Q$;
5) Finalement, la projection d'une nouvelle trace $\vec{x}$ sur la ℓ-ième composante discriminante d'ordre d se calcule ainsi :

$$\epsilon_\ell^{\mathrm{KDA}}(\vec{x}) = \sum_{i=1}^{N_t} \vec{\nu}_\ell[i] K(\vec{x}_i,\vec{x}) \ . \tag{16}$$

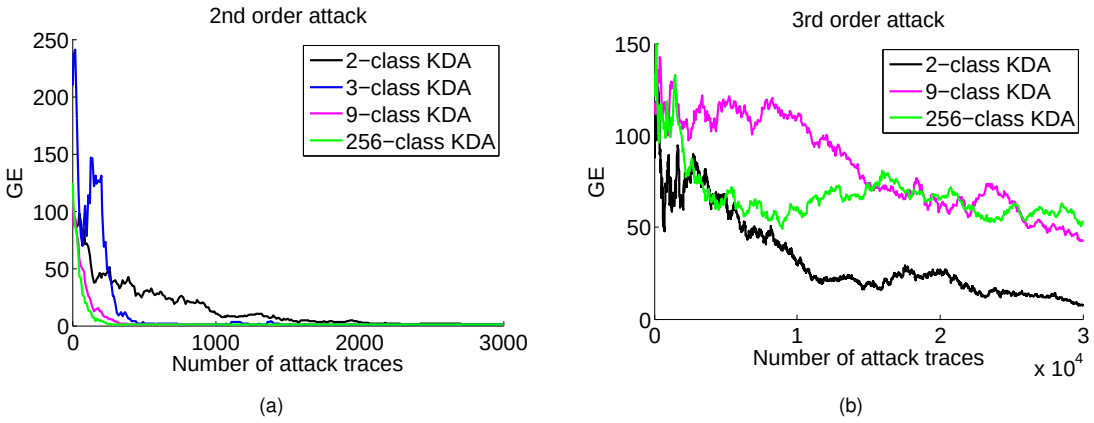

Figure 5 – Comparaison entre les KDA à 2 classes, 3 classes, 9 classes et 256 classes dans un contexte d'attaque du 2nd ordre (a) et du 3ème ordre (b). Pour le 2nd ordre, la KDA arrive à dissocier les données en 256 classes, permettant une caractérisation optimale. Pour le 3ème ordre les données de profilage ne sont pas en nombre suffisant pour réussir une phase de caractérisation avec 256 classes. Diminuer le nombre de classes améliore l'efficacité du pré-traitement et, par conséquence, de l'attaque.

3.3.1 Analyse et résultats

Le premier aspect de la KDA sur lequel nous nous sommes concentrés est l'importance d'une bonne régularisation (obtenue par le choix du paramètre μ dans l'équation (15)) : cette régularisation est la réponse au fait que les méthodes par noyau sont souvent sujettes au sur-apprentissage : ceci signifie, dans le cas de la KDA, que ϵ^{KDA} risque de grouper parfaitement les traces d'apprentissage dans leur classe, mais de ne pas réussir à bien classer les traces d'attaque. La régularisation consiste à rajouter une contrainte à la phase d'apprentissage, dans le but de créer un modèle qui ait le plus possible un même comportement avec les traces d'apprentissage qu'avec les nouvelles données. Nous avons observé que les extracteurs issus d'une bonne régularisation concentraient leurs projections sur des zones des traces très localisées, ce qui est encore le signe (comme pour les extracteurs linéaires) d'une bonne détection implicite des PoIs.

Le deuxième aspect sur lequel nous nous sommes concentrés est le rôle de la forme de la donnée ciblée dans le compromis précision-efficacité. Du fait que la complexité calculatoire de la KDA est $O(N_t^3)$, avec N_t le nombre des traces d'apprentissage il peut être intéressant de diminuer ce nombre N_t pour augmenter l'efficacité du calcul. Cependant, borner N_t réduit la précision de la KDA. Une manière de pallier à cette perte de précision est de réduire le nombre de classes, en choisissant plutôt une variable interne du calcul cryptographique auquelle on applique une fonction non-injective $m(Z)$. Par exemple, les valeurs de Z peuvent être groupées selon leur poids de Hamming : un modèle à 2 classes est donné par ($m(z) = 0$ si $\mathrm{HW}(z) < 4$ et $m(z) = 1$ si $\mathrm{HW}(z) \geq 4$). Une fois qu'un pré-traitement basé sur un certain modèle non-injectif a été réalisé, il parait adéquat de réaliser l'attaque en visant la même donnée sensible. Ayant fixé N_t le nombre de traces d'apprentissage, les résultats de ce type d'attaque sont montrés en figure 5 : si dans un contexte d'attaque du 2nd ordre, on peut observer que la KDA s'entraîne sur un nombre suffisant de données pour réussir une séparation en 256 classes, pour des contextes d'ordre supérieur la conversion vers un problème à 2 classes devient une meilleure stratégie.

3.4 Réseaux Neuronaux Convolutifs

Dans cette dernière partie, nous adoptons une stratégie d'attaque basée sur le paradigme de l'apprentissage profond : à l'aide de réseaux neuronaux à plusieurs couches nous évitons les deux étapes de l'apprentissage, pré-traitement puis caractérisation, mais nous intégrons les pré-traitements de façon implicite dans l'apprentissage. Les réseaux neuronaux, grâce à leurs architectures facilement parallélisables, sont les outils privilégiés aujourd'hui pour résoudre le problème de la classification sur des données de grande dimension, comme les signaux compromettants. Nous nous intéressons en particulier aux réseaux neuronaux convolutifs (*Convolutional Neural Networks*, en anglais, CNN) : étant

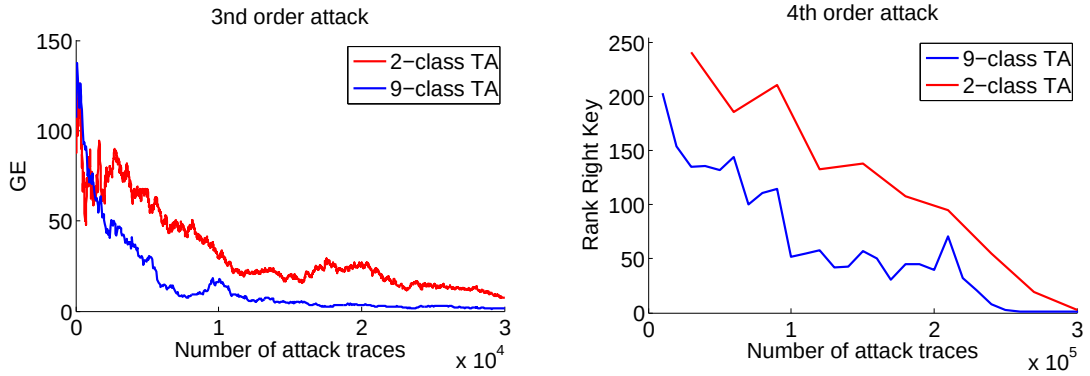

Figure 6 – Gauche : Rang moyen de la bonne clé pour des attaques *template* d'ordre 3, à 2 classes et à 3 classes. Droite : Rang moyen de la bonne clé pour une attaque d'ordre 4 à 2 classes et à 9 classes.

conçus pour être robustes aux déformations géométriques des données, typiques dans le contexte de la reconnaissance de l'image, ils nous ouvrent une nouvelle stratégie pour faire face aux contre-mesures basées sur la désynchronisation dans les acquisitions. Nous nous plaçons donc dans ce contexte, en menant des expériences d'attaque contre ce type de contremesures, en utilisant un réseau CNN, avec des stratégies d'augmentation de données (*Data Augmentation*, DA) bien adaptées au contexte de signaux désynchronisés. Après une brève introduction de ces outils, nous reportons les résultats des expériences en section 3.4.3.

3.4.1 Architecture et apprentissage

Les CNNs appartiennent aux réseaux neuronaux, dont la famille la plus répandue est celle des *Multi-Layer Perceptrons* (MLP). On peut exprimer un MLP destiné à la classification sous la forme :

$$F(\vec{x}) = s \circ \lambda_n \circ \sigma_{n-1} \circ \lambda_{n-1} \circ \cdots \circ \lambda_1(\vec{x}) = \vec{y}\,, \tag{17}$$

où :

– Les fonctions λ_i sont appelées couches entièrement connectées (*Fully-Connected*,FC) et s'expriment comme fonctions affines $\mathbf{A}\vec{x}+\vec{b}$, avec $\mathbf{A} \in \mathbb{R}^{D\times C}$ une matrice de poids et $\vec{b} \in \mathbb{R}^C$ un vecteur de biais, à optimiser à travers l'apprentissage. Ces couches extraient linéairement de l'information des données, de la même manière que les techniques PCA ou LDA.
– Les fonctions σ_i sont appelées couches d'*activation* (ACT) : elles sont généralement non-linéaires, s'appliquent souvent indépendamment à chaque coordonnée de l'entrée, et ne dépendent pas de paramètres entrainables.
– s est la fonction *softmax* : : $s(\vec{x})[i] = \frac{e^{\vec{x}[i]}}{\sum_j e^{\vec{x}[j]}}$. Elle normalise la sortie du réseaux en la rendant interprétable comme une distribution de probabilité $F(\vec{x}) \approx p_{Z \mid \vec{X}=\vec{x}}$.

Les réseaux convolutifs s'obtiennent en rajoutant aux MLPs deux autres typologies de couches :

– Les couches de convolution (CONV) γ, sont des couches linéaires qui partagent leurs poids à travers l'espace. Elles extraient de l'information linéairement à travers des filtres de poids qui agissent localement. Ces filtres glissent le long de leurs entrées et leur action locale est répétée pour chaque entrée. Ceux sont les couches qui se chargent de rendre le réseau robuste aux déformations, lorsque l'information se déplace le long de la trace, et donc robuste à la désynchronisation.
– Les couches de mise en commun (pooling (POOL) en anglais) δ effectuent un sous-échantillonnage des données traitées par le réseau afin de réduire la complexité du calcul due à l'augmentation, niveau par niveau, du nombre de caractéristiques extraites par les couches CONV.

Les paramètres d'un réseau neuronal sont entrainés pendant la phase d'apprentissage. La méthode d'optimisation la plus utilisée est la *descente de gradient stochastique*. Elle est mise en place dans le but de minimiser une fonction de coût qui quantifie l'erreur de classification du réseau sur l'ensemble d'apprentissage. Elle consiste à :

– sélectionner un *mini-lot* de traces d'apprentissage $(\vec{x}_i, z_i)_{i\in I}$ tirées aléatoirement (ici I est un ensemble aléatoire d'indices)
– calculer les sorties du modèle courant pour le mini-lot sélectionné $(\vec{y_i} = F(\vec{x}_i))_{i\in I}$,

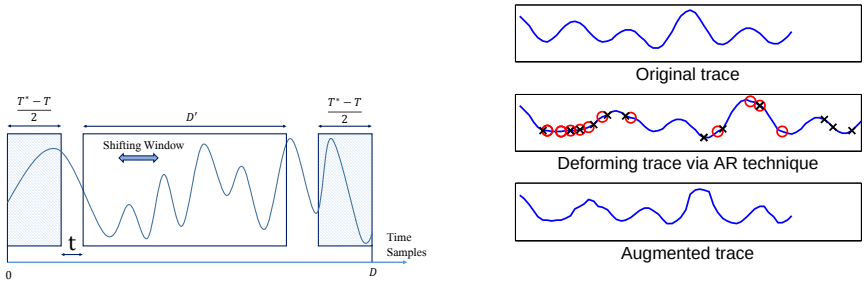

Figure 7 – Gauche : Méthode Shifting pour DA. Droite : Méthode Add-Remove pour DA (les points ajoutés sont marqués par des cercles rouges, les points supprimés par des croix noires.

– évaluer la fonction de coût ; dans le cas de nos expériences il s'agit de l'entropie croisée :

$$\mathcal{L} = -\frac{1}{|I|} \sum_{i \in I} \sum_{t=1}^{|\mathcal{Z}|} \vec{z}_i[t] \log \vec{y}_i[t] \ , \tag{18}$$

– calculer les dérivées partielles de la fonction de coût, par rapport aux paramètres entrainables, à l'aide de la méthode de *rétropropagation du gradient*,
– mettre à jour les paramètres entrainables par soustraction d'un petit multiple du gradient (appelé *taux d'apprentissage*).

Une itération complète sur l'ensemble d'apprentissage est appelée *époque*. La nature des couches, leurs nombres et leurs tailles sont les hyper-paramètres qui définissent l'architecture d'un réseau. Le nombre d'époques, la taille des mini-lots et le taux d'apprentissage sont aussi des hyper-paramètres qui règlent l'apprentissage.

3.4.2 Augmentation de données

Quand un réseau neuronal a une grande capacité, c'est-à-dire qu'il peut représenter des modèles très complexes, et qu'il est entrainé avec un nombre insuffisant de données, il risque de provoquer du sur-apprentissage. Une manière classique de traiter ce problème est l'introduction des méthodes d'augmentation de données. Elles consistent à générer artificiellement des nouvelles données d'apprentissage, en déformant celles réellement acquises. La déformation est obtenue en appliquant des déformations qui préservent la cible de la classification, c'est-à-dire la valeur de la variable sensible manipulée lors de l'acquisition. Nous avons proposé deux techniques d'augmentation de données appelées *Shifting* et *Add-Remove*

Shifting ($\mathrm{SH}_{T^\star}$)

Cette technique simule un effet de délai aléatoire d'amplitude maximale $T^\star$, en sélectionnant une fenêtre glissante sur un signal, comme montré en figure 7. Soit D la taille d'origine du signal. Nous fixons la taille d'entrée de la première couche du réseau neuronal à $D' = D - T^\star$. La technique $\mathrm{SH}_{T^\star}$ consiste alors à (1) tirer un nombre aléatoire uniforme $t \in [0, T^\star]$, et à (2) sélectionner la fenêtre de taille D' à partir du t-ème point de la trace. Pour notre étude, nous souhaitons comparer les techniques SH_T pour différentes valeurs de $T \leq T^\star$, sans changer l'architecture du réseau utilisé (en particulier la taille de l'entrée D'). Notamment, $T \lneq T^\star$ implique que $T^\star - T$ points temporels ne seront certainement jamais sélectionnés. Comme nous supposons que l'information est localisée dans la partie centrale de l'acquisition, nous choisissons de centrer les fenêtres glissantes, en supprimant les premiers et les derniers $\frac{T^\star - T}{2}$ points des traces.

Add-Remove (AR)

Cette technique simule un effet de gigue (jitter en anglais) de l'horloge (figure 7). Nous notons AR_R cette déformation qui consiste en deux étapes :
(1) insérer R points temporels, dont la position est choisie de façon uniformément aléatoire et dont la valeur est obtenue par la moyenne arithmétique des deux valeurs entre lesquelles il s'insère,
(2) supprimer R points temporels, choisis de façon uniformément aléatoire.

Ces deux déformations peuvent être composées : nous notons $\mathrm{SH}_T\mathrm{AR}_R$ l'application de SH_T suivie par celle de AR_R.

3.4.3 Résultats expérimentaux

Pour toutes nos expériences nous avons réglé et fixé une fois pour toutes l'architecture de réseau convolutif utilisé :

$$s \circ [\lambda]^1 \circ [\delta \circ [\sigma \circ \gamma]^1]^4. \tag{19}$$

Attaque par réseau neuronal

La stratégie que nous adoptons pour effectuer une attaque par canaux auxiliaires à l'aide d'un réseau neuronal est la même que pour l'attaque *template*. La différence réside dans le fait que, dans l'attaque template, l'attaquant effectue éventuellement un pré-traitement puis approche les distributions des données à l'aide de modèles génératifs gaussiens, alors que les réseaux neuronaux sont utilisés ici pour construire un modèle discriminatoire, c'est-à-dire qu'ils approchent directement la distribution $F(\vec{x}) \approx p_{Z \mid \vec{X}=\vec{x}}$. Une fois cette approximation faite, l'attaque suit la même stratégie dans les deux approches, propre aux attaques *avancées* de la littérature : l'attaquant acquiert de nouvelles traces d'attaque, à clé inconnue et entrées connues E ; il obtient donc des couples $(\vec{x}_i, e_i)_{i=1,\ldots,N_a}$. Ensuite il effectue des hypothèses de clé $k \in \mathcal{K}$ et associe à chaque hypothèse un score d_k donné par un calcul de probabilité conjointe :

$$d_k = \prod_{i=1}^{N_a} F(\vec{x}_i)[f(k, e_i)] \,. \tag{20}$$

Finalement, le meilleur candidat de clé $\hat{k}$ est celui qui maximise ce score.

Estimation des performances

Le taux de bonne prédiction (accuracy), défini comme le taux de succès sur un certain ensemble de données d'une classification, est la métrique la plus communément utilisée pour le monitorage et l'évaluation d'un réseau neuronal dédié à la classification. Afin de régler les hyperparamètres d'un réseau, il est habituel d'extraire de l'ensemble d'apprentissage une partie des données qui forme un ensemble de *validation*. Ces données ne sont pas utilisées pour la descente de gradient, mais pour le monitorage de la généralisation des résultats du réseau, en particulier pour la prévention du sur-apprentissage. Le *taux de bonne prédiction d'apprentissage*, le *taux de bonne prédiction de validation* et le *taux de bonne prédiction de test* sont respectivement les taux de réussite de la classification sur l'ensemble d'apprentissage, de validation et de test. A la fin de chaque époque les taux de bonne prédiction d'apprentissage et de validation sont calculés. Dans la suite nous utiliserons aussi les quantités suivantes pour résumer les résultats :

– le *taux maximal de bonne prédiction d'apprentissage*, qui est la valeur maximale sur toutes les époques atteinte par le taux de bonne prédiction d'apprentissage,
– le *taux maximal de bonne prédiction de validation*, qui est la valeur maximale sur toutes les époques atteinte par le taux de bonne prédiction validation.

Le taux de bonne prédiction est une métrique parfaitement adaptée au problème de classification, mais, dans le contexte des canaux auxiliaires, il correspond au taux de réussite d'une attaque simple. Pour cette raison, nous utiliserons davantage une autre métrique, plus adaptée à évaluer les performances d'une attaque avancée : nous notons $N^\star$ le nombre minimal de traces nécessaires pour rendre le rang moyen de la bonne clé (*guessing entropy* en anglais) égal à 1 de façon stable.

Expériences en cas de contremesure logicielle

Pour la première expérience, nous avons implémenté sur un microprocesseur Atmega328P une contremesure par interruption aléatoire afin de protéger la fuite d'une seule opération de la forme $Z = \mathrm{HW}(\mathrm{Sbox}(P \oplus K))$. La contremesure consiste à introduire une boucle de r instructions *nop*, avec r tiré aléatoirement dans $[0, 127]$. Ceci provoque un décalage aléatoire des PoIs des traces acquises, sans déformation des motifs liés aux cycles d'horloge. Nous avons acquis un nombre suffisamment petit de traces pour s'assurer de ne jamais pouvoir observer dans les acquisitions toutes valeurs pos-

Table 1 – Résultat de l'attaque CNN, pour diverses techniques DA, en présence d'interruption aléatoire. Pour chaque technique, 4 valeurs sont données : en position a le taux maximal de bonne prédiction d'apprentissage, en position b taux maximal de bonne prédiction de validation, en position c le taux de bonne prédiction de test, obtenu sur les traces d'attaque, en position d la valeur de $N^\star$.

		SH_0		SH_{100}		SH_{500}	
a	b	100%	25.9%	100%	39.4%	**98.4%**	**76.7%**
c	d	27.0%	>1000	31.8%	101	**78.0%**	**7**

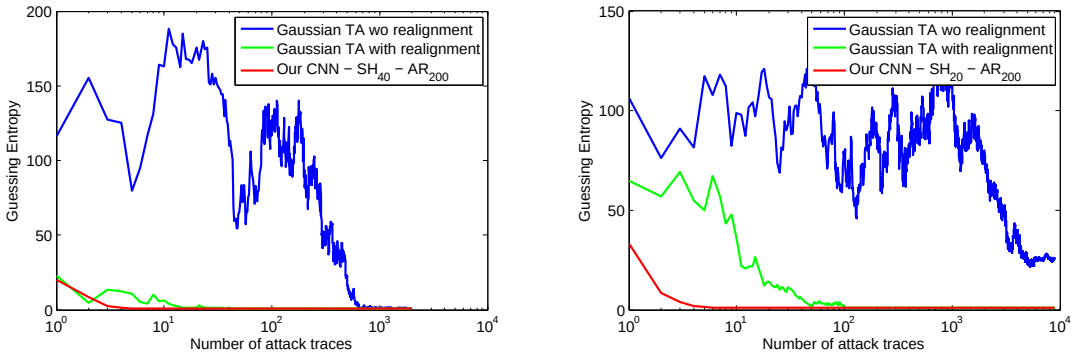

Figure 8 – Comparison entre une attaque template, avec et sans réalignement, et notre stratégie CNN, sur les bases *DS_low_jitter* (left) et *DS_high_jitter* (right).

sibles de Z dans toutes les positions possibles. Ceci pour pouvoir conclure, en cas de succès, que la méthode a su extraire du signal des informations invariantes par décalage.

Le tableau 1 résume les résultats obtenus. En comparant les taux maximaux de bonne prédiction d'apprentissage et de validation, on a un aperçu du risque de sur-apprentissage. Quand l'augmentation de données n'est pas appliquée (cas SH_0) le sur-apprentissage est total : la classification réussit à 100% sur l'ensemble d'apprentissage alors que sur l'ensemble de validation elle reste à 27%. Ceci signifie qu'aucune caractéristique informative a été apprise, amenant à une attaque non conclusive ($N^\star > 1,000$). Nous remarquons que l'augmentation de données diminue fortement le sur-apprentissage : pour SH_{500} l'ensemble d'apprentissage n'est jamais complètement appris et le taux de bonne prédiction de validation s'élève à 78%, pour une guessing entropy de 1 avec seulement $N^\star = 7$ traces d'attaque. Ces résultats confirment que le modèle CNN est capable de caractériser une large fenêtre de points de façon efficace en présence de désynchronisation logicielle.

Expériences en cas de contremesure matérielle simulée

Les contremesures matérielles de gigue agissent au niveau de l'horloge du composant ; en perturbant sa fréquence, des déformations temporelles apparaissent. Les traces acquises présentent des désynchronisations qui s'accumulent au cours des cycles d'horloge. Les expériences suivantes visent à tester le réseau CNN en présence de ce type de déformation. Dans un premier temps nous avons mené ces tests sur des signaux parfaitement synchrones au moment de l'acquisition et désynchronisés de façon artificielle afin de pouvoir maîtriser l'effet déformant. Les résultats, résumés dans le tableau 2, sont finalement obtenus sur deux bases de données, *DS_low_jitter* et *DS_high_jitter* respectivement moins affectée et plus affectée par la désynchronisation. Ces résultats confirment que l'approche par CNN reste efficace en présence de ce type de déformation, et montrent les bienfaits de l'application des techniques DA.

Expériences en cas de contremesure matérielle réelle

Les résultats prometteurs obtenus dans le contexte simulé se retrouvent aussi dans le dernier scénario d'expérience, où la cible est une implémentation matérielle de l'AES sur composant sécurisé moderne. L'implémentation est protégée par un effet de jitter sur l'horloge. Comme montré dans le tableau 3, l'attaque CNN est efficace et les techniques DA atténuent le sur-apprentissage du réseau en améliorant

Table 2 – Résultat de l'attaque CNN, pour différentes techniques DA, en présence de désynchronisation matérielle simulée. Légende identique au tableau 1.

DS_low_jitter								
a	b	SH_0		SH_{20}		SH_{40}		SH_{200}
c	d							
AR_0		100.0%	68.7%	99.8%	86.1%	98.9%	84.1%	
		57.4%	14	82.5%	6	83.6%	6	
AR_{100}		87.7%	88.2%	82.4%	88.4%	81.9%	89.6%	
		86.0%	6	87.0%	5	87.5%	6	
AR_{200}		83.2%	88.6%	81.4%	86.9%	**80.6%**	**88.9%**	
		86.6%	6	85.7%	6	**87.7%**	**5**	
AR_{500}								85.0% 88.6%
								86.2% 5

DS_high_jitter								
a	b	SH_0		SH_{20}		SH_{40}		SH_{200}
c	d							
AR_0		100%	45.0%	100%	60.0%	98.5%	67.6%	
		40.6%	35	51.1%	9	62.4%	11	
AR_{100}		90.4%	57.3%	76.6%	73.6%	78.5%	76.4%	
		50.2%	15	72.4%	11	73.5%	9	
AR_{200}		83.1%	67.7%	**82.0%**	**77.1%**	82.6%	77.0%	
		64.0%	11	**75.5%**	**8**	74.4%	8	
AR_{500}								83.6% 73.4%
								68.2% 11

		$\mathrm{SH}_0\mathrm{AR}_0$		$\mathrm{SH}_{10}\mathrm{AR}_{100}$		$\mathrm{SH}_{20}\mathrm{AR}_{200}$	
a	b	35.0%	1.1%	12.5%	1.5%	**10.4%**	**2.2%**
c	d	1.2%	137	1.3%	89	**1.8%**	**54**

Table 3 – Résultats de l'attaque CNN sur carte à puce protégée par jitter.

l'attaque. Dans ce contexte nous comparons la stratégie classique, qui consiste à effectuer un réalignement, suivi d'un choix minutieux des PoIs, puis d'une attaque template, avec la stratégie par CNN sans réalignement ni extraction explicite de caractéristiques. L'attaque CNN se montre légèrement plus efficace même en absence de pré-traitements.

4 Conclusions et Perspectives

Dans cette thèse, nous nous sommes concentrés sur les attaques par profilage. L'opportunité de caractériser les fuites de la cible ouvre les portes à des approches optimales, permettant l'estimation des distributions de probabilité conditionnelle nécessaire à identifier la clé secrete par maximum *a-posteriori*. Cependant, l'effort d'estimer les distributions de probabilité de données largement multivariées est empêché par la *malédiction de la dimension*. Nos premiers efforts se sont consacrés au développement de technique de réduction de dimension, et nous avons proposé deux contributions à ce sujet. Dans une troisième contribution nous abordons la malédiction de la dimension à l'aide de modèles par réseaux neuronaux.

Le fil rouge de cette thèse a été la croissante conscience du fait que les problèmes pratiques auxquels sous sommes confrontés dans le domaine des attaques par canaux auxiliaires sont presque identiques dans d'autres domaines d'application, et l'approche par apprentissage automatique s'est avérée gagnante dans plusieurs d'entre eux. Nous avons participé alors à une conversion des problématiques SCA, en passant d'une approche statistique classique, à une approche par apprentissage

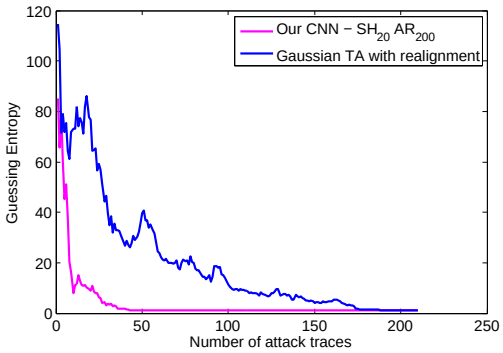

Figure 9 – Comparaison entre l'attaque template gaussienne avec réalignement, et la stratégie par CNN, sur carte à puce protégée par jitter.

automatique. Nous croyons que cette conversion mérite d'être poursuivie dans le future. Une prochaine étape devrait être la définition d'une tâche d'apprentissage automatique qui soit parfaitement adaptée au contexte des attaques avancée : en effet, jusqu'à présent, nous avons utilisé des outils dédiés à la tâche de la classification, qui ne correspond pas parfaitement à ce type d'attaque. Des métriques et des critères d'optimisation spécialisés pour les SCA devraient être proposés. Deuxièmement, dans l'optique de renforcer la résistance des dispositifs sécurisés, des méthodes d'analyse des modèles d'apprentissage profond menant à la réussite des attaques sont nécessaires, afin d'interpréter l'action de ces modèles, identifier les caractéristiques des signaux qui contribuent plus à leur réussite, retrouver et à terme éliminer les vulnérabilités de l'implémentation qui ont permis à ces caractéristiques de fuir à travers les canaux auxiliaires.

Références

[1] C. Archambeau, E. Peeters, F.-X. Standaert, and J.-J. Quisquater. Template attacks in principal subspaces. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin Heidelberg, 2006.

[2] Lejla Batina, Jip Hogenboom, and Jasper G.J. van Woudenberg. Getting more from pca : First results of using principal component analysis for extensive power analysis. In Orr Dunkelman, editor, *Topics in Cryptology CT-RSA 2012*, volume 7178 of *Lecture Notes in Computer Science*, pages 383–397. Springer Berlin Heidelberg, 2012.

[3] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Enhancing dimensionality reduction methods for side-channel attacks. In *International Conference on Smart Card Research and Advanced Applications*, pages 15–33. Springer, 2015.

[4] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Kernel discriminant analysis for information extraction in the presence of masking. In *International Conference on Smart Card Research and Advanced Applications*, pages 1–22. Springer, 2016.

[5] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Convolutional neural networks with data augmentation against jitter-based countermeasures - profiling attacks without pre-processing. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 45–68. Springer, 2017.

[6] Claude Carlet, Jean-Luc Danger, Sylvain Guilley, Houssem Maghrebi, and Emmanuel Prouff. Achieving side-channel high-order correlation immunity with leakage squeezing. *Journal of Cryptographic Engineering*, 4(2) :107–121, 2014.

[7] Suresh Chari, JosyulaR. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski, Cetin K. Koc, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer Berlin Heidelberg, 2003.

[8] Omar Choudary and Markus G Kuhn. Efficient stochastic methods : Profiled attacks beyond 8 bits. *IACR Cryptology ePrint Archive*, 2014.

[9] Omar Choudary and Markus G Kuhn. Efficient template attacks. In *Smart Card Research and Advanced Applications*, pages 253–270. Springer, 2014.

[10] Thomas Eisenbarth, Christof Paar, and Bjorn Weghenkel. Building a side channel based disassembler. In MarinaL. Gavrilova, C.J.Kenneth Tan, and EdwardDavid Moreno, editors, *Transactions on Computational Science X*, volume 6340 of *Lecture Notes in Computer Science*, pages 78–99. Springer Berlin Heidelberg, 2010.

[11] Paul C Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Annual International Cryptology Conference*, pages 104–113. Springer, 1996.

[12] Dimitrios Mavroeidis, Lejla Batina, Twan van Laarhoven, and Elena Marchiori. PCA, eigenvector localization and clustering for side-channel attacks on cryptographic hardware devices. In PeterA. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 7523 of *Lecture Notes in Computer Science*, pages 253–268. Springer Berlin Heidelberg, 2012.

[13] Robert Specht, Johann Heyszl, Martin Kleinsteuber, and Georg Sig. Improving non-profiled attacks on exponentiations based on clustering and extracting leakage from multi-channel high-resolution EM measurements. In *Sixth International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE 2015)*, 2015.

[14] François-Xavier Standaert and Cedric Archambeau. Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems CHES 2008*, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer Berlin Heidelberg, 2008.