



Attaques par canaux cachés : expérimentations avancées sur les attaques template

Abdelaziz M. Elaabid

► To cite this version:

Abdelaziz M. Elaabid. Attaques par canaux cachés : expérimentations avancées sur les attaques template. Cryptographie et sécurité [cs.CR]. Université Paris VIII Vincennes-Saint Denis; Ecole nationale supérieure des telecommunications - ENST, 2011. Français. NNT: . tel-00937136

HAL Id: tel-00937136

<https://theses.hal.science/tel-00937136>

Submitted on 27 Jan 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



École Doctorale : CLI
laboratoire : LAGA
Équipe d'accueil : MPPI

THÈSE

présentée pour obtenir le grade de docteur

de l'université de Paris 8

Spécialité : Informatique

Moulay Abdelaziz EL AABID

Attaques par canaux cachés : expérimentations avancées sur les attaques *template*

Soutenue le 07/12/2011 devant le jury composé de

David NACCACHE
Reynald LERCIER
Jean Luc DANGER
Emmanuel PROUFF
Thanh-Ha LE
Philippe HOOGVORST
Sylvain GUILLEY
Claude CARLET

Rapporteur
Rapporteur
Examineur
Examineur
Examinatrice
Examineur
Co-encadrant de thèse
Directeur de thèse

Remerciements

En première instance, je tiens à exprimer mes remerciements les plus sincères à mon directeur de thèse, Monsieur Claude CARLET, pour m'avoir conseillé, encouragé et soutenu durant toutes ces années de thèse. Il a toujours été disponible et présent pour m'accompagner à toutes les étapes. Grâce à son soutien, j'ai pu être allocataire, moniteur, et puis ATER à l'université Paris 8.

Je tiens également à remercier chaleureusement Sylvain GUILLEY, qui a concrètement dirigé cette thèse. Il m'a accompagné dans mes débuts de chercheur et m'a beaucoup apporté scientifiquement. Je le remercie pour son encadrement et sa disponibilité.

Je remercie David NACCACHE, Reynald LERCIER, Jean Luc DANGER, Emmanuel PROUFF, Thanh-Ha LE, et Philippe HOOGVORST de me faire l'honneur de constituer mon jury de thèse.

Je tiens à remercier tous mes collègues de Paris 8, et tout particulièrement Mme DURAND-RICHARD, qui m'a fait découvrir l'histoire des sciences et de la cryptologie. Une découverte cruciale pour mon orientation vers ce domaine de recherche.

Je remercie Philippe GUILOT, Sihem MESNAGER, Farid MOKRANE, Jean Jacques BOURDIN, et Patrick GREUSSAY mes professeurs devenus mes collègues, pour leurs soutiens et leurs conseils.

Je tiens également à remercier mes amis et mes collègues de Telecom Paristech pour les bons moments qu'on a passé ensemble. En particulier Nidhal SELMAN, Laurent SAUVAGE, Youssef SOUSSI, Housem MAGHREBI, Shivam BHASIN, Maxime NASSAR, Sebastien THOMAS, Olivier MENARD, Sumanta CHAUDHURI, taoufik CHOUTA, et Zouha CHERIF. Sans oublier Tarik GRABA, Guillaume DUC, philippe HOOGVORST, philippe MATHERAT et Jean Luc DANGER, pour les différents échanges concernant mon sujet de thèse et pour l'intérêt qu'ils ont porté à mon travail.

Durant cette thèse, de réelles amitiés se sont créées avec Zahir LARABI et Chadi JABBOUR que je tiens à remercier également.

Enfin, Je remercie chaque membre de ma famille, mes frères Adnane et Hakim, sans oublier mes amis Abdelhadi, Adnane et Adil, pour notre complicité, et pour leurs encouragements.

Mes plus grands remerciements sont réservés à mes parents. Mon père sera sûrement

fier de moi, car c'est en partie pour lui que j'ai franchi le pas et entamer cette aventure. Ma mère m'a toujours soutenu, protégé et entouré de sa tendresse. Je ne pourrais jamais les remercier assez.

Souad, mon épouse, aura mon dernier témoignage de reconnaissance. Elle m'a toujours étonné par son grand sens de la responsabilité et son dévouement à notre famille. Elle m'a supporté, aidé et accompagné à chaque instant. Nos enfants, Aya et Aymane nés pendant la thèse, ont aussi su me soutenir en remplissant ma vie de gaieté et de bonheur.

Résumé

Au début des années 90, l'apparition de nouvelles méthodes de cryptanalyse a bouleversé la sécurité des dispositifs cryptographiques. Ces attaques se basent sur l'analyse de consommation en courant lorsque le microprocesseur d'une carte est en train de dérouler l'algorithme cryptographique. Dans cette thèse nous explorons, principalement, les attaques *template*, et y apportons quelques améliorations pratiques notamment par l'utilisation de différentes techniques de traitement du signal. Nous commençons par étudier l'efficacité de ces attaques sur des mises en oeuvre matérielles non protégées, et explorons au fur et à mesure quelques modèles de fuite d'information. Après cela, nous examinons la pertinence du cadre théorique sur les attaques par profilage présenté par F.-X. Standaert et al. à Eurocrypt 2009. Ces analyses consistent en des études de cas basées sur des mesures de courant acquises expérimentalement à partir d'un accélérateur cryptographique. À l'égard de précédentes analyses formelles effectuées sur des mesures par « simulations », les investigations que nous décrivons sont plus complexes, en raison des différentes architectures et de la grande quantité de bruit algorithmique. Dans ce contexte, nous explorons la pertinence des différents choix pour les variables sensibles, et montrons qu'un attaquant conscient des transferts survenus pendant les opérations cryptographiques peut sélectionner les distingueurs les plus adéquats, et augmenter ainsi son taux de succès. Pour réduire la quantité de données, et représenter les modèles en deux dimensions, nous utilisons l'analyse en composantes principales (ACP) et donnons une interprétation physique des valeurs propres et vecteurs propres. Nous introduisons une méthode basée sur le seuillage de la fuite de données pour accélérer le profilage ainsi que l'attaque. Cette méthode permet de renforcer un attaquant qui peut avec un minimum de traces, améliorer 5 fois sa vitesse dans la phase en ligne de l'attaque. Aussi, il a été souligné que les différents modèles utilisés, ainsi que les échantillons recueillis durant la même acquisition peuvent transporter des informations complémentaires. Dans ce contexte, nous avons eu l'occasion d'étudier comment combiner au mieux différentes attaques en se basant sur différentes fuites. Cela nous a permis d'apporter des réponses concrètes au problème de la combinaison des attaques. Nous nous sommes concentrés également sur l'identification des problèmes qui surgissent quand il y a une divergence entre les *templates* et les traces attaquées. En effet, nous montrons que deux phénomènes peuvent entraver la réussite des attaques lorsque les *templates* sont obsolètes, à savoir, la désynchronisation des traces, et le redimensionnement des traces en amplitudes. Nous suggérons deux remèdes pour contourner ce type de problèmes : le réajustement des signaux et la normalisation des campagnes d'acquisitions. Finalement, nous proposons quelques méthodes du traitement du signal dans le contexte des attaques physiques. Nous montrons que lorsque les analyses sont effectuées en multi-résolution, il y a un gain considérable en nombre de traces nécessaires pour récupérer la clé secrète, par rapport à une attaque ordinaire.

Abstract

In the 90's, the emergence of new cryptanalysis methods revolutionized the security of cryptographic devices. These attacks are based on power consumption analysis, when the microprocessor is running the cryptographic algorithm. Especially, we analyse in this thesis some properties of the *template attack*, with examples from attacks against an unprotected ASIC implementation. We point out that the efficiency of *template* attacks can be unleashed by using a relevant power model, and we provide some practical improvements by the use of different signal processing techniques. Furthermore, we investigate the relevance of the theoretical framework on profiled SCAs presented by F.-X. Standaert et al. at Eurocrypt 2009. The analyse consists in a case-study based on side-channel measurements acquired experimentally from a hardwired cryptographic accelerator. Therefore, with respect to previous formal analyses carried out on software measurements or on simulated data, the investigations we describe are more complex, due to the underlying chip's architecture and to the large amount of algorithmic noise. In this context, we explore the appropriateness of different choices for the sensitive variables, and we show that a skilled attacker aware of the register transfers occurring during the cryptographic operations can select the most adequate distinguisher, thus increasing its success rate. The principal component analysis (PCA) is used to represent the *templates* in some dimensions, and we give a physical interpretation of the *templates* eigenvalues and eigenvectors. We introduce a method based on the thresholding of leakage data to accelerate the profiling or the matching stages. This method empowers an attacker, in that it saves traces when converging towards correct values of the secret. Concretely, we demonstrate a 5 times speed-up in the on-line phase of the attack. Also, it has been underlined that the various samples garnered during the same acquisition can carry complementary information. In this context, there is an opportunity to study how to best combine many attacks with many leakages from different sources or using different samples from a single source. That brings some concrete answers to the attack combination problem. Also we focus on identifying the problems that arise when there is a discrepancy between the *templates* and the traces to match. Based on a real-world case-study, we show that two phenomena can hinder the success of *template* attacks when the precharacterized *templates* are outdated : the traces can be desynchronized and the amplitudes can be scaled differently. Then we suggest two remedies to cure the *template* mismatches : waveform realignment and acquisition campaign normalization. Eventually, we propose in a methodological manner, some applications of Wavelet transforms in the side-channel context. We show that SCAs when performed with a multi-resolution analysis are much better, in terms of security metrics, than considering only the time or the frequency resolution. Actually, the gain in number of traces needed to recover the secret key is relatively considerable with respect to an ordinary attack.

Table des matières

Remerciements	i
Résumé	iii
Abstract	v
1 Introduction générale	1
1.1 Avant-propos	1
1.2 Standard de chiffrement de données : l'algorithme DES	3
1.3 Standard de chiffrement de données : l'algorithme AES	5
1.4 Les attaques algorithmiques	8
1.5 Les attaques physiques	9
1.5.1 Les attaques par analyse de consommation	10
1.5.2 L'analyse différentielle de consommation, ou DPA	11
1.5.2.1 Acquisition des données	12
1.5.2.2 Exploitation de ces données	12
1.5.3 L'analyse du courant par corrélation, ou CPA	13
1.6 Une étude théorique	13
1.6.1 Adversaire	14
1.6.2 Définitions	14
1.7 Problématique	15
1.8 Plan du document	16
2 Configuration, mises en œuvres et métriques	19
2.1 Les cryptoprocresseurs étudiés	19
2.1.1 Architecture SECMAT	20
2.1.2 Les acquisitions	21
2.2 Métriques de comparaison	21
2.2.1 L'entropie pour quantifier l'information	22

2.2.1.1	L'entropie conditionnelle	23
2.2.1.2	Définition du taux de succès et l'entropie estimée	23
3	Les attaques <i>template</i>	25
3.1	Schéma d'attaque	25
3.2	Un modèle probabiliste	26
3.3	La vraisemblance	28
3.4	L'attaque <i>template</i> avec recherche de points d'intérêt	29
3.4.1	La recherche de points d'intérêt	29
3.4.2	Attaque <i>template</i> et points d'intérêt	30
3.4.2.1	Phase d'apprentissage	30
3.4.2.2	Phase d'attaque proprement dite	31
3.5	L'attaque <i>template</i> dans les sous-espaces principaux	31
3.5.1	L'analyse en composantes principales	31
3.5.2	Recherche des composantes principales	32
3.5.3	Attaque <i>template</i> avec ACP	33
3.5.3.1	Profilage	33
3.5.3.2	Attaque en ligne	35
4	Vers un modèle pour les attaques <i>template</i> ?	37
4.1	Les attaques <i>template</i> et la génération de clés de tours	37
4.1.1	Campagne #1 : clé nulle, sauf 6 bits	38
4.1.2	Campagne #2 : clés aléatoires	38
4.1.3	Comparaison des attaques	39
4.2	Nouveau modèle d'attaque	41
4.2.1	Les attaques <i>template</i> avec le modèle du poids de Hamming	41
4.2.2	Les attaques <i>template</i> et le modèle de la distance de Hamming	43
5	Améliorations pratiques des attaques <i>template</i>	51
5.1	Les attaques <i>template</i> : de la ACP à SECMAT	51
5.2	Amélioration des attaques grâce à un modèle de fuite adéquat.	55
5.2.1	Taux de succès	55
5.2.2	L'entropie conditionnelle	58
5.2.3	Modèles Invisibles	60
5.3	Suppression du bruit par seuillage	60
5.4	Discussion	61

6	Quelles attaques combinées ?	67
6.1	Attaques combinées et métriques basées sur des partitionnements multiples	69
6.1.1	Les variables sensibles et les attaques <i>template</i>	71
6.1.1.1	Les modèles combinés	71
6.1.1.2	Premier Choix : Évaluation avec attaque limitée.	72
6.1.1.3	Second Choix : Évaluation avec entraînement limité.	73
6.1.1.4	L'entropie conditionnelle	73
6.2	Attaques par corrélation combinée	75
6.2.1	Des techniques pour retrouver les POIs	76
6.2.1.1	Le sosd <i>vs</i> sost <i>vs</i> IM.	76
6.2.1.2	L'ACP à distance.	77
6.2.2	Les échantillons combinés	80
6.2.2.1	Les observations	80
6.2.2.2	Combinaison principale d'échantillons et résultats.	80
7	La portabilité des <i>templates</i>	85
7.1	Méthodologie	85
7.2	Re-synchronisation horizontale : POC et AOC	89
7.2.1	AOC : La Corrélacion en amplitude	90
7.2.2	POC : Corrélacion en phase	91
7.2.3	AOC vs POC	92
7.3	Attaque avec de faux décalages	95
7.4	Homothétie verticale	97
7.5	Modélisation pour des campagnes mal alignées	99
7.6	Estimation de la portabilité	103
8	Les bienfaits du traitement du signal	107
8.1	L'analyse multi-résolution	108
8.1.1	Transformée de Fourier	109
8.1.2	Transformée de Fourier à Court Terme	109
8.1.3	Transformée en ondelettes	110
8.1.3.1	Transformée en ondelettes continu (CWT)	110
8.1.3.2	Transformée en ondelettes discrète (DWT)	111
8.2	Applications aux SCAs	112
8.2.1	Les ondelettes pour la détection des motifs cryptographiques	112

8.2.2	L'information mutuelle et les ondelettes pour filtrage des traces	113
8.2.2.1	Filtrage des traces avec les ondelettes	113
8.2.2.2	Amélioration du filtrage du bruit	115
8.2.3	Les ondelettes pour la compression des traces	115
8.2.4	Les ondelettes pour retrouver les clés	117
8.2.4.1	Applications des ondelettes et résultats	117
9	Attaques aveugles et différents obstacles	123
9.1	Attaques sur <i>DPA contest 2</i>	123
9.1.1	Présentation du <i>DPA contest</i>	123
9.1.2	Plate-forme d'attaque <i>template</i>	123
9.1.3	Résultats de l'attaque <i>template</i> au concours <i>DPA contest</i>	124
9.1.4	Comparaison avec l'attaque stochastique	124
9.1.4.1	L'attaque stochastique	124
9.1.4.2	Comparaison des attaques <i>template</i> et stochastique	125
9.2	Attaques et bruit exotique	126
9.2.1	Comment situer la fuite?	126
9.2.2	Les disproportions des bits de données	127
9.2.3	Investigations approfondies sur les bits d'entrée d'une <i>S-box</i> DES	128
9.3	Attaques et masquage	131
9.3.1	Méthode de duplication	131
9.3.2	Exploitation des fuites exotiques	132
10	Conclusions et perspectives	135
Annexes		139
A		139
A.1	Les valeurs de la clé et du registre CD au premier tour	139
A.2	Résultats détaillés du DPAcontest	140
A.3	Suite des résultats du chapitre 9	140
Publications		147
A.4	Articles avec actes	147
A.5	Article avec acte soumis en attente	147
A.6	Communications internationales	147
A.7	Communications nationales	148

Bibliographie

154

Table des figures

1.1	Un tour de l'algorithme DES.	4
1.2	Chargement de la clé.	5
1.3	Les boîtes de substitution.	7
1.4	Exemple d'une boîte de substitution.	7
1.5	chiffrement AES	8
2.1	Chemin de données du DES utilisé dans ce rapport	21
2.2	Plateforme des attaques	22
3.1	Exemple de trace	26
3.2	Les points d'intérêt.	30
4.1	Résultats de l'attaque <i>template</i> utilisant 32 composantes.	38
4.2	Densités de probabilité correspondants à la campagne #1	39
4.3	Densités de probabilité correspondant à la campagne #2	40
4.4	1 ^{er} vecteur propre de la campagne #1 avec l'"Id" comme classification	40
4.5	1 ^{er} vecteur propre de la campagne #2 avec l'"Id" comme classification	41
4.6	Les 2 ⁶ valeurs propres de la campagne d'acquisition #2 avec ϕ_0	42
4.7	La logique combinatoire dans le registre "C"	43
4.8	Densités de probabilité de la campagne #2 avec ϕ_1	44
4.9	Comparaison des 3 premières valeurs propres/bits de la clé	44
4.10	les 2 ⁶ valeurs propres de la campagne d'acquisition #2 avec ϕ_1	45
4.11	Premier vecteur propre de la campagne d'acquisitions #2 avec ϕ_1	46
4.12	Second vecteur propre de la campagne d'acquisitions #2 avec ϕ_1	46
4.13	Les densités de probabilité pour la campagne d'acquisition #2 avec ϕ_2	47
4.14	les 2 ⁶ valeurs propres pour la campagne d'acquisition #2 avec ϕ_2	47
4.15	Premier vecteur propre de la campagne d'acquisition #2 avec ϕ_2	48

4.16	Traces différentielles obtenues avec ϕ_1 (en haut) et ϕ_2 (en bas).	49
5.1	Circulation des données dans le premier tour DES	53
5.2	Les différents modèles du tableau 5.1.	54
5.3	Différence entre “bon”/“mauvais” vecteur propre pour le modèle B	56
5.4	Les vecteurs propres, pour les modèles A, B, C, D & E	56
5.5	Taux de succès, pour les modèles A, B, C, D & E	57
5.6	Comparaison du taux de succès entre tous les modèles	58
5.7	Comparaison des moyennes des modèles D et E avec le 1 ^{er} vecteur propre	58
5.8	Comparaison des entropies conditionnelles	59
5.9	Comparaison du taux de succès avec seuillages	61
5.10	Amélioration du taux de succès avec le seuillage.	62
5.11	Amélioration de l’entropie conditionnelle avec le seuillage.	62
5.12	Évaluation <i>vs</i> Diagramme d’attaque	64
5.13	Fuite <i>vs</i> diagramme d’attaque	65
5.14	Les risques de confiance	65
6.1	vecteurs propres et seuillage de 40%	73
6.2	Comparaison entre M1, M2, M3/Taux de succès : premier choix	74
6.3	Comparaison M1, M2 et M3 /Taux de succès/second choix	74
6.4	Comparaison de l’entropie conditionnelle entre les différents modèles. . .	75
6.5	CPA, sosd, sost, IM	78
6.6	Taux de succès de la CPA après ACP à 0 cm.	79
6.7	Taux de succès de la CPA après ACP à 25 cm.	80
6.8	Corrélations à 25cm (fausse/bonne hypothèse)	81
6.9	Amélioration de l’attaque par combinaison de POIs.	83
6.10	Amélioration de l’attaque par combinaison de POIs.	83
7.1	Trace Moyenne et écart-type pour la campagne A.	88
7.2	Trace Moyenne et écart type pour la campagne B.	88
7.3	Trace Moyenne et écart type pour la campagne C.	88
7.4	Re-synchronisation et quelque résultats	93
7.5	comparaison avant synchronisation	94
7.6	Comparaison après synchronisation	94
7.7	Vecteurs propres et comparaison	95
7.8	À la recherche du meilleur décalage possible pour la campagne B/A. . .	97

7.9	Les métriques dans des conditions idéales	98
7.10	les attaques <i>template</i> et l'homothétie	98
7.11	Comparaison avec différents facteurs de décalages	100
7.12	Les taux de succès en fonction du bruit dans les traces attaquées.	102
7.13	Taux de succès pour des coefficients (a) positifs et (b) négatifs	103
7.14	Efficacité de l'attaque	104
7.15	Les attaques <i>template</i> avec pré-caractérisation vs DPA/CPA	105
8.1	Une illustration d'une décomposition DWT à 3 niveaux	112
8.2	Illustration de la CWT sur un triple chiffrement AES.	114
8.3	Calcul de l'IM avec DWT au premier niveau.	116
8.4	Comparaison du taux de succès pour les 8 S-boxs	118
8.5	Comparaison de l'entropie estimée pour les 8 S-boxs	119
8.6	Taux de succès et attaques <i>template</i> avec ondelettes	120
8.7	Entropies estimées et attaques <i>template</i> avec ondelettes	120
8.8	Taux de succès de la CPA et ondelettes	121
8.9	Entropies estimées de la CPA et ondelettes	122
8.10	Les ondelettes et les SCAs	122
9.1	Attaque <i>template</i> vs attaque stochastique aux S-boxes 2 et 3	125
9.2	Bloc Feistel	126
9.3	Consommation des quatre bits de sortie de la <i>S-box</i> #1	128
9.4	Vecteurs propres : trace entière et fenêtre [7500 :10500	129
A.1	Taux de succès partiels et entropies estimées partielles.	141
A.2	Les vecteurs propres correspondant à chacun des six bits de la S-box 1 .	142
A.3	Les vecteurs propres correspondant à chacun des six bits de la S-box 2 .	145
A.4	Les taux de succès correspondant à chacun des six bits	146

Liste des tableaux

1.1	Les décalages pendant le " <i>key schedule</i> " DES.	6
4.1	Comparaison des valeurs propres vs fonctions de classification	50
5.1	Les cinq modèles examinés.	53
6.1	Évolution des corrélations et signal final après estimation sur 2.000 traces	70
6.2	Information et Probas du poids de Hamming sur 8 bits	81
7.1	Caractéristiques de configuration	87
7.2	Décalage temporel optimale par POC	95
A.1	les valeurs de la clé K du DES et du registre CD au premier tour.	140
A.2	DPAcontest et résultats partiels de l'entropie estimée	143
A.3	DPAcontest et résultats partiels du taux de succès	144

Chapitre 1

Introduction générale

1.1 Avant-propos

À notre époque, la cryptologie connaît une réelle évolution du fait, en partie, de deux facteurs essentiels :

1. d'un côté, le développement des moyens de télécommunications, qui ont eu pour effet de multiplier les activités où intervient la cryptologie, et
2. aussi, d'un autre côté, l'échange de savoir au niveau académique.

En effet, jusqu'à une date relativement récente, la cryptologie était le terrain de compétences réservées aux services officiels. Depuis les années 80, les universitaires se sont emparés du sujet, et ont considérablement accéléré son évolution.

La diffusion des techniques de chiffrement a permis d'assurer l'intégrité et la confidentialité des communications, tout en permettant à tout-à-chacun de se convaincre de l'honnêteté des méthodes de signature et/ou de chiffrement, conditions indispensables à une utilisation commerciale généralisée. Ainsi, depuis une vingtaine d'années, la cryptologie s'est enrichie de nouvelles techniques de chiffrement électronique, qui ont contribué à l'optimisation des algorithmes et de leur mise en œuvre matérielle. Cette dernière est une étape importante (ce que nous allons souligner tout au long de ce manuscrit) dans la sécurité et aussi dans l'étude de l'algorithme.

Il faut savoir que la cryptologie est la branche des mathématiques traitant des procédés cryptographiques, mais aussi des méthodes cryptanalytiques. La cryptographie est la partie consacrée à la protection des données en assurant les principes d'intégrité, d'authenticité et de confidentialité. Différents algorithmes et protocoles de chiffrement ont été conçus dans ce but.

En général, ces algorithmes consistent en l'application d'une bijection entre deux espaces de messages, choisie dans une famille de bijections à l'aide d'une donnée secrète : la clé. Cette transformation peut être une fonction de **chiffrement** $E(M) = C$, ou de **déchiffrement** $D(C) = M$, tel que C est le texte chiffré et M le texte en clair.

Nous pouvons, ainsi, grâce à ce type de procédé bénéficier de la sûreté des échanges, mais à priori, seulement avec un petit temps d'avance. Assurément, la cryptographie est en permanence remise en cause par la cryptanalyse qui est l'art de déchiffrer les communications secrètes.

C'est un combat permanent, dont les deux concurrents sont enrichis en permanence par de nouveaux procédés.

Les principales règles qui gèrent cet antagonisme sont les suivantes¹ :

- l'algorithme doit être public et donc connu par tout le monde, et
- sa sécurité doit reposer sur les trois paramètres suivants :
 - la qualité du protocole d'usage de l'algorithme,
 - la qualité de l'algorithme et
 - la longueur de la clé.

En effet, la transparence en matière de conception de primitives cryptographiques permet de mieux évaluer la sécurité des algorithmes.

La notion de clé cryptographique permet de regrouper les algorithmes cryptographiques en deux grandes catégories :

- les algorithmes **symétriques**, à clé unique, connue des deux seuls participants et
- les algorithmes **asymétriques** avec, pour chaque participant, une clé publique, connue de tous et une clé secrète, connue du seul participant.

Cette clé qui est changée régulièrement dans le cas de chiffrements symétriques est *mixée* avec les messages à chiffrer. Plus elle est longue plus il est difficile de la retrouver par une attaque exhaustive. Cette l'attaque est la plus simple, du point de vue conceptuel. Elle consiste à essayer toutes les clés possibles, et sa complexité augmente exponentiellement en fonction du nombre de symboles composant la clé de l'algorithme.

Ainsi l'attaquant qui représente la cryptanalyse, dont le but est d'épier l'information secrète, doit essentiellement concentrer ses efforts sur les clés du chiffrement. Ces clés sont par conséquent le point faible de la cryptographie, il faut donc les manipuler avec prudence.

Les progrès de la cryptanalyse entraînent nécessairement des progrès en cryptographie et vice-versa. C'est une évolution sans fin. On peut se demander si la confidentialité des messages ne se trouve pas altérée dans ces progressions.

Beaucoup d'algorithmes de chiffrement sont *théoriquement* robustes contre la cryptanalyse. Mais il est difficile de prédire une prospérité perpétuelle surtout dans la situation actuelle où de nouveaux types d'attaques ont vu le jour telles que les attaques physiques^{1.5}. Dans nos travaux, nous avons principalement attaqué deux algorithmes de chiffrement :

- DES²1.2 et
- AES³1.3.

¹les principes de Kerckhoffs : énoncés par Auguste Kerckhoffs à la fin XIXème siècle dans un article en deux parties. La cryptographie militaire du Journal des sciences militaires (vol. IX, pp. 538, Janvier 1883, pp. 161191, Février 1883).

²Data Encryption Standard

³Advanced Encryption Standard

Ces deux algorithmes ont été mis en œuvre en utilisant :

- soit un circuit dédié de type ASIC⁴,
- soit un circuit programmable de type FPGA⁵.

Avant d’entrer dans les détails, nous commençons par présenter ces deux algorithmes de chiffrement.

1.2 Standard de chiffrement de données : l’algorithme DES

Le Standard de chiffrement de données DES⁶ est un algorithme de chiffrement par blocs produit par le NIST⁷. Un chiffrement par bloc, ou de Feistel⁸ consiste à découper des données en blocs de taille fixe.

Le principe de l’algorithme DES est de transformer un bloc de 64 bits de données confidentielles, appelé texte clair, en un autre bloc de 64 bits de données, appelé texte chiffré. Ceci en utilisant une bijection standardisée paramétrée par une clé de 56-bit. Cette bijection est conçue de telle façon qu’il est impossible de récupérer le texte clair à partir du texte chiffré sans la connaissance de la clé.

Globalement, l’algorithme de chiffrement opéré sur un texte donné, consiste à respecter les étapes suivantes :

1. Fractionner le texte en blocs de 64 bits (8 octets) ;
2. Permuter les blocs avec une permutation initiale IP ;
3. Découper les blocs en deux parties de 32 bits chacune : gauche et droite, nommées G et D ;
4. Assurer *des tours* de permutations et de substitutions répétées 16 fois, en faisant intervenir la clé cryptographique ;
5. Concaténer les parties G et D et faire une permutation inverse de la permutation initiale.

La figure 1.1 décrit un *tour* DES.

Différents « modes opératoires du DES », sont standardisés dans FIPS⁹ 81, pour utiliser DES lorsque les blocs des messages à chiffrer ont une longueur supérieure à 8 octets. Depuis sa création, DES a été exploité par beaucoup de produits cryptographiques (matériel ou logiciel) nécessitant la confidentialité des données. Cependant, à partir de 2001, il a été remplacé par l’AES1.3.

Chaque étape de l’algorithme DES est d’importance, car elle assure, soit des besoins de diffusion pour une bonne propagation des bits, soit des besoins de confusion

⁴Application-Specific Integrated Circuit

⁵Field-Programmable Gate Array

⁶Data Encryption Standard

⁷The National Institute of Standards and Technology(NIST)

⁸Horst Feistel cryptologue à IBM

⁹Federal Information Processing Standards

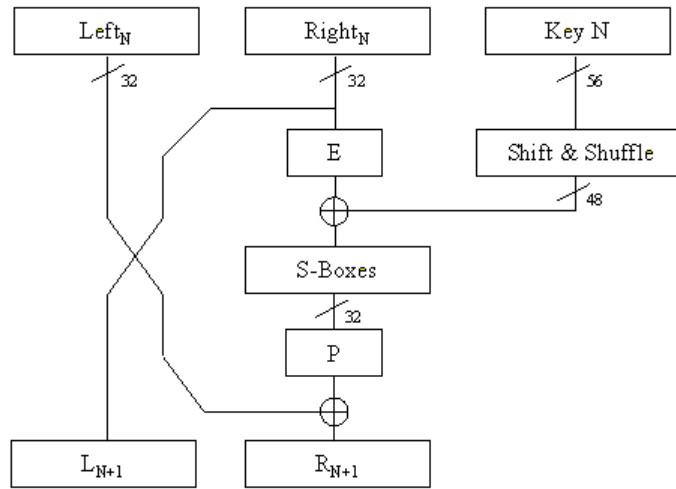


FIG. 1.1 – Un tour de l’algorithme DES.

assurés par les fonctions non linéaires. Ces dernières, sont appelées aussi fonctions de substitutions, ou plus communément “S-box”.

Les figures 1.3, et 1.4 donnent une idée sur une S-box du standard DES.

La clé K est utilisée après un processus qui permet de la segmenter en 16 clés élémentaires (une par tour) K_i . Cette opération est souvent désignée par le terme anglo-saxon “*key scheduling*”. Elle consiste en :

- permutation des bits de K ,
- division du résultat de la permutation en deux blocs de 28 bits.
- pour chaque tour :
 - rotation circulaire de chaque sous-bloc de 1 ou 2 positions vers la gauche,
 - permutation des bits.

Le point faible de cet algorithme reste sa clé qui est de petite taille, et non variable, ce qui n’offre pas la possibilité de l’adapter par rapport aux différents niveaux de sécurité. Nous verrons dans la section 1.4 quelques unes des attaques cryptanalytiques qui sont menées contre DES, et qui ont poussé à des améliorations, comme le 3DES : 3 applications successives du DES sur un bloc de données, avec deux ou trois clés différentes.

Nous allons mettre à l’épreuve, au cours de ces travaux, l’algorithme DES face à la cryptanalyse physique. Nous nous intéresserons à toutes ses spécificités, et regarderons de plus près ses points faibles, et ses points forts. Depuis sa création, le standard DES a été exploité par beaucoup d’applications qui nécessitent la confidentialité des données. Cependant, à partir de 2001, il a été remplacé par un autre standard. Cet algorithme est l’AES¹⁰. Il a d’ailleurs été spécialement étudié dans les travaux de cette thèse.

¹⁰Advanced Encryption Standard

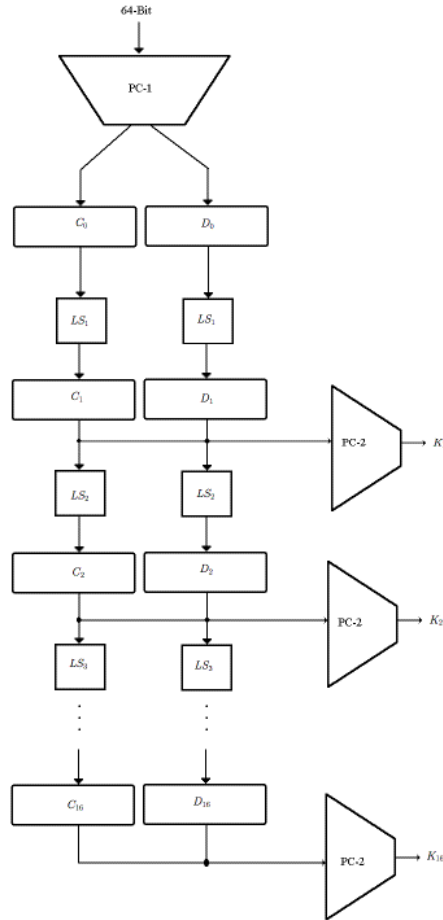


FIG. 1.2 – Chargement de la clé.

1.3 Standard de chiffrement de données : l’algorithme AES

Compte tenu des faiblesses de l’algorithme DES, notamment sa clé trop courte, le NIST a lancé un appel d’offre afin de définir un nouveau standard, qui rende les méthodes d’attaques classiques comme les cryptanalyses linéaire ou différentielle de la section 1.4 au moins aussi difficiles que l’attaque exhaustive.

En Octobre 2000, ce fut l’algorithme ‘Rijndael’ qui a remporté le concours et qui est ainsi devenu le nouveau standard de chiffrement [20].

Ce dernier a été inventé par les chercheurs belges Joan Daemen et Vincent Rijmen. Il opère par blocs de 128 bits et une clé qui, cette fois, est de longueur variable¹¹. La

¹¹c’était une des exigences du concours

Itérations	Décalages à gauche
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

TAB. 1.1 – Les décalages pendant le "key schedule" DES.

longueur est soit de 128, 192 ou de 256 bits et permet d'avoir un compromis entre sécurité et efficacité.

AES est un algorithme itératif, dont le nombre de tours dépend de la longueur de la clé. Ainsi pour une entrée de longueur 128 bits, le nombre de tours est égal à 10, 12 pour 192 bits et 14 pour 256 bits.

Les tours d'AES sont basés sur quatre différentes transformations effectuées de façon répétée et avec un certain ordre. Ces transformations sont décrites dans la figure 1.5.

Le bloc de données de 128 bits est considéré comme une matrice d'octets 4×4 appelée matrice d'états. L'algorithme consiste en une manipulation d'une donnée initiale avec ajout de la clé, pendant 9 tours (lorsque la longueur de la clé est de 128 bits), et un dernier tour (modifié). Un module de chargement de clé (« *key schedule* ») séparé est utilisé pour générer ce qu'on appelle les sous-clés, ou les clés de tours, à partir de la clé initiale ; une sous-clé est également représentée par une matrice d'octets 4×4 .

Un tour entier AES comporte en général quatre étapes :

1. Une substitution non linéaire appliquée à la matrice d'états : « SubBytes ».
2. Une permutation d'octets circulaire sur la même ligne : « ShiftRows ».
3. Une multiplication dans $GF(2^8)$ pour chaque colonne : « MixColumns ».
4. Un simple XOR avec la sortie du registre de clés : « AddRoundKey ».

Ainsi, La transformation SubBytes remplace chaque octet du bloc à l'aide d'une S-box qui est une table de substitution inversible de 256 entrées. Ensuite, on utilise

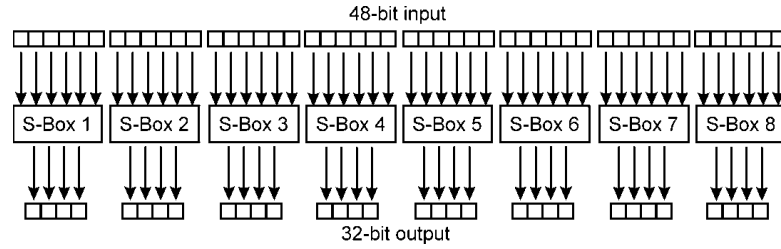


FIG. 1.3 – Les boîtes de substitution.

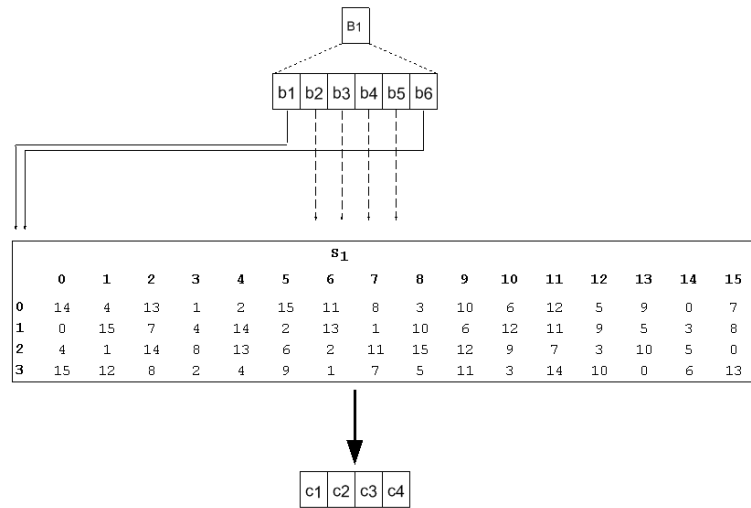


FIG. 1.4 – Exemple d’une boîte de substitution.

la transformation ShiftRows, où chaque ligne de la matrice d’entrée est décalée par 0, 1, 2 ou 3 octets à gauche pour chaque tour. La transformation MixColumns opère individuellement sur chaque colonne. Elle peut être définie comme une multiplication sur la matrice d’états, où chaque colonne est traitée comme un polynôme dans $GF(2^8)$ et est ensuite multipliée modulo $x^4 + 1$ par un polynôme fixé $p(x)$:

$$p(x) = (0x03)x^3 + (0x01)x^2 + (0x02)x + (0x02)$$

Enfin, AddRoundKey permet d’ajouter la sous-clé du tour.

Aujourd’hui AES est très présent dans les communications sécurisées, et gagne de plus en plus en notoriété. Cette importance, fait qu’il devient maintenant une cible privilégiée des études cryptanalytiques. Nous allons voir dans les sections suivantes, qu’il existe différentes manières de forger la cryptanalyse. La plus conventionnelle consiste à examiner les primitives cryptographiques, et leurs propriétés mathématiques. Une autre démarche, s’intéresse plutôt à la mise en œuvre de la primitive dans un matériel de

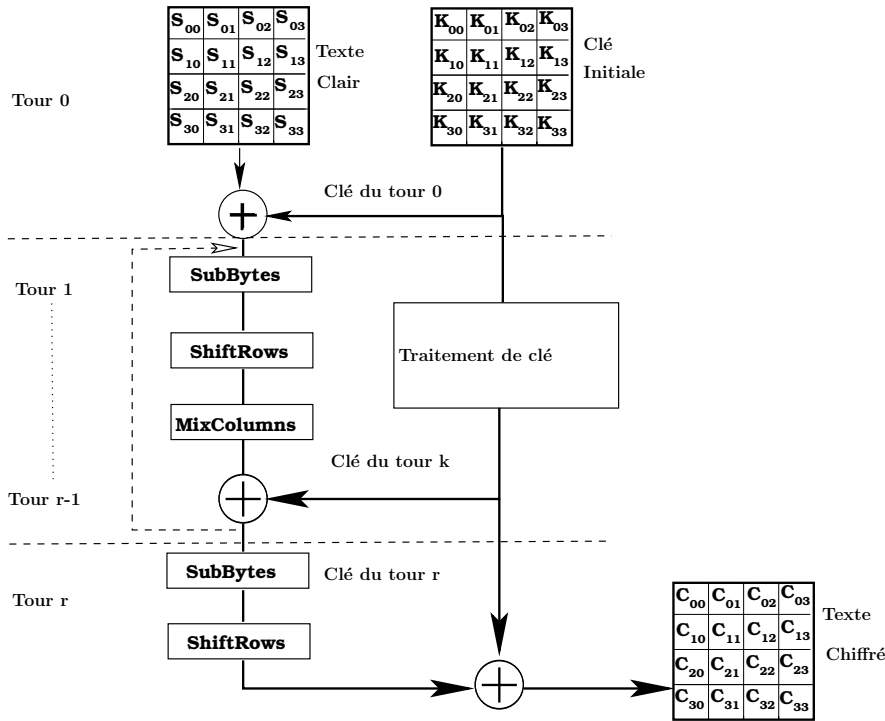


FIG. 1.5 – chiffrement AES

chiffrement. Après notre brève introduction de ces deux standards bien connus, et très utilisés, nous nous penchons sur les techniques cryptanalytiques, dont les plus importantes sont :

1.4 Les attaques algorithmiques

Certaines propriétés observées dans les algorithmes de chiffrement permettent de déceler des faiblesses dans la conception, qui peuvent mener à des attaques de différents types :

L'attaque à chiffré seul

L'adversaire n'a à sa disposition que le trafic chiffré, qu'il essaie d'utiliser pour retrouver soit le texte clair, soit la clé.

L'attaque à clair connu

L'adversaire possède plusieurs chiffrés et les messages clairs correspondants. Il doit retrouver la ou les clés qui ont été utilisées pour le chiffrement. L'attaque linéaire fait partie de cette catégorie et a été présentée par Mitsuru Matsui en 1993. Elle consiste en une approximation linéaire de la structure interne du chiffrement. Cette attaque est très efficace sur le standard de chiffrement DES.

L'attaque à clair choisi

L'attaquant peut choisir un nombre arbitraire de textes clairs et obtenir les chiffrés correspondants. Cette attaque est plus efficace que l'attaque à clair connu, car l'attaquant peut choisir des clairs spécifiques, qui donneront plus d'informations sur la clé. L'attaque différentielle, qui fait une analyse statistique des changements dans la structure du chiffrement après une modification constante sur les entrées, fait partie de cette catégorie. Cette attaque a été conçue et présentée à la conférence Crypto 1990 par Eli Biham et Adi Shamir. Des algorithmes comme AES sont spécialement conçus pour résister à ce type d'attaque.

L'attaque à chiffré choisi

L'adversaire choisit différents chiffrés et dispose en plus d'une boîte noire qui lui permet de déchiffrer.

1.5 Les attaques physiques

On appelle attaque physique, toute attaque menée non contre la structure mathématique d'un algorithme mais plutôt contre un ou plusieurs composants, en général électroniques, du dispositif physique mettant en œuvre l'algorithme.

Elle vise à exploiter des défauts du système de mise en œuvre. Ce genre d'attaque s'est développé de plus en plus pour devenir aujourd'hui l'une des composantes essentielles de la quête de renseignements.

En effet, une sécurité "mathématique" ne garantit pas forcément une sécurité physique lors de l'utilisation du chiffrement.

Ces attaques communément appelées attaques par canaux cachés¹², peuvent être invasives, par l'isolation du composant électronique et la mise à nu du circuit physiquement, ou non-invasives, par la mesure de paramètres physiques extérieurs au matériel pendant son activité, sans toucher à son intégrité physique. Ces attaques se basent généralement sur une connaissance approfondie de l'architecture du matériel et portent sur différents types de données :

Attaque temporelle : elle est basée sur une comparaison du temps mis pour effectuer certaines opérations ;

Attaque par faute : elle consiste en l'introduction volontaire d'erreurs dans le système pour provoquer certains comportements révélateurs ;

Analyse de rayonnement électromagnétique : Elle est basée sur le fait qu'une particule chargée de grande énergie émet un rayonnement électromagnétique. Ce rayonnement est analysé pour déduire l'information secrète.

Analyse de consommation : qui constitue le cœur de cette thèse. Elle consiste à analyser la consommation en courant. Ce type d'attaque a plusieurs variantes, dont

¹²En anglais Side Channel Attacks ou SCAs

nous avons étudié les spécificités sur lesquelles nous présenterons quelques résultats originaux dans ce rapport.

1.5.1 Les attaques par analyse de consommation

Ces attaques consistent en général à faire des analyses statistiques sur un nombre de signaux fournis après une étape d'acquisition sur le circuit observé. Ce sont des méthodes qui permettent très rapidement de casser des cryptosystèmes, dont les faiblesses liées à la mise en œuvre ne sont pas habituellement prises en compte.

En effet, elles exploitent les fuites physiques du matériel et utilisent ces informations de différentes manières. On peut se demander ce que signifie la «fuite d'information par canaux cachés».

Ceci se reflète par exemple dans les failles induites par le passage de la description d'une fonction de l'algorithme d'un langage de haut niveau vers sa description en terme de portes logiques (on appelle ceci la synthèse logique).

Ainsi, les portes logiques telles qu'elles sont actuellement conçues "dissipent" de l'énergie, de façon corrélée avec la séquence de données traitées¹³.

On peut donc mesurer cette dissipation afin de déduire certaines informations sur le secret traité par le cryptoprocresseur. Ces méthodes sont utilisées en fonction des informations acquises et dépendent de l'algorithme cryptographique.

Le calcul de la dissipation se fait en étudiant de plus près le bilan énergétique d'un circuit. Comme pour tout système physique, on remarque que ces circuits consomment de l'énergie électrique et produisent de la chaleur.

Les études sur la dissipation [50, 7] sont la base des attaques sur les circuits. Les technologies utilisées aujourd'hui dans les dispositifs électroniques sont principalement du CMOS¹⁴. Dans cet environnement, une variable booléenne est associée à une équipotentielle, que l'on peut assimiler à un condensateur. Le changement d'état de cette variable correspond à la charge ou à la décharge de ce condensateur. Si U est la tension d'alimentation du système, pour une équipotentielle de capacité C portant une charge Q , la montée de tension entre 0 et U , correspond à une variation de la variable booléenne. Cette variation provoque une dissipation d'énergie par le circuit, au moment de faire passer la charge $Q = CU$. Cette énergie est égale à $\int_0^Q \frac{q}{C} dq = \frac{1}{2} \frac{Q^2}{C} = \frac{1}{2} CU^2$. Pour une dissipation en fonction du temps de calcul, on présente la moyenne de l'énergie dissipée en une seconde par :

$$P_d = f \times N \times \tau \times \frac{CU^2}{2} \quad (1.1)$$

où :

¹³La corrélation entre les valeurs secrètes et la dissipation d'énergie est une constatation expérimentale. Sauf cas particulier, elle ne peut pas être déduite des seules équations théoriques du système de chiffrement.

¹⁴Complementary metal oxide semi-conductor

- f est la fréquence d'horloge,
- N est le nombre d'équipotentiels qui varient (*c.-à-d.* de variables booléennes) et
- τ est la probabilité de transition pour une équipotentielle, pendant la durée d'une période d'horloge.

Par conséquent, le calcul exprimé au niveau bas, par des transitions d'équipotentiels, produit des fuites physiques dont l'énergie ou la consommation est donnée dans l'équation 1.1. L'une des premières attaques qui a pu exploiter cette fuite est l'analyse simple de consommation ou la (SPA)¹⁵ [38]. Grâce aux variations d'énergie, cette attaque peut identifier les moments importants du chiffrement. Lorsqu'une instruction est exécutée par le microprocesseur, au même moment une donnée est manipulée bit par bit. Or le profil du courant varie selon qu'on traite le bit "1" ou le bit "0". Il suffit donc d'identifier une période précise du déroulement de l'algorithme durant laquelle on exécute au moins une instruction qui manipule les données bit par bit. Par exemple, le moment d'exécution du « *key schedule* » de la clé du DES, peut être clairement reconnu, en devinant les décalages et les itérations. Ceci se fait quand le bruit dans le canal caché n'est pas très grand. Dans ce sens, l'information doit être claire, pour pouvoir trouver la clé en interprétant la consommation du courant durant les calculs cryptographiques. Quand ces conditions ne sont pas remplies, la SPA n'est pas utilisable. Elle est alors abandonnée au profit d'autres techniques telles que la DPA1.5.2.

1.5.2 L'analyse différentielle de consommation, ou DPA

Le principe de l'attaque DPA¹⁶ repose sur le fait que la consommation en courant du processeur exécutant des instructions dépend de la donnée exécutée. Cette attaque a été largement étudiée durant la dernière décennie, et a fait l'objet de différentes analyses scientifiques largement publiées [38, 51, 27, 28, 61, 10, 81, 46, 52, 37, 48]. Elle utilise une analyse statistique sur un grand nombre d'échantillons qui permet de réduire le bruit en calculant les moyennes.

En général, l'adversaire commence par :

- identifier une *partie gérable* de la clé. L'analyse statistique devra être faite pour toute valeur que peut prendre cette *partie gérable* ;
- pour chaque opération de chiffrement i , enregistrer la consommation instantanée du dispositif sous forme d'un vecteur $V_i = (V_i^1, V_i^2, \dots, V_i^L)$, où L est le nombre d'échantillons enregistrés pour chaque opération.

L'attaque se déroule généralement en deux phases :

- l'acquisition des données et
- l'exploitation de ces données.

¹⁵Simple Power Analysis

¹⁶Differential Power Analysis

1.5.2.1 Acquisition des données

La DPA peut être soit une attaque à “chiffré seul” ou une attaque “à clair choisi”. Dans le premier cas, N textes chiffrés provenant de textes clairs inconnus sont enregistrés. Dans le second cas, N textes clairs sont choisis aléatoirement et leur chiffrement est demandé.

Dans les deux cas, il faut enregistrer la consommation électrique du dispositif pendant les opérations de chiffrement. On obtient ainsi un ensemble $\mathcal{C}$, de N couples (M_i, V_i) , où M_i est un texte clair ou un texte chiffré et V_i est un vecteur de $\mathbb{R}^L$, où L est le nombre d'échantillons recueillis lors de chaque opération.

1.5.2.2 Exploitation de ces données

L'exploitation des N couples (M_i, V_i) commence par l'identification d'une *partie gérable* de la clé. L'analyse statistique décrite ci-dessous est faite indépendamment pour chaque valeur possible de cette *partie gérable* :

1. Choisir une valeur booléenne interne au chiffrement, dont la valeur peut être calculée en fonction de M_i et d'une hypothèse sur la *partie gérable*.
2. Partitionner $\mathcal{C}$ en deux sous-ensembles $\mathcal{C}_0$ et $\mathcal{C}_1$, suivant la valeur recalculée de la variable booléenne choisie.
3. Calculer

$$\text{DPA}_0 = \frac{1}{|\mathcal{C}_0|} \sum_{i \in \mathcal{C}_0} V_i \quad \text{et} \quad \text{DPA}_1 = \frac{1}{|\mathcal{C}_1|} \sum_{i \in \mathcal{C}_1} V_i$$

4. et

$$\text{DPA} = \text{DPA}_1 - \text{DPA}_0$$

En effet si l'hypothèse de clé est fausse, alors chaque ensemble contient en moyenne autant de courbes correspondants à la manipulation d'un "0" que de courbes correspondants à un "1". Les deux ensembles sont pratiquement équivalents en terme de consommation en courant et donc le signal DPA sera asymptotiquement égal à zéro. Si le cas où l'hypothèse de clé est correcte, $\mathcal{C}_0$ contiendra effectivement les courbes correspondants à la manipulation d'un "0" et $\mathcal{C}_1$ à ceux correspondants à "1". La valeur prise par le signal DPA à l'instant (inconnu mais supposé constant) où le bit choisi prend sa valeur correspond à la différence de consommation du courant par le microprocesseur selon qu'il manipule un "1" ou un "0". La courbe (x, DPA_x) présente alors un pic qui traduit la validité de l'hypothèse faite sur la *partie gérable*.

Et ainsi de proche en proche, il est possible de découvrir toute la clé secrète.

Classiquement l'analyse en courant tente d'exploiter une dépendance avec le poids de Hamming d'une variable sensible. Ceci est appelé le “modèle du poids de Hamming”, qui consiste à calculer le nombre de bits dans un mot de données mis à '1'. Mais depuis 1998 et l'introduction de la CPA¹⁷ [9] 1.5.3, par Brier *et al*, le “modèle de la distance

¹⁷en anglais Correlation Power Analysis

de Hamming” est plus largement utilisé. C’est une attaque qui a été étudiée, et utilisée largement dans le domaine de la cryptanalyse physique, et a fait l’objet de nombreuses améliorations [42, 44, 5, 32, 39, 70].

Elle se base sur la supposition que les fuites de données à travers un canal auxiliaire dépend du nombre de bits de commutation d’un état à un autre à un moment donné.

1.5.3 L’analyse du courant par corrélation, ou CPA

Le raisonnement par corrélation est fondé sur la dépendance entre la consommation en courant du circuit et la distance de Hamming des données manipulées. L’attaque CPA part de la la supposition que les fuites de données à travers un canal auxiliaire dépend du nombre de bits de commutation d’un état à un autre à un moment donné. Selon ce modèle, la relation entre la consommation en courant et la distance de Hamming est linéaire, et par conséquent la clé correcte est celle qui maximise les corrélations entre elles. Le plus souvent, pour une mesure de consommation W et une distance de Hamming H entre deux états liés à l’information, c’est le facteur de corrélation de Pearson $\rho(W, H)$ qui est utilisé :

$$\rho(W, H) \doteq \frac{\text{cov}(W, H)}{\sigma(W) \cdot \sigma(H)},$$

Pour obtenir le modèle de la distance de Hamming, des hypothèses sont faites sur la “partie gérable” de la clé. Nous remarquons que $\rho(W, H)$ suit l’inégalité de Cauchy-Schwarz : $-1 \leq \rho(W, H) \leq +1$. La clé correcte est obtenue lorsque ce facteur tend vers ± 1 pour la bonne hypothèse de clé.

Pour ces valeurs, les variables sont parfaitement linéairement liées par une relation de croissance ou de décroissance selon le signe. Une valeur de 0 indique que les variables ne sont pas linéairement liées entre elles. On peut considérer qu’on a une forte corrélation si le coefficient de corrélation est supérieure à 0,8.

1.6 Une étude théorique

L’étude des attaques physiques a été initiée pour la première fois par Micali et Reyzin dans [55]. En effet une étude théorique était indispensable pour pouvoir concevoir des contre-mesures efficaces et éviter les astuces ou les manipulations qui ne font que retarder les échéances tout en risquant de fragiliser encore plus les circuits par de nouvelles méthodes plus élaborées, sans parfois, vraiment les renforcer face aux attaques déjà connues.

En fait, le but de ces études est d’arriver à créer des distingueurs qui permettent d’affirmer si un circuit est faible ou non contre un adversaire [78].

Dans cette section, nous faisons le point sur les différents cadres théoriques des attaques physiques, ainsi que les différents modèles qu’on peut construire.

1.6.1 Adversaire

On peut envisager de le classer en deux principaux types :

- Un adversaire qui a accès à certaines informations comme la consommation en courant du circuit, ainsi que l'algorithme utilisé. Il peut utiliser l'algorithme en lui administrant des données choisies pour analyser ses réponses.
- Un autre adversaire qui a accès à un clone du matériel attaqué auquel il peut fournir des données et des clés de son choix pour *s'entraîner* et ainsi être prêt pour une attaque « en ligne ».

Dans les deux cas, son but consiste à retrouver des informations secrètes manipulées par le circuit grâce à l'étude de la consommation du circuit.

Dans [55], l'adversaire peut être différent d'une attaque à l'autre mais il est choisi de façon à être le plus fort possible.

Dans [78], l'adversaire utilise une stratégie étendue-réduite pour retrouver séparément des parties de la clé secrète. En d'autres termes, l'adversaire définit une fonction $f : K \rightarrow S$ qui associe à chaque clé k une classe s_g telle que $|S| \ll |K|$.

Aussi, nous utiliserons les définitions suivantes, toujours selon [55] :

1.6.2 Définitions

Machine de Turing abstraite à mémoire virtuelle : en abrégé « AVTM¹⁸ », c'est une machine de Turing probabiliste ayant accès à un vecteur illimité de bits, numérotés à partir de 1.

Ordinateur abstrait : c'est une collection $\mathcal{A} = A_1, A_2, \dots, A_n$ d'AVTM. Les données d'entrée de $\mathcal{A}$ sont, par définition égales à celles de A_1 , dite « AVTM distinguée ».

Machine de Turing physique à mémoire virtuelle : en abrégé « PVTM¹⁹ », c'est un couple $P = (A, L)$ où A est une AVTM et L une *fonction de fuite*, qui sera décrite plus loin.

Ordinateur physique : c'est une collection de PVTM. Si $A = A_1, \dots, A_n$ est un ordinateur abstrait et $P_i = (L_i, A_i)$, on peut voir P_i comme une réalisation physique de A_i et, en général, $P = P_1, \dots, P_n$ une réalisation physique de A .

Ainsi, en supposant $E_k(.) = E_k$ où $k \in K$ une famille d'AVTM indexés par k , qui représente la clé, et (L, E_K) un ordinateur physique qui associe à E_k une fonction de fuite L , l'adversaire peut être vu comme un algorithme $Adv_{E_K, L}$, de complexité en temps τ , de complexité en mémoire m . Nous pouvons fixer à q le nombre d'expérimentations de l'adversaire sur l'ordinateur physique. Le but de l'adversaire est de deviner la classe de clé $s_g = f(k)$ avec une bonne probabilité.

¹⁸Acronyme du terme : *Abstract Virtual-Memory Turing Machine* utilisé dans [55].

¹⁹Acronyme du terme : *Physical Virtual-Memory Turing Machine* utilisé dans [55].

D'après les définitions données, l'évaluation de la sécurité physique d'un dispositif sera, par conséquent, basée sur la qualité de l'ordinateur physique et la force de l'adversaire. Ces deux aspects poussent à étudier la quantité d'information donnée par une fonction de fuite, ainsi que la façon dont cette information peut être utilisée pour réussir une attaque.

L'analyse théorique des attaques physiques dans [55] tente aussi de formaliser les fuites de calculs en se basant sur les axiomes suivants :

1. Les calculs et seulement les calculs provoquent des fuites d'information. Il n'y a pas de fuites tant que l'information secrète n'est pas utilisée pendant des calculs.
2. Les mêmes calculs provoquent des fuites d'information différentes sur différents calculateurs. Les mises en œuvre d'un algorithme peuvent être différentes d'un circuit à l'autre. En d'autres termes : les mêmes opérations élémentaires peuvent donner lieu à des fuites d'information dépendant de la plate-forme.
3. Les fuites d'information dépendent des mesures choisies. La quantité d'information, capturée par un adversaire pendant une attaque par analyse de consommation du procédé de mesures, peut être corrompu par du bruit par exemple.
4. La fuite d'information est locale. La quantité maximale d'information qui peut être extraite d'un circuit est la même pour toute exécution de l'algorithme avec les mêmes données manipulées.
5. Toutes les fuites d'information peuvent être calculées à partir de la configuration interne du circuit cible. Étant donné un circuit, la fuite d'information d'une PVTM $P = (L, A)$ est une fonction polynomiale de :
 - (a) la configuration interne du circuit C ,
 - (b) du type de mesures choisies M et,
 - (c) de certains aléas et bruits pendant le calcul.

D'un point de vue pratique, ces axiomes ne reflètent pas tous les phénomènes physiques qui peuvent être observés.

Par exemple, les quantités d'énergie requises par une mémoire dynamique pour son rafraîchissement périodique peuvent faire l'objet d'une attaque physique. Ceci peut refléter une contradiction avec l'axiome 1. Cependant ces « petites fuites » sont plus difficiles à exploiter par rapport à celles produites par les calculs.

1.7 Problématique

Les expérimentations sur les attaques physiques et notamment les attaques *template* que nous allons présenter dans le chapitre 3, ne sont pas évidentes dans le cas des traces réelles qui sont acquises soit sur des circuits dédiés, soit sur des circuits programmables. Nos premiers tests ont été un échec, et ont montré qu'il y a une grande diversité entre les propositions théoriques et les applications pratiques.

Ce cas peut être problématique pour un évaluateur en charge d'apprécier la robustesse des circuits face à ce genre d'attaques. Il peut facilement donner un mauvais jugement s'il ne tient pas compte de tous les éléments techniques.

En effet, plusieurs autres facteurs rentrent en jeu :

- le bruit dans les signaux,
- le choix des modèles,
- les erreurs d'interprétations,
- le choix des points d'intérêt,
- les éléments de comparaisons....

Dans le cadre de ces travaux de thèse, nous menons des études expérimentales approfondies afin de sensibiliser les évaluateurs sur les mauvaises déductions par rapport à la robustesse des circuits. Nous analysons les entraves et proposons des améliorations.

1.8 Plan du document

Outre cette introduction et la conclusion finale, ce document est composé de 8 chapitres.

Dans le deuxième chapitre, nous commençons par présenter le cadre expérimental de nos attaques. Ensuite, nous décrivons notre plateforme et l'architecture itérative du circuit analysé. Enfin, nous définissons les métriques utilisées pour comparer les attaques et mettre en évidence nos améliorations.

Le troisième chapitre est consacré à la description des attaques *template*. Nous détaillons les procédures de profilage et d'attaque en ligne et parlons de la sélection des fuites ciblée grâce au choix des points d'intérêt, et notamment par l'analyse par composantes principales.

Dans le quatrième chapitre, nous présentons la différence entre une attaque aveugle sans choix de modèle et une attaque basée sur un modèle de consommation. Nous faisons une attaque *template* sur le *key schedule* du standard DES, et montrons comment étaler plus de fuite d'information grâce à de nouvelles fonctions de sélections basées sur l'algorithme de chiffrement.

Dans le cinquième chapitre, nous faisons une comparaison entre cinq modèles de fuite, et découvrons qu'une attaque *template* peut réussir même sans la connaissance de l'architecture interne du circuit. Nous discutons aussi de la pertinence du cadre théorique sur les attaques présenté par F.-X. Standaert et al. à Eurocrypt 2009, et donnons une méthode basée sur le seuillage pour accélérer le profilage et l'attaque en ligne.

Le chapitre six est l'occasion de parler de la combinaison de différents partitionnements sur les attaques *template*. Nous détaillons les difficultés basées sur le nombre de traces dans le profilage, et présentons l'apport des vecteurs propres de l'analyse par composantes principales dans la métrique de robustesse du circuit. En effet, cette dernière a besoin de tous les vecteurs propres par rapport à la métrique qui estime la force de

l'adversaire où nous avons besoin d'un compromis sur les vecteurs propres. Nous présenterons aussi différentes méthodes de choix des points d'intérêts dans le cadre de traces électromagnétiques. L'attaque sera nettement améliorée grâce à une combinaison de ces points.

Dans le chapitre 7, nous traitons le cas où il y a une divergence entre les *templates* du profilage et les traces attaquées. Nous parlons de la désynchronisation des traces, et du redimensionnement des traces en amplitudes. Pour éviter ce type d'obstacles, nous proposons deux démarches basées sur le réajustement et la normalisation des campagnes d'acquisitions.

Le huitième chapitre est dédié à quelques techniques du traitement du signal dans le cadre des attaques par canaux auxiliaires. Nous faisons une comparaison d'attaques, où nous discutons de l'apport d'un passage dans le domaine fréquentiel, par rapport au domaine temporel, et montrons qu'on peut combiner les deux grâce à la transformée en ondelettes. En effet, une attaque basée sur les coefficients de cette transformée, est très efficace dans le cas de traces bruitées, car elle permet de passer d'un taux de succès de 0% à 100%.

Enfin, le neuvième chapitre est l'occasion d'exposer mes résultats au concours international Dpacontest auquel j'ai participé avec une attaque *template*. Je présente aussi quelques remarques sur des fuites constatées dans les signaux et qui sont loins des instants de chiffrement. Je montrerai comment étudier ce fuites et les utiliser dans une attaque *template*.

Chapitre 2

Configuration, mises en œuvres et métriques

Les attaques physiques se font nécessairement sur des cryptoprocresseurs. Pour bien comprendre ces attaques, il faut savoir qu'un adversaire doit connaître un minimum de choses sur le fonctionnement du circuit cible dédié aux fonctions cryptographiques. La connaissance de l'algorithme cryptographique fait bien évidemment partie de ce minimum. Ces co-procresseurs spécialisés permettent de réaliser des opérations ou des tâches cryptographiques (Signature numérique, chiffrement DES ou AES, etc.) sur des dispositifs.

Dans le cahier des charges de sa conception, ce type de processeur doit gérer beaucoup de contraintes, telles que la gestion des clés, ou même la résistance aux différentes attaques physiques ou algorithmique.

La gestion des données se fait sur une mémoire supposée fragile car c'est une cible privilégiée pour les adversaires. Pour ceci, les données sont normalement chiffrées avec une clé de session spécifique au programme, et sont stockées en mémoire. Le déchiffrement se fera à la demande du processus cryptographique lui même. De ce fait seul le processeur légitime peut les lire.

Il existe aussi des algorithmes qui interdisent les modifications du contenu de la mémoire par un espion, et les valeurs des registres peuvent être effacées pour ne pas divulguer d'informations sensibles.

Afin d'illustrer par des exemples le travail développé dans cette thèse, nous avons mené des attaques réelles contre un cryptoprocresseur capable de mettre en œuvre indifféremment les algorithmes DES, triple-DES ou AES.

2.1 Les cryptoprocresseurs étudiés

Nos attaques ont été mises en pratique en utilisant le circuit, SECMAT, conçu au laboratoire de l'Institut Telecom, dans le cadre du CMP [15, p. 62 à 63]. Ce dernier est un

prototype de circuit académique à $0.13 \mu\text{m}$, conçu pour l'évaluation des attaques cryptographiques. Il comporte, outre une unité centrale compatible avec le microprocesseur 6502, une ROM d'initialisation, 32 kilobytes de RAM et un UART¹, deux coprocesseurs cryptographiques capables de mettre en œuvre respectivement DES et AES.

2.1.1 Architecture SECMAT

Dans le “system-on-chip” SecMat, les co-processeurs cryptographiques ont leurs propres lignes d'alimentation électrique, ce qui nous permet de capturer leur consommation, sans que la consommation du reste du système ne perturbe nos mesures. Cette situation est favorable à l'attaque.

L'architecture du coprocesseur DES est décrite plus en détail dans [29]. Nous la résumons ci-dessous :

- un registre de 64 bits (“IF”) est utilisé pour le chargement/déchargement des opérateurs à partir de/vers la mémoire de 8 bits.
- un registre de 64 bits (“ LR”) contenant les messages de tours.
- un registre de 56-bit (“ CD”) contenant les clés de tour.

Le chiffrement est effectué de manière itérative, parallèlement à la génération des clés de tour à partir de la clé globale. Le contrôle est fait par la machine d'état suivante, indépendante des données :

Cycle d'horloges $1 \mapsto 7$: K_1 à K_7 chargés dans IF, octet par octet.
 Cycle d'horloges 8 : K_8 chargés dans IF ; $CD := IF$.
 Cycle d'horloge $8 \mapsto 15$: M_1 to M_7 chargés dans IF, octet par octet.
 Cycle d'horloge 16 : M_8 chargés dans IF ; $LR := IF$.
 Cycle d'horloge $16 \mapsto 31$: tours 1 à 15 de DES.
 Cycle d'horloge 32 : Tour 16 du DES ; $IF := LR$.
 Cycle d'horloge $32 \mapsto 40$: (IF) est écrit dans la mémoire, octet par octet.

L'activité de la clé pendant les cycles d'horloge est résumée comme suit :

Cycle d'horloge $1 \mapsto 8$: La clé est chargée dans IF octet par octet.
 Cycle d'horloge $8 \mapsto 16$: La clé est progressivement effacée dans IF par le message entrant.
 Cycle d'horloge $16 \mapsto 32$: La génération des clés (les transferts LS ou LS² dans le CD) est activée.
 Cycle d'horloge $32 \mapsto 40$: La clé est intacte pendant cette phase, donc pas d'activité.

La figure 2.1.1 donne une idée sur le chemin de données du DES. Toutes les expérimentations sur DES proposées dans cette thèse sont issues de cette mise en œuvre.

¹Universal Asynchronous Receiver Transmitter

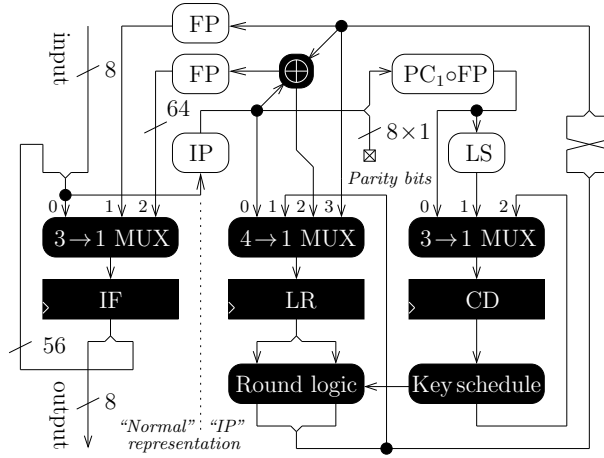


FIG. 2.1 – Chemin de données du DES utilisé dans ce rapport

2.1.2 Les acquisitions

Une campagne d’acquisitions consiste en l’enregistrement de traces de consommation pour différents couples {clé, message}. Comme décrit dans [27], l’appareil d’acquisition est un oscilloscope INFINIUM, 54,855A d’AGILENT. Le modèle de sonde est un 1132A, avec une bande passante de 5 GHz. Le kit de connectivité différentielle E2669A a été utilisé. Les traces de consommation montrées dans cette thèse ont été acquises avec un connecteur en soudure. Chaque trace est moyennée 64 fois par l’oscilloscope pour filtrer le bruit d’environnement et augmenter la résolution verticale de 8 à 12 bits. La figure 2.2 résume en images le processus d’acquisitions.

2.2 Métriques de comparaison

Tout au long de ce rapport, nous allons réaliser des comparaisons entre composants et éléments des attaques physiques. Toute la question est de savoir avec quels paramètres comparer. En effet, il y a beaucoup de variables qui entrent en jeu, et il faut, parfois, faire des compromis, et isoler certains éléments protagonistes faisant partie des attaques/analyses. Une métrique classique de comparaison des signaux est le rapport signal-bruit, noté SNR^2 dans la littérature anglo-saxonne. Ce dernier représente le rapport entre l’information pertinente et l’information non significative. Ceci permet de quantifier une fuite.

Mais tout d’abord, nous devons bien faire la différence entre deux éléments importants dans les attaques :

- la quantité de la fuite d’information et

²Signal-Noise Ratio

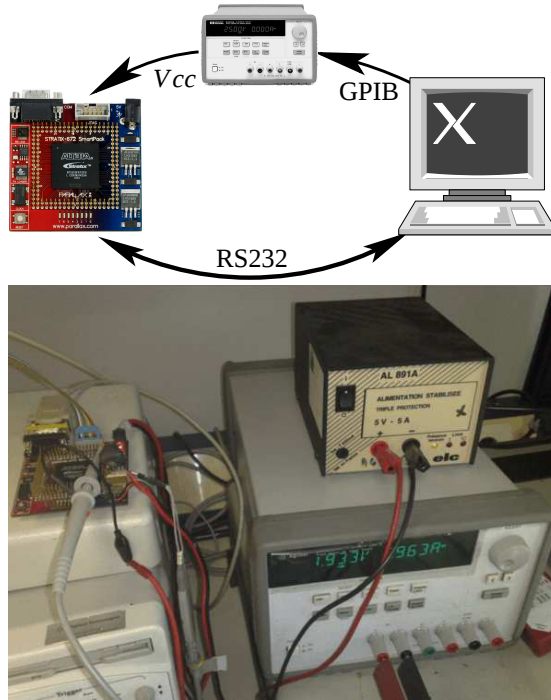


FIG. 2.2 – Plateforme des attaques

- la force de l'attaquant.

Grâce à la quantité de fuite d'information nous pouvons comparer deux circuits face à un attaquant et la force d'un adversaire peut être estimée par sa réussite face à une fuite de données.

2.2.1 L'entropie pour quantifier l'information

Le concept d'entropie a été introduit par C. Shannon dans [72]. Il s'agit d'une mesure fiable de l'incertitude associée à une variable aléatoire. Dans notre cas, nous utilisons l'entropie de Shannon pour caractériser la fuite d'information à partir d'un dispositif cryptographique. Nous désignons par :

- S_K la variable aléatoire discrète représentant la clé cible et s_K une réalisation de cette variable ;
- X la variable aléatoire discrète représentant les données entrées au dispositif ciblé et x une réalisation de cette variable ;
- L la variable aléatoire correspondant aux observations générées par les données entrées au dispositif, et l une réalisation de cette variable ;
- $\Pr(s_K \mid \mathbf{L})$ la probabilité conditionnelle d'avoir une classe de clé s_K sachant la fuite l .

L'une des raisons pour la quantification de l'information est de mesurer la qualité du circuit face à une fonction de fuite donnée. Pour ceci, nous utiliserons pour nos comparaisons l'entropie conditionnelle, définie dans la section 2.2.1.1. Notre autre objectif est de mesurer comment cette information (la fuite) est utilisée pour réussir à retrouver la clé. Dans la suite, nous considérons le taux de succès ou l'entropie estimée, définis dans la section 2.2.1.2, pour évaluer la force de l'adversaire.

Ces deux indicateurs nous permettent ensuite de garantir la sécurité du circuit (en sachant sa fuite) contre un adversaire plus ou moins fort.

2.2.1.1 L'entropie conditionnelle

D'après la définition de l'entropie conditionnelle de Shannon, l'incertitude conditionnelle de S_K sachant L , notée par $\mathbf{H}(S_K | L)$, est définie par les équations suivantes :

$$\begin{aligned} \mathbf{H}(S_K | \mathbf{L}) &\doteq \sum_{s_K} \sum_l -\Pr(s_K, l) \cdot \log_2 \Pr(s_K | l) \\ &= -\sum_{s_K} \Pr(s_K) \sum_l \Pr(l | s_K) \cdot \log_2 \Pr(s_K | l). \end{aligned} \quad (2.1)$$

Nous définissons la matrice d'entropie conditionnelle par :

$$\mathbf{H}_{s_K, s_{Kc}} \doteq -\sum_l \Pr(l | s_K) \cdot \log_2 \Pr(s_{Kc} | l), \quad (2.2)$$

où s_K et s_{Kc} sont respectivement la clé correcte et la clé candidate.

D'après les équations (2.1) et (2.2), nous déduisons :

$$\mathbf{H}(S_K | \mathbf{L}) = \sum_{s_K} \Pr(s_K) \mathbf{H}_{s_K, s_K}. \quad (2.3)$$

Les valeurs des éléments de la diagonale de cette matrice doivent être observées avec attention. En effet, le théorème 1 dans [77] stipule que si elles sont minimales parmi toutes les autres classes de clés s_K , alors ces classes de clés peuvent être récupérées par un adversaire bayésien.

2.2.1.2 Définition du taux de succès et l'entropie estimée

L'adversaire mentionné dans 1.6.1 est un algorithme qui vise à deviner une classe de clé s_K avec une grande probabilité.

Taux de succès : Il est estimé par un adversaire, en calculant le nombre de fois que l'attaque est réussie face à des données différentes (messages ..). Ainsi, le taux de succès quantifie la force d'un adversaire et évalue ensuite la robustesse d'un

dispositif cryptographique face à son attaque. On peut aussi parler de taux de succès d'ordre supérieur, qui se base sur la tolérance qu'on peut donner à un attaquant dans le classement des clés. Un autre élément important est

l'entropie estimée ³ : elle est définie comme la moyenne du *classement* de la clé pendant les attaques. En effet, les SCAs se basent généralement sur des suppositions exhaustives sur les clés, et dont le but est de *classer* la clé en premier parmi les autres. C'est une métrique pertinente car elle permet de voir au fur et à mesure le comportement de l'attaquant et permet de savoir si la clé recherchée remonte dans le classement et si elle chute face à des traces bruitées.

³En anglais Guessing Entropy

Chapitre 3

Les attaques *template*

3.1 Schéma d'attaque

L'attaquant dispose d'un clone du dispositif attaqué, à la différence que ce clone ne contient pas de valeur secrète, dont il a la maîtrise totale. L'attaque procède en deux phases :

phase d'entraînement :

L'attaquant choisit une *partie gérable* de la clé et, pour chaque valeur possible K_i de cette *partie gérable*, il chiffre un grand nombre de messages aléatoires, tout en enregistrant la consommation électrique du dispositif durant les chiffrements. Le terme *partie gérable* de la clé vient de que l'analyse doit être faite pour chaque valeur de cette sous-clé, à la manière d'une attaque exhaustive.

Lorsque ces données sont recueillies, un traitement statistique, que nous détaillerons dans la suite, permet d'identifier des caractéristiques de consommation dépendant de la valeur de K_i .

phase d'attaque en ligne :

L'attaque proprement dite consiste à chiffrer quelques messages à l'aide du dispositif contenant une clé secrète K , en enregistrant bien sûr la consommation électrique. Lorsque ces données sont enregistrées, les caractéristiques dépendantes de la clé dans les traces provenant du clone sont utilisées pour déterminer à quel groupe de traces issues du clone les traces issues du dispositif "ressemblent" le plus. L'attaque permet ainsi de déterminer une partie de la clé secrète K . Du fait que l'attaquant a la maîtrise totale du dispositif clone, il peut recueillir autant d'information que nécessaire pour pouvoir reconnaître des ensembles de sous-clés qui recouvrent la totalité de K .

Compte tenu du modèle puissant de l'attaquant, ces attaques sont considérées comme les plus puissantes formes d'attaques par canaux cachés publiées à ce jour. Elles permettent de casser des mises en œuvres même si elles sont protégées. On part aussi de la supposition que l'attaquant ne peut obtenir qu'un nombre très réduit d'échantillons du canal auxiliaire cible.

À cet effet, et comme l'adversaire dispose d'un matériel programmable d'apprentissage, il pourra obtenir ce que l'on appellera tout au long de ce rapport les *templates*.

Cette première étape est effectuée sur le dispositif clone, avec un ensemble réduit de possibilités sur la variable sensible exécutée. par exemple, un adversaire peut exécuter le même code pour différentes valeurs de bits de la clé, et construire ainsi O_{tpl} *templates*.

Après, l'adversaire utilise ces *templates* pour classifier une trace t_K (voir la figure 3.1 qui représente une trace extraite durant un chiffrement DES) prise sur le dispositif ciblé en limitant le choix des bits de la clé K . Ainsi, il peut déduire quelle valeur/opération est exécutée.

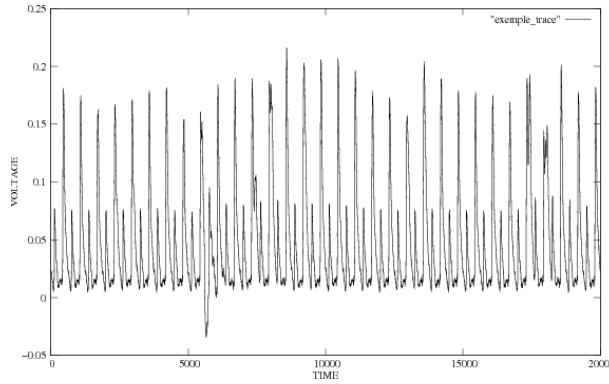


FIG. 3.1 – Un exemple d'une trace capturée par l'oscilloscope représentant un chiffrement DES.

Enfin, il utilise d'une manière incrémentale des préfixes de plus en plus long de K car l'espace des clés est énorme et la construction des *templates* pour chaque clé est impossible.

L'utilisation de l'approche du maximum de vraisemblance [33] permettra de sélectionner l'opération pour laquelle la probabilité est maximale. Dans la section suivante nous donnerons des méthodes d'estimation des variables aléatoires correspondant aux échantillons acquis. Nous étudierons la modélisation de la dissipation d'information par des lois de probabilité, et regarderons quel modèle statistique pour la décision dans le cas d'une attaque *template*.

3.2 Un modèle probabiliste

L'analyse probabiliste et le choix de travailler avec une classification nécessitent un modèle statistique utilisable dans la théorie de la décision. Cette analyse repose sur la règle de Bayes qui permet de tester des probabilités *a posteriori* (après l'observation concrète de certaines entités), connaissant les distributions de probabilité conditionnelles *a priori* (indépendamment de n'importe quelle contrainte sur les variables observées).

Ainsi, définir un modèle probabiliste pour un échantillon, c'est décider que les données pourraient être issues de la simulation d'une certaine loi de probabilité.

Le postulat de modélisation de base de toute étude statistique est que les données observées sont des réalisations de variables aléatoires. Quand les résultats d'expériences ne sont pas exactement reproductibles, nous supposons qu'ils sont les réalisations de variables aléatoires. Dans ces cas, la théorie des probabilités apporte un ensemble de lois (les grands nombres, ..), permettant d'extraire les données donc de fonder une prédiction ou une décision.

Quand on recueille des données, comme par exemple lors de l'acquisition de traces de consommation, on sait bien qu'un autre échantillon recueilli dans des conditions similaires, serait différent du premier, à cause du bruit et des conditions dans lesquelles cette expérience est effectuée. Ce deuxième échantillon ressemblerait au premier au sens où sa moyenne, sa variance, ses quantiles, prendraient des valeurs proches. On travaillera donc avec des variables aléatoires, qui sont par définition, des entités réelles observées, en principe, par le sondage d'événements élémentaires.

Mathématiquement, on peut définir une correspondance entre l'espace des événements Ω et l'ensemble des réels R . En effet une variable aléatoire unidimensionnelle peut être vue comme une fonction $X : \Omega \longrightarrow R$ telle que chaque intervalle $]a, b[$ de R détermine un événement $X^{-1}(]a, b[)$. Les variables aléatoires unidimensionnelles ne suffisent pas à l'étude des phénomènes de classification. Dans nos études sur les signaux qui sont considérés comme des vecteurs de dimension d , il faut utiliser des variables aléatoires multidimensionnelles ou vectorielles de dimension $d : X : \Omega \longrightarrow R^d$ tel que $X = (X[1], X[2], \dots, X[d])$.

Il est clair que la représentation de chaque classe de données dépendra principalement de la loi de distribution statistique adoptée : un choix difficile et parfois arbitraire !

La loi normale est parfois adoptée sans justification autre que sa fonction de densité symétrique et décroissante, pour approximer un échantillon concentré autour d'un point. Elle peut se généraliser à plusieurs dimensions en remplaçant la moyenne (μ) par un vecteur et la variance (σ^2) par une matrice de covariance C .

Pour un vecteur X , nous avons deux situations possibles :

- Si la matrice de covariance C est singulière, alors les échantillons se trouvent dans un même hyperplan, ce qui réduit la dimension de l'espace. Cela détecte le fait que les composantes de la variable aléatoire X présentent une dépendance linéaire.
- Si la matrice de covariance C est définie positive, alors on peut définir une métrique basée sur une distance qui accorde un poids moins important aux composantes les plus bruitées.

Dans les deux cas, on dispose d'une interprétation géométrique de la loi de probabilité, qui peut en justifier l'utilisation.

Ainsi, pour modéliser l'échantillon des traces, on utilise la densité gaussienne multivariée, c'est-à-dire la distribution d'un vecteur dont les composantes sont des variables aléatoires gaussiennes.

Cette densité gaussienne s'exprime par :

$$p(x) = \frac{1}{\sqrt{(2\pi)^N |C|}} \exp\left(-\frac{1}{2}(x - \mu)^T C^{-1}(x - \mu)\right)$$

où à partir d'un nombre fini de réalisations x_k , le vecteur de moyenne μ et la matrice de covariances C sont estimés par :

$$\mu = \frac{1}{K} \sum_{k=1}^K x_k \quad (3.1)$$

$$C = \frac{1}{K-1} \sum_{k=1}^K (x_k - \mu)(x_k - \mu)^T \quad (3.2)$$

On peut écrire cette matrice de covariances autrement en définissant la matrice X par :

$$X = \begin{pmatrix} x_1^T - \mu^T \\ x_2^T - \mu^T \\ \vdots \\ x_N^T - \mu^T \end{pmatrix}$$

D'où la formule simplifiée de la matrice de covariance :

$$C = \frac{1}{K-1} X^T X \quad (3.3)$$

3.3 La vraisemblance

Soit $E = \{e_1, \dots, e_k\}$ un ensemble fini, $\{\mathcal{L}p_\theta\}$ une famille de lois de probabilité sur E , et n un entier. La vraisemblance est définie pour tout vecteur $(x_1, \dots, x_n)$ d'éléments de E et pour une valeur θ associée à la famille $\{\mathcal{L}p_\theta\}$, dans l'équation suivante :

$$L(x_1, \dots, x_n, \theta) = \prod_{i=1}^n \mathcal{L}p_\theta(x_i)$$

En considérant un échantillon théorique $(X_1, \dots, X_n)$ de la loi $\mathcal{L}p_\theta$, et en supposant que les variables aléatoires $X_1, \dots, X_n$ sont indépendantes et de même loi $\mathcal{L}p_\theta$, la probabilité que cet échantillon théorique ait pour réalisation l'échantillon observé $(x_1, \dots, x_n)$ est le produit des probabilités pour que X_i prenne la valeur x_i , à savoir :

$$\mathbf{P}[(X_1, \dots, X_n) = (x_1, \dots, x_n)] = L(x_1, \dots, x_n, \theta)$$

Estimer un paramètre d'une loi par la méthode du maximum de vraisemblance, c'est prendre comme valeur de ce paramètre celle qui maximise la vraisemblance, à savoir la probabilité d'observer les données comme réalisation d'un échantillon de la loi $\mathcal{L}p_\theta$. Supposons que pour toute valeur $(x_1, \dots, x_n)$, la fonction $\theta \mapsto L(x_1, \dots, x_n, \theta)$ admette un maximum unique, la valeur $\hat{\theta}$ pour laquelle ce maximum est atteint dépend de $(x_1, \dots, x_n)$:

$$\hat{\theta} = \tau(x_1, \dots, x_n) = \arg \max L(x_1, \dots, x_n, \theta)$$

Si $(X_1, \dots, X_n)$ est un échantillon possible de la loi $\mathcal{L}p_\theta$, la variable aléatoire :

$$T = \tau(X_1, \dots, X_n) ; ,$$

est l'estimateur du maximum de vraisemblance de θ .

Après cette introduction sur la décision probabiliste qui sera utilisée pour choisir la clé parmi les suppositions, nous montrons dans la section suivante comment réduire un vecteur d'échantillons en ne gardant que l'information pertinente.

3.4 L'attaque *template* avec recherche de points d'intérêt

3.4.1 La recherche de points d'intérêt

En pratique, le nombre d'échantillons par trace est très grand (de l'ordre de 10^5 dans nos expérimentations). Cette taille rend impossible la construction de la base de données (les *templates*), nécessaire à l'attaque :

- une unique matrice contiendrait 10^{10} valeurs, soit 8.000.000.000 octets pour 1 clé possible, sachant que, même pour un chiffrement aussi simple que DES, il y aurait 64 matrices ;
- le stockage des matrices sur disque allongerait de façon démesurée les durées de calcul ;
- le calcul de l'inverse d'une matrice carrée de côté n ayant une complexité en $O(n^3)$, le calcul nécessiterait $O(10^{15})$ opérations et
- les méthodes de calcul connues de matrices inverses ne sont pas capables de traiter de telles matrices pour des raisons de stabilité numérique.

Cette obstruction est supprimée en remarquant que tous les points de la courbe ne contiennent pas la même quantité d'information utile. Nous appellerons « **point d'intérêt** » un point apportant une quantité d'information significative. La sélection d'un « petit » nombre de ces « points d'intérêt » permet d'écarter les obstructions citées ci-dessus, liées à la taille des matrices de covariance.

Quelques techniques palliatives ont été élaborées pour éviter ce souci, comme le fait de chercher les points d'intérêt aux instants pendant les quels on observe de larges différences entre les traces moyennes de chaque *template* [12].

Une autre méthode similaire a été présentée par Recheberger et Oswald [65]. Elle consiste à faire la somme des carrés des différences entre les traces moyennes des *templates*

et choisir les points où les maxima de la courbe sont atteints sur des périodes d'horloges. la figure 3.2 donne un exemple de cette méthode. D'autres techniques seront détaillées plus amplement dans la section 6.2.

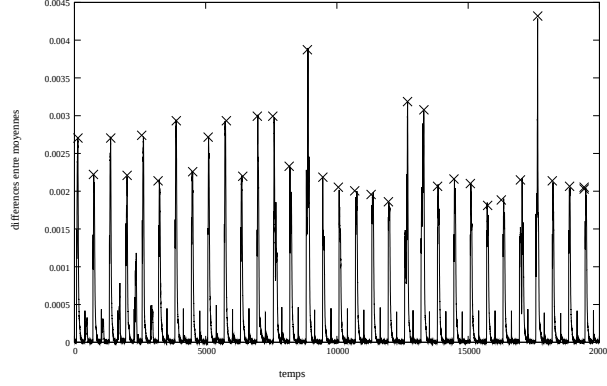


FIG. 3.2 – Les points d'intérêt.

3.4.2 Attaque *template* et points d'intérêt

Le principe de l'attaque *template* est inchangé : durant une phase d'apprentissage, l'attaquant détermine des caractéristiques liées à la clé, qu'il tente de reconnaître sur les traces issues du système-cible dans la phase d'attaque.

3.4.2.1 Phase d'apprentissage

La phase d'apprentissage suit les étapes ci-dessous :

1. l'attaquant recueille un grand nombre L d'échantillons sur le dispositif expérimental pour chacune des O_{tpl} opérations ;
2. il calcule les signaux moyens M_i pour chacune des O_{tpl} opérations ;
3. il calcule les carrés des différences entre tous les signaux moyens M_i pour sélectionner N points $P_1, \dots, P_N$, où l'on observe une *grande* différence, qui deviendront les *points d'intérêt* ;
4. pour chaque opération, on prend le vecteur de bruit pour un échantillon T , $B_i(T) = (T[P_1] - M_i[P_1], \dots, T[P_N] - M_i[P_N])$, et on calcule les matrices de covariances, définies par :

$$\sum_{B_i} [u, v] = \text{cov}(B_i(P_u), B_i(P_v)).$$

3.4.2.2 Phase d'attaque proprement dite

La seconde étape consiste à classer une trace S . Ceci se fait :

1. en calculant la probabilité d'observation de S , pour savoir si elle provient d'une opération déjà testée durant la première étape sur le dispositif clone. En fait nous calculons le vecteur de bruit n en utilisant le signal moyen M_i du *template* ;
2. en calculant la probabilité d'observation de n en utilisant l'expression 3.4.
3. et en classifiant avec utilisation de la méthode du maximum de vraisemblance.

$$p_{B_i}(\mathbf{n}) = \frac{1}{\sqrt{(2\Pi)^N |\Sigma_{B_i}|}} \exp\left(-\frac{1}{2} \mathbf{n}^T \Sigma_{B_i}^{-1} \mathbf{n}\right), \quad \mathbf{n} \in R^N \quad (3.4)$$

3.5 L'attaque *template* dans les sous-espaces principaux

La difficulté principale des attaques *template* est de gérer la grande quantité d'information ou de données, qui peut être vue comme une matrice X de dimensions (n, p) ,

contenant les observations : $X = \begin{pmatrix} x_1^1 & \dots & x_1^p \\ \vdots & \vdots & \vdots \\ x_i^1 & x_i^j & x_i^p \\ \vdots & \vdots & \vdots \\ x_n^1 & \dots & x_n^j \end{pmatrix}.$

où x_i^j est la valeur de l'échantillon i pour la variable j . Les lignes de cette matrice correspondent aux traces extraites du circuit.

Nous avons vu dans la section précédente, qu'on peut choisir des points d'intérêt pour réduire les dimensions, et ne garder que les instants de fuite. Toutefois une autre méthode, permet de réduire la dimensionnalité, en se basant sur les vecteurs propres d'une matrice de covariance. C'est l'analyse en composantes principales (ACP) [4]).

3.5.1 L'analyse en composantes principales

L'analyse en composantes principales (ACP) fait partie d'une famille de techniques statistiques (les méthodes multidimensionnelles) utilisées pour traiter des données provenant de situations où plusieurs variables sont mesurées simultanément. Lorsque plusieurs mesures continues sont observées ou mesurées sur un échantillon, il est rare que toutes les mesures prises soient indépendantes, c'est-à-dire qu'elles apportent une information complètement différente l'une de l'autre. L'analyse en composantes principales constitue un outil pour évaluer et représenter la redondance entre plusieurs mesures ou variables. Elle est souvent utilisée pour représenter graphiquement et de manière synthétique les faits saillants d'un ensemble de données. Grâce à cette méthode descriptive, les fichiers de données comprenant un grand nombre de variables (mesures) peuvent être analysés et résumés graphiquement avec la structure sous-jacente des données.

Elle permet l'analyse de la structure de la matrice de variance-covariance (la variabilité, dispersion des données).

Soit p le nombre minimal de variables nécessaires $(X_1, X_2, \dots, X_p)$ pour rendre compte de toute la variabilité du système. L'objectif de l'ACP est de déterminer $q < p$ variables $(C_1, C_2, \dots, C_k, \dots, C_q)$, combinaisons linéaires de X_i , rendant compte d'un maximum de cette variabilité. Il est clair que la quantité de variabilité représentée est croissante avec q . Il est donc nécessaire de faire un choix arbitraire soit de q , soit d'un minimum de variabilité devant être représentée. Généralement, on se limite à $q = 2$ ou $q = 3$, qui permettent une représentation graphique des données.

3.5.2 Recherche des composantes principales

Comme il vient d'être annoncé, la recherche des nouvelles composantes : $C_1, C_2, \dots, C_k, \dots, C_q$, consiste à trouver q combinaisons linéaires $(C_i, i = 1, 2, \dots, q)$ des $X_1, \dots, X_p$, avec des coefficients à déterminer et vérifiant les conditions suivantes :

$$C_k = a_{1k}X_1 + a_{2k}X_2 + \dots + a_{pk}X_p \quad (3.5)$$

où les a_{jk} sont des coefficients à déterminer. Les C_k doivent respecter les conditions suivantes :

- elles doivent être 2 à 2 non corrélées,
- de variance maximale,
- et d'importance décroissante.

la première composante principale C_1 est donc de variance maximale. Elle représente géométriquement une nouvelle direction dans le nuage de points qui suit l'axe d'allongement (étirement) maximal du nuage.

C_{i1} est la coordonnée du point i sur l'axe C_1 qui est la projection de x_i sur C_1 . Ces projections doivent être les plus dispersées possibles pour respecter la condition de variance maximale.

Le choix des autres composantes se fait d'une manière itérative telle que chaque nouvelle composante soit orthogonale aux autres et que la variance en soit maximale. En effet, la variance s'écrit de la manière suivante :

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^T (x_i - \mu) \quad (3.6)$$

On cherche à maximiser la projection de la variance sur les composantes. Ainsi, on considère l'opérateur de projection orthogonale, π , sur une droite de vecteur directeur unitaire V . Il s'écrit de la manière suivante :

$$\pi = VV^T, \text{ avec } V^TV = 1.$$

La variance des observations projetées s'écrit donc :

$$\begin{aligned}
\sigma_V^2 &= \frac{1}{n-1} \sum_{i=1}^n (\pi(x_i - \mu))^T (\pi(x_i - \mu)) \\
&= \frac{1}{n-1} \sum_{i=1}^n (VV^T(x_i - \mu))^T (VV^T(x_i - \mu)) \\
&= \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^T (VV^T VV^T) (x_i - \mu) \\
&= \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^T (VV^T VV^T) (x_i - \mu) \\
&= \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^T (VV^T) (x_i - \mu) \\
&= \frac{1}{n-1} \sum_{i=1}^n ((x_i - \mu)^T V) (V^T) (x_i - \mu) \\
&= \frac{1}{n-1} \sum_{i=1}^n V^T (x_i - \mu) (x_i - \mu)^T V \\
&= \frac{1}{n-1} V^T [\sum_{i=1}^n (x_i - \mu) (x_i - \mu)^T] V \\
&= V^T \Sigma V
\end{aligned}$$

où Σ est la matrice de variances covariances, symétrique définie positive.

Ainsi le fait de maximiser la variance des observations projetées avec $V^T V = 1$ revient à trouver une solution à un problème de maximisation sous contraintes. Il faut donc utiliser la fonction de Lagrange : $L = V^T \Sigma V + \lambda(1 - V^T V)$.

Nous calculons les conditions nécessaires d'optimalité :

$$\partial_V L = 0 \quad \implies \quad \Sigma V = \lambda V$$

Comme la matrice de variance-covariance est symétrique, ses valeurs propres sont réelles. Comme elle définie positive, les valeurs propres sont toutes positives et les vecteurs propres peuvent être choisis orthonormés.

La variance des observations projetées s'écrit alors : $\sigma_V^2 = V^T \Sigma V = V^T \lambda V = \lambda$.

Donc la solution est de projeter les données sur le vecteur propre ayant la valeur propre la plus élevée.

Afin de trouver les autres axes de variance maximale, nous recherchons itérativement les axes avec $V_{i-1}^T V_{i-1} = 1$ et $V_{i-1}^T V_i = 0$, où les V_i sont les vecteurs propres correspondant aux valeurs propres ordonnées de façon décroissante et qui sont naturellement orthonormés.

3.5.3 Attaque *template* avec ACP

3.5.3.1 Profilage

Comme pour l'attaque avec recherche de points d'intérêt, on a besoin de P_k traces correspondants à O_{tpl} opérations. Les traces $\{t_{p_k}\}_{p_k}^{P_k}$ sont des vecteurs de n dimensions.

De la même manière, le modèle gaussien de bruit est considéré en supposant que les traces $\{t_{p_k}\}_{p_k}^{P_k}$ sont construites à partir de la distribution gaussienne multivariée (voir équation (3.4)). Pour classifier une trace on utilise le principe du maximum de vraisemblance, comme expliqué dans 3.3.

L'utilisation de l'analyse par composantes principales permet de réduire notre échantillon de traces et de le représenter avec moins de composantes. On considère qu'on dispose d'un ensemble d'observations $\{t_k\}_{k=1}^K$ qui correspondent aux moyennes empiriques des traces (n -dimensionnelles) associées à l'ensemble des opérations. L'ACP choisit les premières directions principales $\{w_i\}_{i=1}^m$, telles que $m \leq n$ forment une base de dimension m du sous-espace qui prend la variance maximale des $\{t_k\}_{k=1}^K$, c'est-à-dire (voir section 3.5.2) une base constituée des vecteurs propres de la matrice de covariance empirique, donnée par :

$$\bar{S} = \frac{1}{K-1} \sum_{k=1}^K (t_k - \bar{t})(t_k - \bar{t})^T \quad (3.7)$$

où $\bar{t} = \frac{1}{K} \sum_{k=1}^K t_k$ est la moyenne des moyennes des traces correspondants à chaque *template*.

Soit $T = (t_1 - \bar{t}, \dots, t_K - \bar{t})$ la matrice des traces moyennes centrées. Par définition, la matrice de covariance empirique est donnée par $\frac{1}{K} T T^T$.

Notons que U et Δ sont respectivement les matrices des vecteurs propres et des valeurs propres de $\frac{1}{K} T T^T$.

Ainsi on a : $(\frac{1}{K} T T^T) U = U \Delta$.

On multiplie les deux cotés de l'équation par T , ce qui implique :

$$T * (\frac{1}{K} T T^T) U = T * U \Delta \Rightarrow (\frac{1}{K} T T^T T) U = (T U) \Delta$$

Et donc :

$$\bar{S}(TU) = (TU) \Delta \quad (3.8)$$

Cette dernière expression montre que TU est la matrice des K vecteurs propres de $\bar{S}$. Dans le but d'avoir une base orthonormée, on doit normaliser ces vecteurs propres, en prenant :

$$\mathbf{V} = \frac{TU}{\|TU\|} \quad (3.9)$$

et ainsi les directions principales $\{W_i\}_{i=1}^I$ sont les colonnes de $\mathbf{V}$ correspondants aux I plus grandes valeurs de Δ .

Notons $\{W\}_{N \times I}$ la matrice des principales directions.

Cette matrice de projection va nous permettre de réduire l'espace des échantillons en estimant les moyennes projetées par :

$$\nu_k = W^T \mu_k \quad (3.10)$$

et les matrices de covariances des traces projetées par :

$$\Lambda_k = W^T \Sigma_k W \quad (3.11)$$

3.5.3.2 Attaque en ligne

Pour une trace tr acquise contenant un secret s (la clé correcte) sur le circuit cible, nous commençons par faire une projection de cette trace dans la nouvelle base. Nous avons donc une nouvelle trace avec un nombre de composants réduit.

$$tr_p = W^T tr \quad (3.12)$$

Nous calculons ensuite la probabilité de la clé sachant la fuite tr_p pour chaque *tem-plate* (ν_i, Λ_i) :

$$Pr(s \mid tr_p) = \frac{Pr(tr_p \mid s)Pr(s)}{\sum_{s_g} Pr(tr_p \mid s_g)Pr(s_g)} \quad (3.13)$$

tels que s_g sont les clés supposées, et :

$$Pr(tr_p \mid s) = \frac{1}{\sqrt{(2\pi)^N |\Lambda_i|}} \exp\left(-\frac{1}{2}(tr_p - \nu_i)^T \Lambda_i^{-1} (tr_p - \nu_i)\right) \quad (3.14)$$

Cette formule est valable pour une seule trace.

Pour faire une attaque avec n traces nous utilisons la formule :

$$Pr(s \mid tr_p^1 \dots tr_p^n) = \frac{\prod_{i=1}^n Pr(tr_p^i \mid s)Pr(s)}{\sum_{s_g} \prod_{j=1}^n Pr(tr_p^j \mid s_g)Pr(s_g)} \quad (3.15)$$

Chapitre 4

Vers un modèle pour les attaques *template* ?

Nos premières expérimentations sur les attaques *template* ont montré que retrouver la clé du chiffrement en se basant uniquement sur les suppositions de clés n'est pas évident. Dès le début de nos expérimentations, nous avons constaté qu'il ne faut pas utiliser dans la phase d'attaque une trace qui a servi pendant la phase d'entraînement. En effet, le fait qu'elle soit, en quelque sorte, « contenue » dans les moyennes fait que sa classification est beaucoup plus simple, au détriment du réalisme de l'attaque : dans le « monde réel », les traces d'entraînement et d'attaque ne proviennent jamais du même système et ne peuvent donc pas être identiques. Ces premières difficultés nous ont poussées à une analyse ciblée et ont animé les expérimentations que nous détaillons dans les sections suivantes.

4.1 Les attaques *template* et la génération de clés de tours

Dans cette étude, deux campagnes d'acquisition de 20.000 traces ont été réalisées à partir du chiffrement par l'algorithme DES de messages clairs aléatoires :

- une première campagne a utilisé des clés telles les six bits de clé de tour entrant dans la S-Box 1 du premier tour étaient aléatoires, tous les autres étant nuls et
- une seconde campagne, durant laquelle les clés étaient totalement aléatoires.

L'étude des résultats (positifs et négatifs) des attaques menées à partir de ces deux campagnes d'acquisition nous ont permis de mieux comprendre l'attaque *template*. Dans la suite, nous utiliserons l'analyse par composantes principales pour le choix des points d'intérêt, et nous attaquons la valeur de la clé à l'entrée de la première S-box du premier tour.

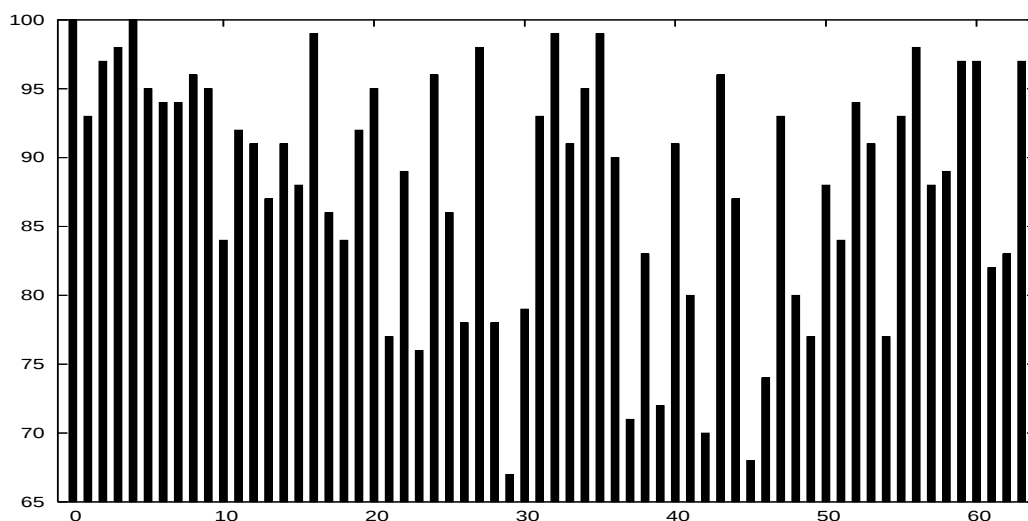


FIG. 4.1 – Résultats de l’attaque *template* utilisant 32 composantes.

4.1.1 Campagne #1 : clé nulle, sauf les 6 bits entrant dans la S-box 1 au 1er tour

Nous refaisons dans les détails l’attaque présentée dans [4]. L’attaque menée à partir sur la campagne #1 retrouve la clé en utilisant une seule trace du même type que celles de la phase d’apprentissage avec une très bonne probabilité. La figure 4.1 montre le taux de succès d’une attaque utilisant à chaque fois 32 vecteurs propres et avec une seule trace extraite de l’appareil attaqué. L’axe des “x” correspond à la clé réelle tandis que celui des ordonnées donne la probabilité de succès de l’attaque dans l’intervalle [65 % :100 %]. La probabilité s’approche très rapidement de 100 % en ajoutant plus de traces correspondant aux clés légèrement résistantes.

4.1.2 Campagne #2 : clés aléatoires

Cette campagne correspond au cas général où on prend en compte toutes les possibilités expérimentales notamment au niveau du bruit. En attaquant une trace cible nous n’avons pas pu retrouver la clé aussi aisément que pour la campagne #1.

Pour comprendre les raisons de cet échec, nous analysons la distribution de la densité de probabilité pour chacune des 64 clés. Cette analyse se base sur toutes les traces des deux campagnes. La procédure consiste à :

- calculer la moyenne et la covariance pour chaque ensemble de traces correspondant à chacune des 64 clés (les *templates*),
- faire la projection sur les deux premiers vecteurs propres donnés par l’ACP de chaque moyenne, covariance et trace des 64 ensembles.
- calculer les densités de probabilités gaussiennes pour chaque trace en l’associant à

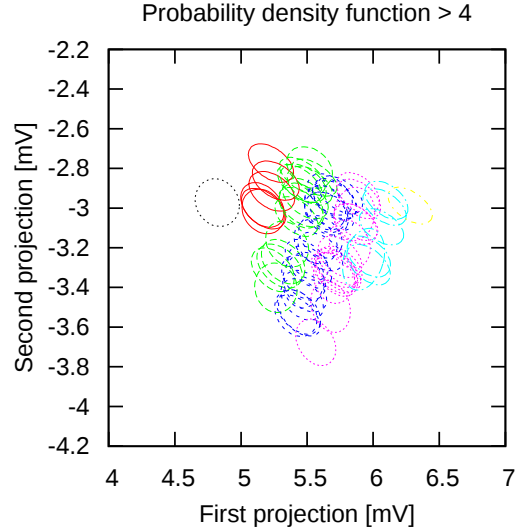


FIG. 4.2 – Les densités de probabilité correspondants à la campagne d’acquisition #1.

sa clé.

- examiner visuellement en deux dimensions les disparités ou ressemblances entre les densités de probabilités.

Nous illustrons ces densités de probabilité sur les figures 4.2 et 4.3 et remarquons que ceux de la campagne #1 sont distinctes alors que celles de la campagne #2 sont indiscernables. Nous remarquons également sur la figure 4.2 que les densités de probabilité de la campagne #1 suivent une logique dépendante du poids de Hamming de la clé.

4.1.3 Comparaison des attaques

Le vecteur propre associé à la plus grande valeur propre obtenue par la campagne #1 est représenté sur 4.4, tandis que le même vecteur propre obtenu par la campagne #2 est représenté sur 4.5. Les deux figures indiquent les instants où l’énergie est dissipée pendant que la clé est chargée et utilisée.

Le fait que l’énergie dissipée contribue au vecteur propre, nous remarquons sur la figure 4.4 que même si nous attaquons uniquement au premier tour, il y a des *pics* secondaires non liés à l’usage de la clé. Comme les messages d’entrée sont aléatoires, leurs chemins de données ne contribuent pas à la consommation moyenne.

Dans la campagne #1, où l’attaque est réussie, le vecteur propre (4.4) est parfaitement corrélé à une activité au 16 tours de DES. Les *pics* au début des tours {1, 3, 9, 10, 16} ont la même amplitude. Alors que, pour la campagne #2, le premier vecteur

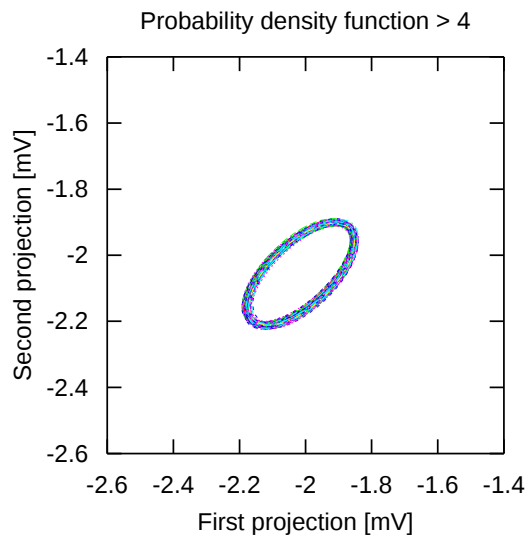


FIG. 4.3 – Les densités de probabilité correspondant à la campagne d’acquisition #2.

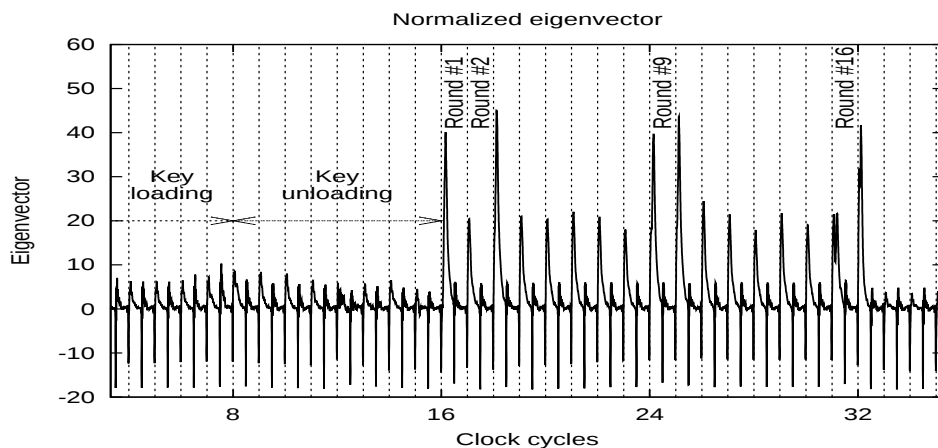


FIG. 4.4 – Premier vecteur propre de la campagne #1 avec l’“identité” comme classification.

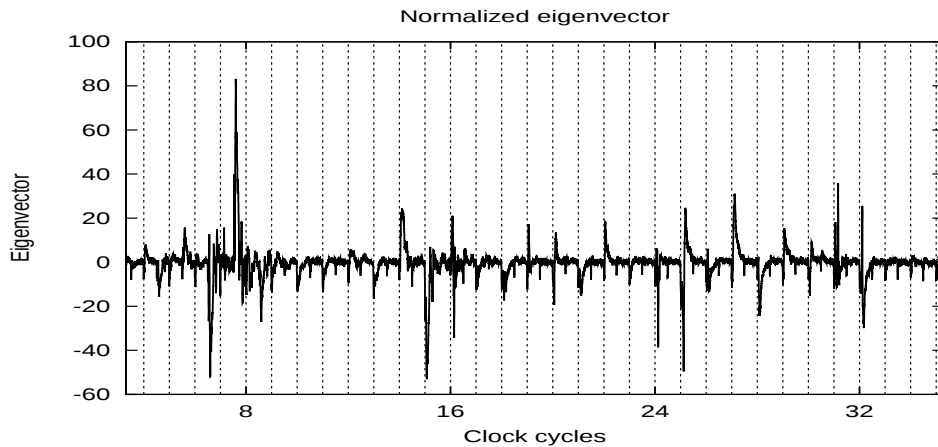


FIG. 4.5 – Premier vecteur propre de la campagne #2 avec l’“identité” comme classification.

propre est décorrélé d’avec l’activité de la clé, et donne même un *pic* avec une grande amplitude juste avant l’instant de déchargement de la clé. Cette fausse indication donne un indice sur l’échec de l’attaque. Face à ce considération, nous présenterons des outils à la section 4.2 pour avoir des explications.

4.2 Nouveau modèle d’attaque

Sachant que les boîtes de substitution ne sont pas l’unique cible des attaques *template*, l’attaquant peut choisir entre différentes stratégies. Cela amène à introduire de nouveaux modèles d’assaillants.

L’attaquant qui veut récupérer des informations sur un secret commence par isoler une partie du secret, sur lequel une recherche exhaustive est possible. Puis, il construit des classes qui représentent la part du secret à extraire. Généralement, il y a une classe par valeur possible de la partie de secret isolée mais rien ne s’oppose à ce qu’une fonction plus complexe soit utilisée. Lorsque le dispositif de fonctionnement interne est inconnu, il est naturel de créer une classe par valeur possible de la partie isolée du secret puisque cela reflète le fait que l’attaquant a à choisir les classes en “aveugle”. Une autre stratégie qui, va aider à améliorer sensiblement l’attaque, est discuté dans la section suivante 4.2.1.

4.2.1 Les attaques *template* avec le modèle du poids de Hamming

L’association d’une classe à une valeur possible de sous-clé de tour conduit à des attaques infructueuses. L’étude des *templates* permet de comprendre la raison de cet échec : les *templates* construits sont très proches les uns des autres, comme indiqué

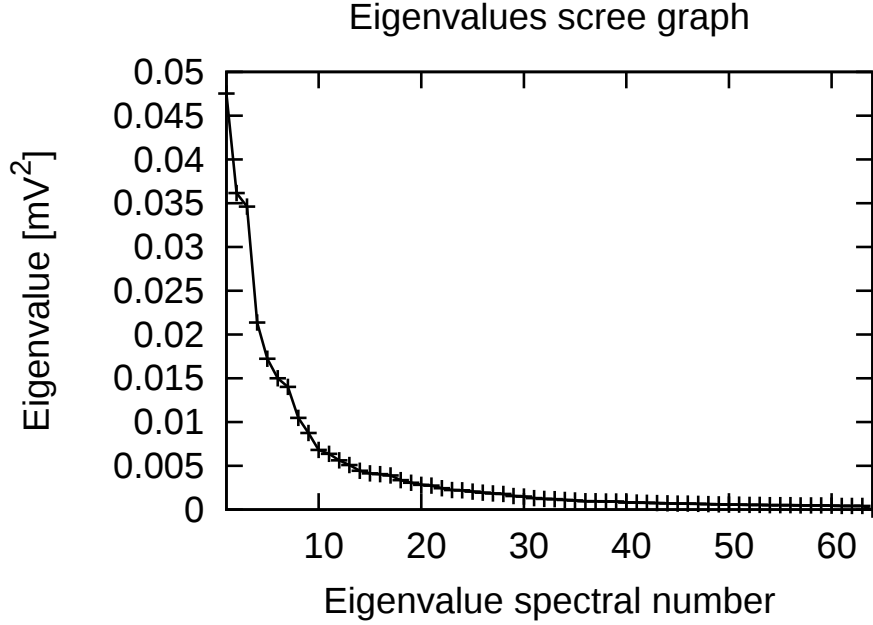


FIG. 4.6 – Les 2^6 valeurs propres de la campagne d’acquisition #2 avec ϕ_0 .

dans la figure 4.3. Les $K = 2^6$ valeurs propres sont représentés figure 4.6. Définissons le **nombre de valeurs propres “représentatives”** comme le nombre minimal de valeurs propres représentant 85 % de la variance totale :

$$K_{\text{représentative}} \doteq \min \left\{ k \in]0, 2^6[\mid \sum_{l=0}^k \lambda_l \geq 0.85 \times \sum_{l=0}^{2^6-1} \lambda_l \right\} = 20 \quad (4.1)$$

D’après notre construction dans la section 4.1, nous définissons par $\boxed{\phi_0 \doteq Id}$ la fonction de classification que nous avons utilisé dans le profilage. Cependant, du fait que les bits aléatoires sont utilisés lors de chaque tour et que les autres bits de la clé sont nuls, le nombre de transitions dans les registres à décalage est *grosso modo* le double du poids de Hamming des 6 bits aléatoires.

Cela conduit à la situation paradoxale selon laquelle la distance de Hamming (sachant que l’état précédant est égale la constante ‘zéro’) dégénère en un poids de Hamming. Ceci est illustré dans le tableau A.1(voir annexe A.1) .

En effet, ce dernier décrit la campagne d’acquisition #1, durant laquelle chacune des 64 clés utilisées partageaient la propriété suivante : au premier tour, la clé utilisée pour chacune des S-boxes de 2 à 8 était 0x00. Le contenu initial du registre CD est aussi examiné : Le vecteur propre 4.5 montre qu’il y a une activité porteuse d’information juste au début du chargement de la clé.

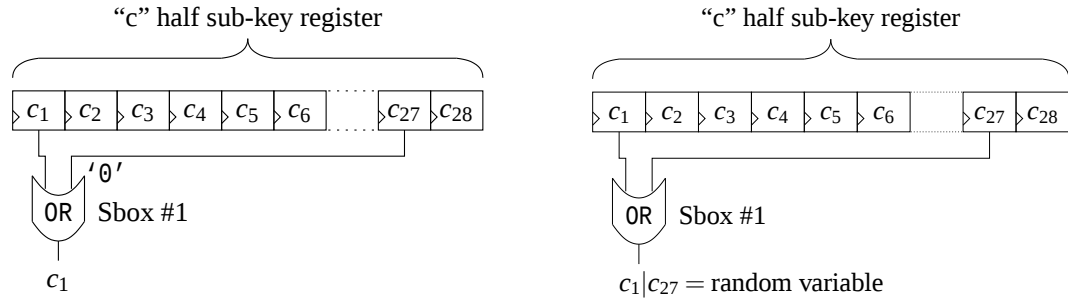


FIG. 4.7 – La logique combinatoire dans le registre “C” avec bruit (à droite) et conformes (à gauche)

Le poids de Hamming du registre CD, qui est constant pendant le key schedule de DES, est donné dans la dernière colonne du tableau A.1 (voir annexe A.1). Ceci explique pourquoi les 2^6 densités de probabilités (voir 4.2) se regroupent en 7 sous-classes de poids de Hamming constant, ce qui la cause de la dégénérescence de la distance de Hamming.

4.2.2 Les attaques *template* et le modèle de la distance de Hamming

Dans la section précédente, un modèle de consommation inconnu a été supposé. Néanmoins, dans certaines situations, le modèle de consommation peut être déduit grâce à une analyse de dissipation de chaque partie de l’algorithme de chiffrement. Notre objectif sera d’augmenter la plus grande valeur propre des modèles et par conséquent la variance totale. Notre motivation est que les valeurs propres qui correspondent à la variance sont par conséquent liés à la dispersion des densités de probabilité. Maintenant, une attaque *template* réussira mieux si les densités sont bien séparées les unes des l’autres.

Dans un ASIC *open-source*, comme SecMat, le modèle de consommation est en effet bien connu. La consommation est proportionnelle au nombre de commutations des portes logiques L’énergie dissipée par une partie du matériel est donc mesurée par une distance de Hamming.

Cependant, la meilleure façon de distinguer les *templates* est de les construire de telle manière que leurs dissipations diffèrent de manière représentative les unes des autres.

En nous aidant, des résultats de la section précédente, une meilleure application est la distance de Hamming entre deux valeurs successives de la clé dans le registre CD. Sachant que LS est le “décalage circulaire à gauche” du DES, deux fonctions de classification présentent un intérêt réel :

- $\phi_1 : k \mapsto k \oplus \text{LS}(k)$ et
- $\phi_2 : k \mapsto k \oplus \text{LS}^2(k)$,

Compte tenu de notre architecture, nous utilisons ϕ_1 . La classification des *templates* avec cette nouvelle fonction sera proche de la fonction “Id” utilisée dans sur la campagne #1. En effet, un autre facteur de la logique combinatoire utilisée provoquera certaines incohérences. Ce phénomène, habituellement appelé “bruit algorithmique”, est représenté

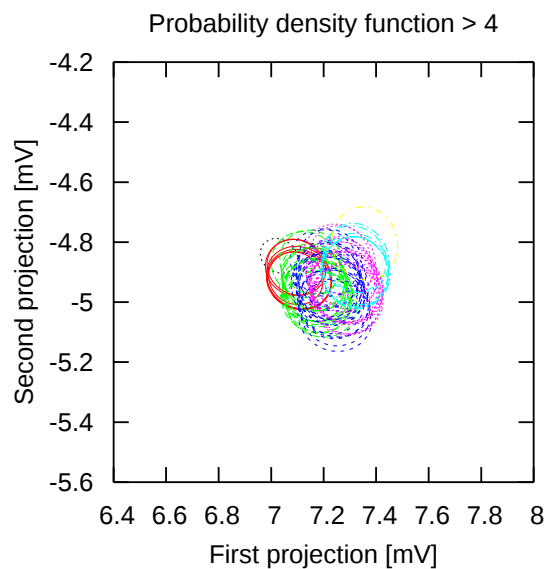


FIG. 4.8 – Les densités de probabilités correspondants à la campagne d’acquisition #2 avec ϕ_1 .

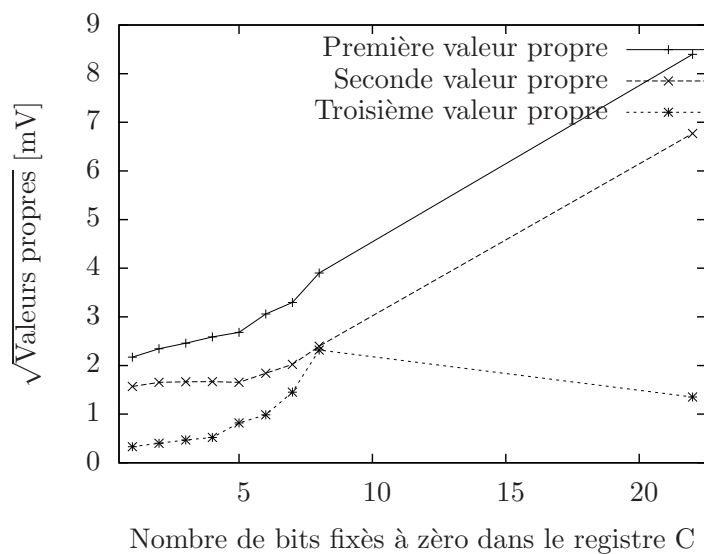


FIG. 4.9 – Les trois principales valeurs propres lorsque 1, 2, ..., 8 bits de la clé sont forcés à zéro.

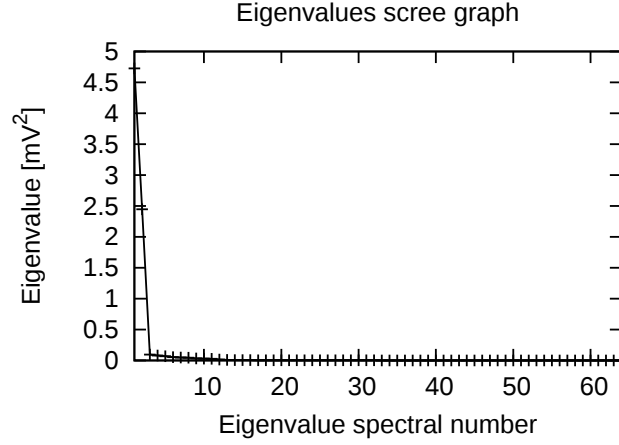


FIG. 4.10 – les 2^6 valeurs propres de la campagne d’acquisition #2 avec ϕ_1 .

dans la partie inférieure de la figure 4.7. Dans cette figure, une seule porte de la S-box est représentée. Par exemple la dissipation de la S-box est liée à la valeur du 27ème bit du registre C. Ainsi quand le bit est égale à zéro nous aurons une consommation équivalente à la campagne #1.

Nous calculons la racine carrée des valeurs propres pour illustrer la proportionnalité de la consommation à la quantité de transitions. Nous avons :

- $\sqrt{70.50} = 8.4$ mV de la logique active est recueillie dans la campagne #1, alors que
- uniquement $\sqrt{4.71} = 2.2$ mV de l’activité entière de la clé est extraite dans la campagne #2.

A partir des figures précédentes, nous concluons que l’activité de la S-box est $(8, 4 - 2, 2)/2, 2 = 2, 9$ fois plus élevée que le registre CD seul. l’effet d’un changement des entrées d’une boîte “S” a des conséquences ayant à la fois une durée plus grande et des conséquences au niveau de la consommation plus importantes que celles de la commutation dans CD. La propagation dans la logique de la S-box (à *tous* les tours, non seulement au *premier*) est donc très profonde et est une des raisons du succès de l’attaque *template* sur la campagne #1.

Cette affirmation est corroborée par le fait que les valeurs propres augmentent quand quelques traces avec certains bits de clés (1 ou plusieurs parmi l’ensemble {9, 1, 58, 50, 42, 34, 26, 18 }) sont choisis pour construire les *templates*. Ces bits sont 8 bits successif du registre C. Le résultat est montré dans la figure 4.9. Une explication complète de la “tendance” exigerait plus de mesures, parce que quand 8 bits sont forcés à zéro, il en reste seulement 11 traces par *template* dans la campagne #2!

Grâce aux nouvelles fonctions, les valeurs propres deviennent significatives. Nous avons donc un meilleur distingueur. Les vecteurs propres qui sont des combinaisons linéaires de traces mettent en évidence la consommation d’une sous-partie utile à l’atta-

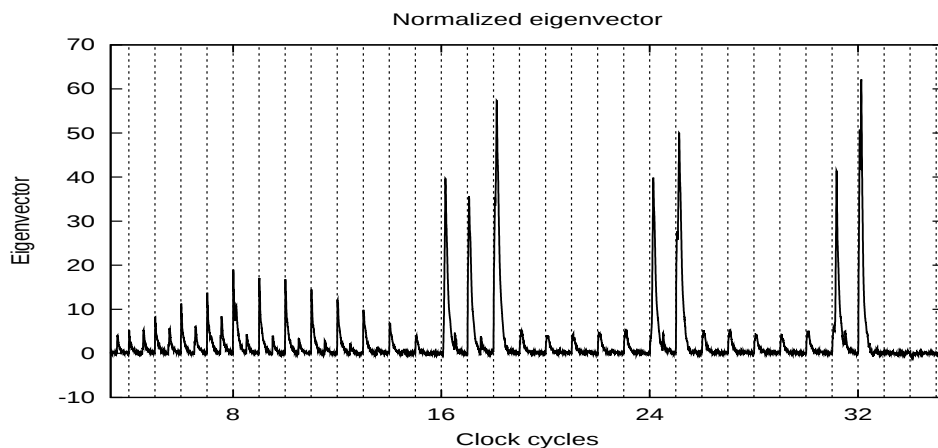


FIG. 4.11 – Premier vecteur propre de la campagne d’acquisitions #2 avec ϕ_1 .

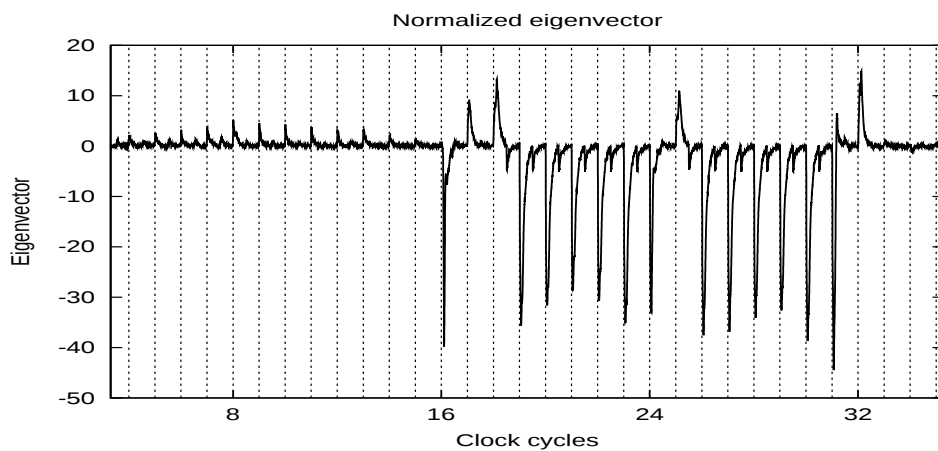


FIG. 4.12 – Second vecteur propre de la campagne d’acquisitions #2 avec ϕ_1 .

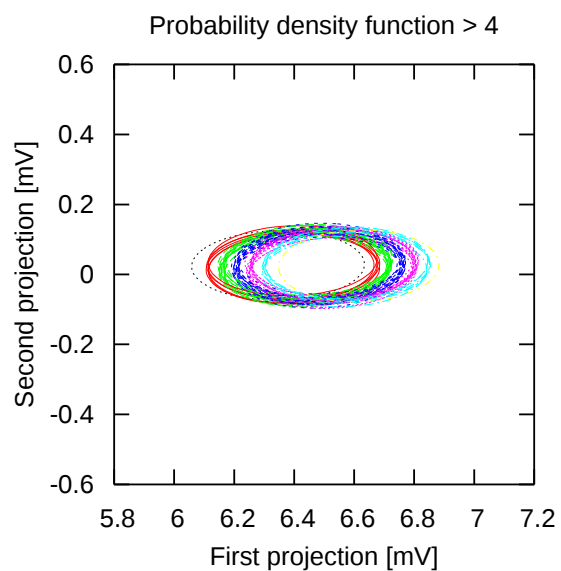


FIG. 4.13 – Les densités de probabilité pour la campagne d’acquisition #2 avec ϕ_2 .

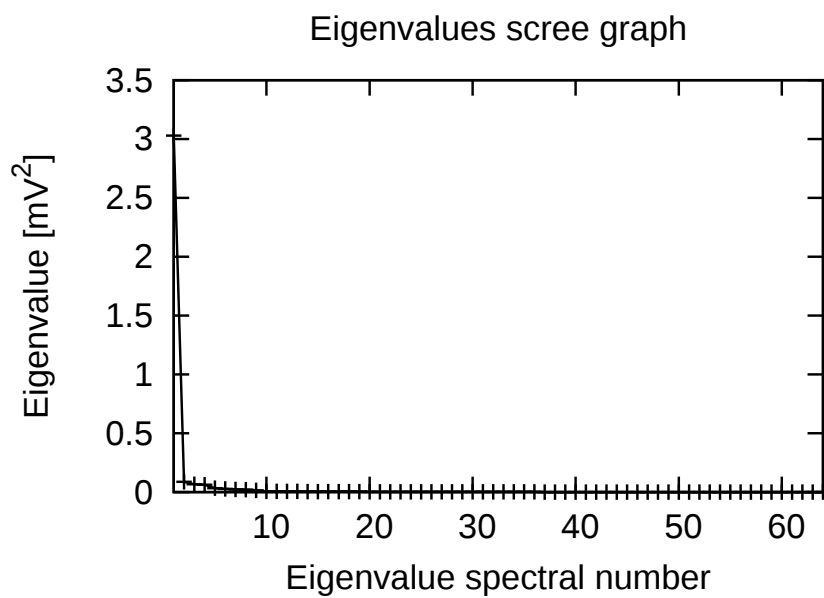


FIG. 4.14 – les 2^6 valeurs propres pour la campagne d’acquisition #2 avec ϕ_2 .

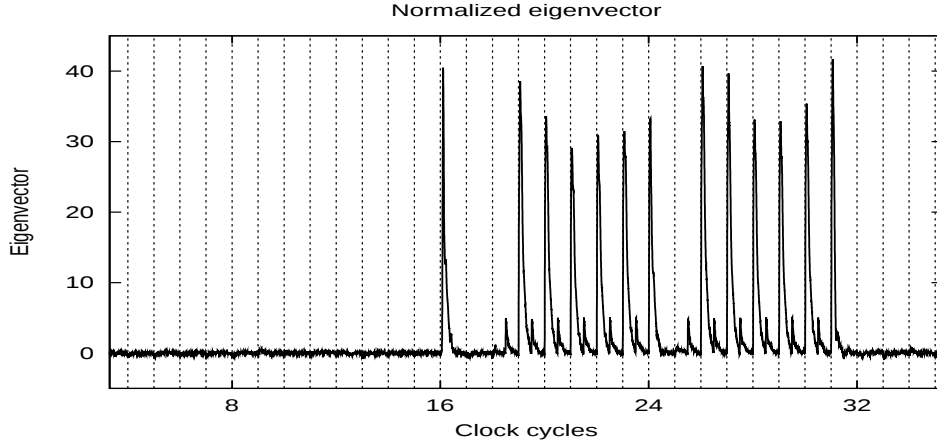


FIG. 4.15 – Premier vecteur propre de la campagne d’acquisition #2 avec ϕ_2 .

quant et indiquent exactement l’importance de la fuite dans le temps.

Notons que le *template* construit avec ϕ_1 comme fonction de classification a deux valeurs propres représentatives. Le premier vecteur propre (figure 4.11) représente l’activité LS dans la génération de clés de tours. Le second vecteur propre (figure 4.12) saisit réellement la dissipation du décalage des deux bits LS^2 . Cette opération est en effet réalisée d’abord en décalant par un bit (il y a un multiplexeur dédié à cette opération) et le second en décalant par un autre bit (un multiplexeur dédié s’occupe aussi de cette opération). Ces deux opérations, se produisant à proximité les uns des autres dans le temps, se comportent réellement comme deux décalages LS, et donc dissipent $\phi_1 \circ \phi_1 = \phi_1^2$. Étant donné que :

$$\begin{aligned}
 \phi_1^2(c) &= c \oplus LS(c) \oplus LS(c \oplus LS(c)) \\
 &= c \oplus LS(c) \oplus LS(c) \oplus LS^2(c) \quad // \text{ par linéarité dans LS} \\
 &= c \oplus LS^2(c) = \phi_2(c),
 \end{aligned}$$

il n’est donc pas surprenant de constater que le deuxième vecteur propre de l’ACP avec comme fonction de classification ϕ_1 (figure 4.12) soit semblable au premier vecteur propre de ϕ_2 (modulo un signe arbitraire, voir figure 4.15).

Le *template* construit avec ϕ_2 (figure 4.13) est clairement unidirectionnelle et il est conforme avec l’existence d’une seule valeur propre représentative (figure 4.14).

Dans la figure 4.16, les vecteurs propres sont comparés à des traces différentielles DPA obtenues à partir de la campagne d’acquisition #2. Les figures sont très similaires, ce qui renforce l’interprétation du premier vecteur propre comme le principal indicateur des instants de fuites.

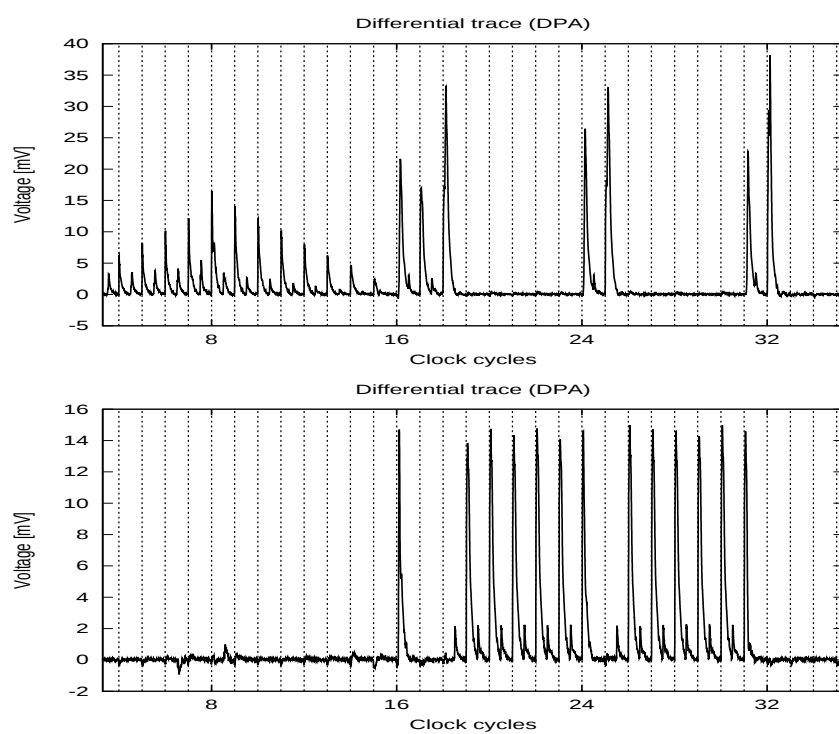


FIG. 4.16 – Traces différentielles obtenues avec ϕ_1 (en haut) et ϕ_2 (en bas).

TAB. 4.1 – Comparaison entre les valeurs propres de la campagne #2 pour les trois fonctions de classifications.

Première valeur propre		
ϕ_0	ϕ_1	ϕ_2
0.05 mV ²	4.73 mV ² ($\times 99$ w.r.t ϕ_0)	3.03 mV ² ($\times 64$ w.r.t ϕ_0)
Somme des 2 ⁶ valeurs propres		
ϕ_0	ϕ_1	ϕ_2
0.30 mV ²	7.77 mV ² ($\times 26$ w.r.t ϕ_0)	3.46 mV ² ($\times 12$ w.r.t ϕ_0)

Le tableau 4.1 résume les valeurs propres obtenues dans la campagne #2 avec les fonctions de classification ϕ_0 , ϕ_1 et ϕ_2 . Il est clair que l'utilisation des modèles de consommation ϕ_1 ou ϕ_2 augmente de plusieurs ordres de grandeur les variances dans les sous-espaces principaux.

Chapitre 5

Améliorations pratiques des attaques *template*

Ce chapitre examine la pertinence du cadre théorique sur les attaques par profilage présenté par F.-X. Standaert et al. à Eurocrypt 2009. Cette analyse consiste en une étude de cas sur la base de mesures obtenues expérimentalement à partir d'un opérateur cryptographique matériel. La structure de l'architecture embarquée, source d'une plus grande quantité de bruit algorithmique, fait que les investigations que nous décrivons sont plus complexes que des analyses formelles effectuées sur des mesures ou sur des données obtenues par simulation. Dans ce contexte, nous montrons cependant que deux techniques peuvent grandement améliorer à la fois le profilage et l'attaque effective. Tout d'abord, nous explorons la pertinence des différents choix pour les variables sensibles. Nous montrons qu'un attaquant ayant la connaissance du registre des transferts qui se produisent au cours des opérations cryptographiques peut sélectionner les distingueurs les plus adéquats, ce qui augmente son taux de succès. Ensuite, nous introduisons une méthode basée sur le seuillage de la fuite de données qui permet d'accélérer le profilage ainsi que l'attaque. En effet, en s'appuyant sur ce principe, il est possible de prévoir la forme de certains vecteurs propres, anticipant ainsi leur estimation en vue de leur valeur asymptotique par la réduction à zéro de composantes contenant principalement de la non-information et par conséquent du bruit. Cette méthode renforce l'attaque, puisque, concrètement, nous gagnons en vitesse dans la phase d'attaque en ligne et augmentons le taux de succès.

5.1 Les attaques *template* : de la ACP à SECMAT

Nos investigations sont effectuées sur un cryptoprocresseur DES non protégés avec une architecture itérative. Nous estimons l'entropie conditionnelle et le taux de succès pour différents modèles de fuite : En fait, les modèles sont construits à partir de distingueurs de classifications lors du profilage. Nous cherchons à faire des comparaisons qui nous permettront de mieux caractériser la fuite. Nous sommes ainsi confrontés au problème

du choix des variables sensibles pertinentes. En fait, il a déjà été souligné, par exemple dans la section 2 de [76], qu'il existe des variables plus sensibles que d'autres du point de vue de l'attaquant. Ainsi, la sécurité d'un dispositif cryptographique ne devrait pas être traitée dans un cadre général, mais doit dépendre de chaque variable sensible. Nous devons choisir une variable qui est sensible (c'est à dire reliée à la clé) et prévisible (la taille de la sous-clé est gérable par une recherche exhaustive). Cela conduit à plusieurs questions :

- où attaquer ? Par exemple, à l'entrée ou à la sortie d'une S-box ?
- les valeurs simples ou la distance entre deux valeurs consécutives ? Dans ce dernier cas, avons-nous besoin du schéma de la mise en œuvre de l'algorithme dans le circuit ?
- en envisageant une transition comme variable sensible, est-il utile d'examiner la distance de Hamming au lieu de la distance *simple*¹ ?

Pour trouver le meilleur compromis sécurité/attaquant, nos expériences tenteront de répondre à ces questions. Nous étudions les modèles suivants de fuite :

- Le **modèle A** est l'entrée de la première S-box, c'est un modèle sur 6-bit, défini par : $(R_0 \oplus K_1)[1, 6]$.
- Le **modèle B** est la sortie de la première S-box, c'est un modèle sur 4-bit, défini par : $S(R_0 \oplus K_1)[1, 4]$
- Le **Modèle C** est la valeur du premier tour correspondant au fan-out sortant de la première S-box , c'est un modèle sur 4-bit, défini par : $R_1\{9, 17, 23, 31\} = P^{-1}(R_1)[1, 4] = (S(R_0 \oplus K_1)) \oplus P^{-1}(L_0)[1, 4]$.
- Le **Modèle D** est la transition du modèle C.
- Le **Modèle E** est le poids de Hamming du modèle D.

Nous illustrons notre étude sur le premier tour, le dernier tour donnant des résultats similaires. Les définitions mathématiques des modèles de A à E sont reprises dans la table 5.1, en reprenant les notations du NIST FIPS 46 pour quelques fonctions internes à l'algorithme DES et en utilisant S comme raccourci pour $S_1 || S_2 || \dots || S_8$. Pour une meilleure lisibilité, nous fournissons également une illustration du flux de données dans la figure 5.1.

Le terme "modèle" est utilisé pour désigner le choix de la variable qui dépend du clair ou du chiffré, et dont les valeurs déterminent le nombre de *templates*. Ce vocabulaire peut être trompeur, car il a également été utilisé dans la littérature pour représenter d'autres quantités. Dans le document original sur la DPA [38], le modèle est en fait appelé "fonction de sélection". Dans [62], le terme "modèle de fuite" qualifie la façon dont une variable sensible est concrètement liée à la fuite ; par exemple, selon leur terminologie, le choix du poids de Hamming est un modèle de "réduction" de la variable sensible. À cet égard, E représenterait un modèle de fuite de D. Nous n'utilisons pas ce vocabulaire, car nous pensons que l'attaquant ne peut pas faire la différence entre une fuite interne et une observation physique. Ce détail dirige vers une sophistication de l'attaque, qui est mieux saisie par le concept derrière les attaques stochastiques [71]. La notion de "variable

¹Étant donnés deux chaînes de bits x_0 and x_1 Nous appelons leur distance simple le mot $x_0 \oplus x_1$, opposée à leur distance de Hamming définie comme l'entier $|x_0 \oplus x_1|$.

Les différents Modèles	description Mathématique	Abbre-viation	Nature	Dist-ance
A	$(R_0 \oplus K_1)[1 : 6]$	R0+K1	Combi.	Non
B	$S(R_0 \oplus K_1)[1 : 4]$	S(R0+K1)	Combi.	Non
C	$R_1\{9, 17, 23, 31\} =$ $P^{-1}(R_1)[1 : 4] =$ $(S(R_0 \oplus K_1)) \oplus$ $P^{-1}(L_0)[1 : 4]$	R1	Séquentiel	Non
D	$(R_0 \oplus R_1)\{9, 17, 23, 31\}$	R0+R1	Séquentiel	Oui
E	$ (R_0 \oplus R_1)\{9, 17, 23, 31\} $	R0+R1	Séquentiel	Oui

TAB. 5.1 – Les cinq modèles examinés.

sensible” n’est pas adéquate non plus, car il n’y a pas de lien direct entre la fuite et une valeur spécifique de l’algorithme (comme pour le modèle A, B ou C). Donc, pour résumer, nous rappelons que nous employons le terme modèle pour définir la relation entre une trace et son *template*. Nous notons que ces modèles peuvent être considérés comme deux familles de distingueurs. Les modèles A et B sont concernés par des instants spécifiques au cours du chiffrement, alors que les modèles B, C et D se concentrent sur les registres. Pour plus de compréhension, ces modèles sont représentés sur la figure 5.2.

Chaque cas étudié correspond à un nombre de *templates*. Le modèle A utilise $64 = 2^6$ *templates*, car il représente l’activité à l’entrée d’une S-box, alors que le modèle B s’applique à $16 = 2^4$ *templates* qui sont les valeurs de sortie d’une S-box. Les Modèles C et D sont également composés de 16 *templates*, et enfin, le modèle E se concentre sur les 5 *templates* qui sont la classifications des différentes valeurs de la distance de Hamming sur 4 bits. La campagne d’acquisition consiste en une collection de 80 000 traces. La moitié des ces traces est utilisée pour le profilage et l’autre moitié pour l’attaque en

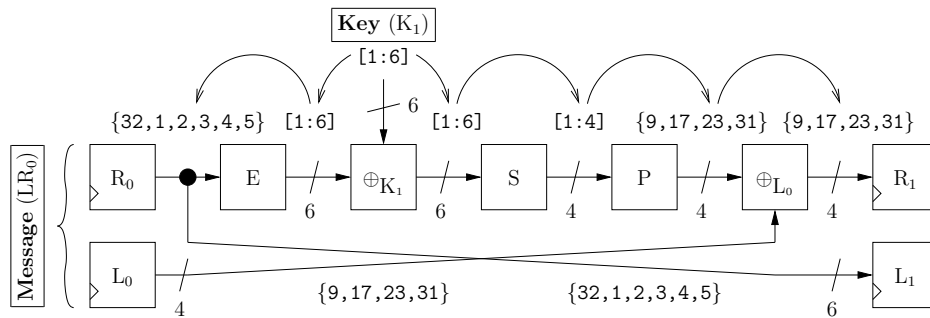
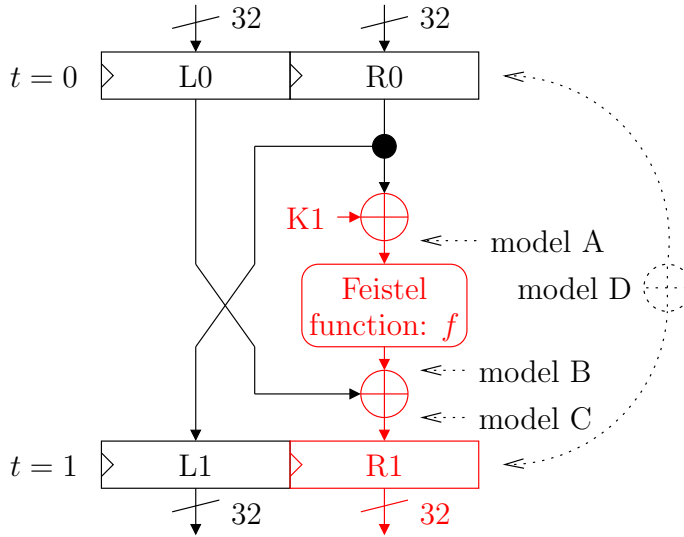


FIG. 5.1 – Circulation des données dans le DES impliqué dans l’attaque au premier tour.

ligne. Les *templates* comme indiqué ci-dessus sont dans la pratique, un ensemble de moyennes et de covariances pour les classes construites à partir des modèles. En outre l'analyse par composante principale permet de construire des nouvelles composantes qui seront utiles pour réduire nos échantillons de données en vue des attaques en ligne. En pratique, le premier vecteur propre est utilisé afin de faire une projection des données de consommation du circuit. En effet, il est le vecteur le plus important car il concerne la plus grande variance. On peut également utiliser le deuxième ou troisième vecteur propre s'ils sont pertinents, à savoir s'ils correspondent à une variance importante. Dans ce cas, nous pouvons augmenter la valeur de l'attaque. D'autre part, en utilisant des vecteurs propres qui n'apportent que la non information ou du bruit, nous allons seulement à nuire à l'attaque. Nous désignerons ces vecteurs comme des "mauvais" vecteurs propre, par oppositions aux premiers vecteurs propres qui sont les plus importants. Dans la figure 5.3 on peut voir la différence entre un "bon" (le second) et un "mauvais" (le treizième) vecteur propre. Dans un circuit non protégé, il apparaît clairement que la fuite est bien localisée dans le temps. Par conséquent, un "bon" vecteur propre (correspondant à une grande valeur propre) a également des composantes à peu près nulles partout mais supérieures à zéro aux véritables moments de fuites. Cette propriété sera d'ailleurs la base d'une amélioration qui sera présentée plus loin dans ce document. Selon les vecteurs propres représentés à la figure 5.4, nous pouvons voir précisément les instants où notre circuit fuit suivant les modèles A, B, D, E ou F. Plus une composante du vecteur propre

Attack on the first round of DES



Caption: black = known values; red = unknown sensitive values

FIG. 5.2 – Les différents modèles du tableau 5.1.

à une grande valeur, plus la fuite est importante à cet instant.

5.2 Amélioration des attaques grâce à un modèle de fuite adéquat.

5.2.1 Taux de succès

Les taux de succès sont représentés dans la figure 5.5 comme une fonction de deux paramètres : le premier est le nombre de traces du profilage et le second est le nombre de traces utilisées pour l’attaque en ligne. Pour toutes les représentations, on remarque que le taux de réussite est une fonction croissante en termes de nombre de traces.

Le modèle A semble meilleur que B. Intuitivement, on aurait pu s’attendre à des performances similaires, car le passage par la S-box est une fonction quasi-bijective. Ce n’est pas exactement vrai sur DES, puisque les S-box du DES ont une fan-in de 6 fils et un petit fan-out de 4 fils. Toutefois, elles sont conçues de telle sorte que si deux bits d’entrée sont fixes, la S-box devient bijective (en réponse à une exigence d’équilibre).

Par conséquent, la raison pour laquelle B est légèrement moins bonne que A est liée à la mise en œuvre de l’algorithme dans le circuit. En effet, dans la logique combinatoire du premier tour, B correspond à une fuite située plus loin que A. Ainsi B est davantage sujet à des changements intra-horloge de synchronisation dans les transitions et à l’activité des “glitches”² qui caractérise la logique combinatoire des CMOS.

Après cette discussion, on pourrait aussi s’attendre à ce que le modèle C soit meilleur que A, puisque le modèle C est en rapport avec l’activité d’un registre. Toutefois, cela est faux dans la pratique. Une explication possible est que le vecteur propre de A a plus de pics que ceux de C, et perçoit donc plus d’informations dispersées dans la trace.

La figure 5.6(a) montre que les modèles D et E sont de loin les meilleurs pour mener une attaque. En effet, peu de traces sont nécessaires pour que la probabilité de réussite de l’attaque atteigne rapidement 100%. Seulement 200 traces environ, permettent d’avoir une probabilité de 90% de réussite. Avec 250 traces, la probabilité s’élève à 100%. La conclusion est que la connaissance des transferts dans les registres dans un circuit peut améliorer considérablement l’attaque. Dans un circuit CMOS non protégés, ces modèles sont inégalés. Ces intuitions ont été déjà évoquées dans la littérature (par exemple dans [9] ou dans [76]), mais jamais démontrées. Grâce à cette expérience, nous avons en effet formellement démontré ce point : les distances entre les états consécutifs furent plus dans les circuits CMOS.

Finalement, nous tentons de comparer les modèles D (type-template [12]) et E (type-stochastique [71]). Pour être dans les mêmes conditions, nous calculons le taux de succès pour un certain nombre (commun) de traces utilisées dans chaque *template* ; En effet, 16/5 de traces en plus sont nécessaires pour le profilage de D. Le résultat, donné dans

²Ce terme correspond à une défaillance électrique d’un circuit, due à des re-convergences de chemins de longueur différentes et aussi car les portes logiques mettent en effet un certain temps pour réagir

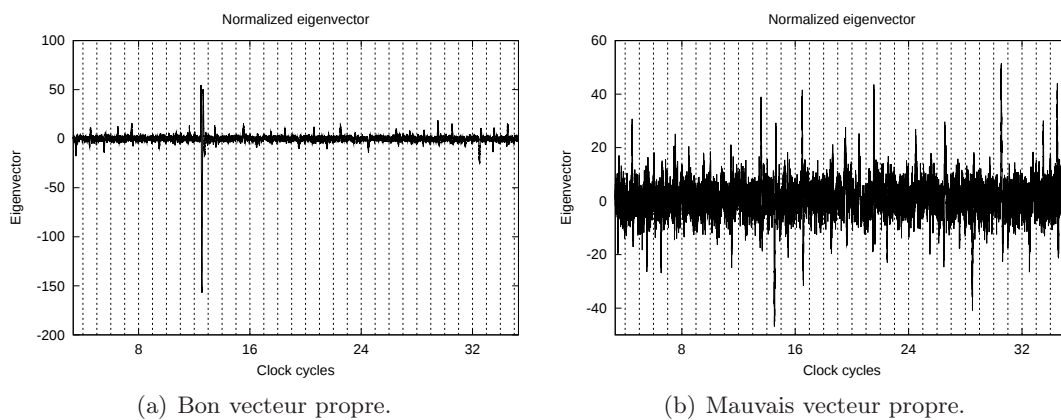


FIG. 5.3 – La différence entre un “bon” et un “mauvais” vecteur propre pour le modèle B.

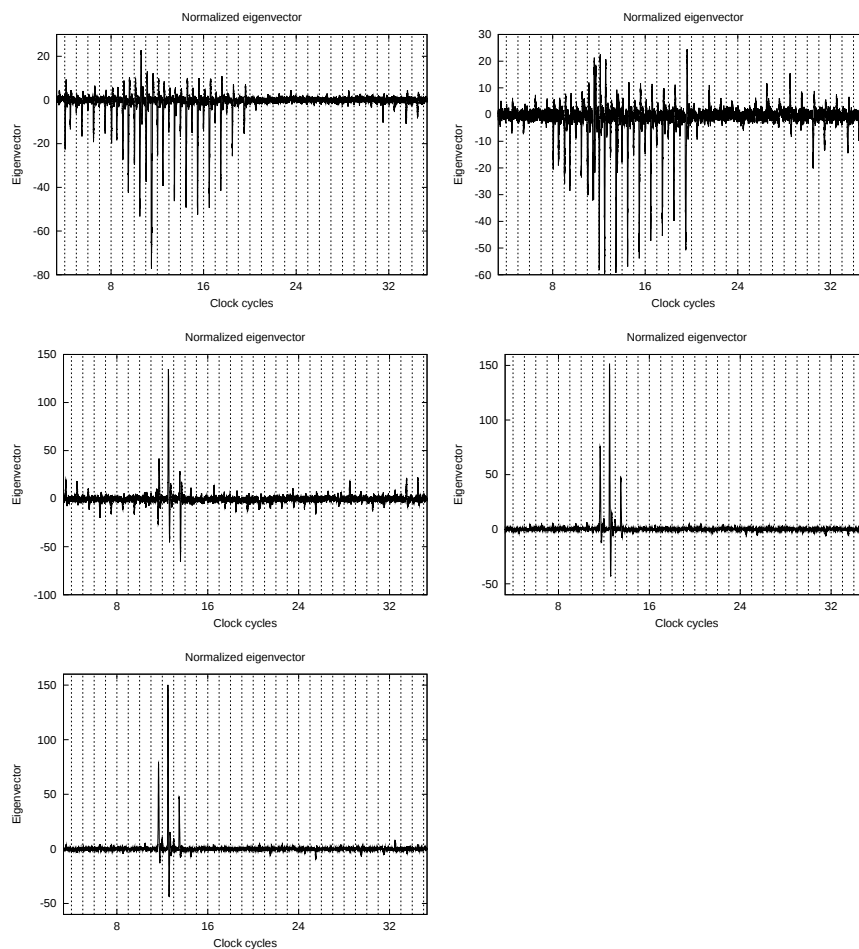


FIG. 5.4 – Les vecteurs propres, pour les modèles A, B, C, D & E (de la gauche vers la droite, du haut vers le bas).

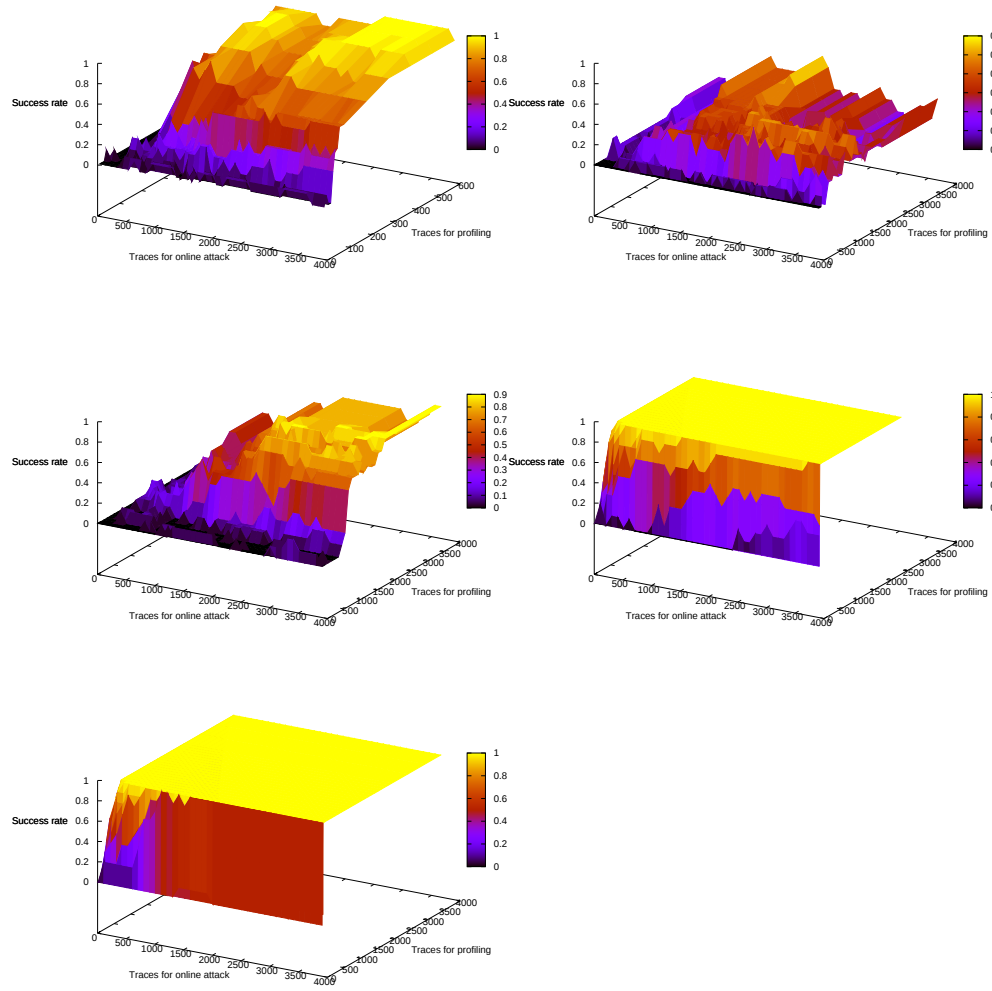


FIG. 5.5 – Taux de succès, pour les modèles A, B, C, D & E (de la gauche vers la droite, du haut vers le bas).

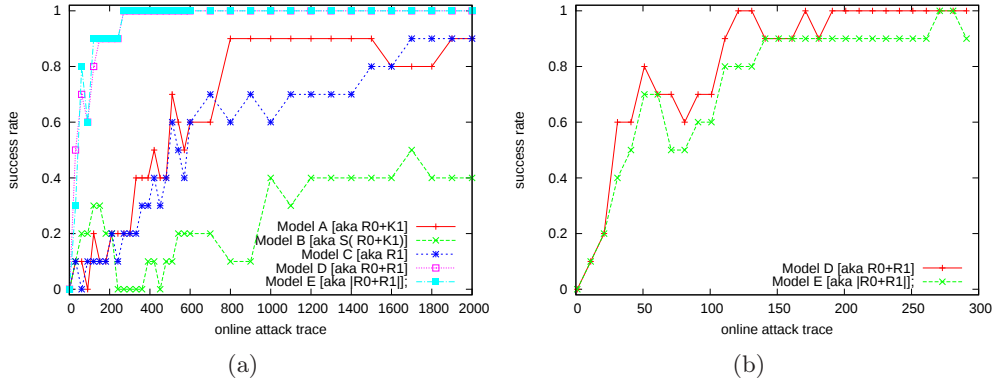


FIG. 5.6 – Comparaison du taux de succès, (a) entre tous les modèles pour le même nombre de traces profilage et (b) entre les modèles D et E pour le même nombre de traces de profilage par classe.

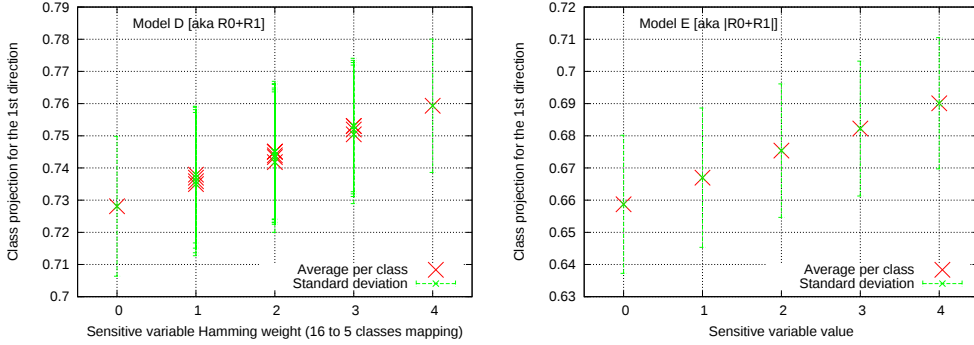


FIG. 5.7 – Comparaison entre les moyennes des modèles D et E en utilisant un seul vecteur propre.

la Figure 5.6(b) montre que D est meilleur que E. La raison en est que E est un modèle approximatif. La figure 5.7 montre en effet que la dégénérescence vers des classes identiques en poids de Hamming n'est pas parfaite dans la pratique ; Toutefois, pour un nombre donné de mesures dédiées au profilage, E est favorisé.

5.2.2 L'entropie conditionnelle

Comme le montre la figure 5.8(a), l'entropie conditionnelle diminue en fonction du nombre de traces utilisées dans la phase de profilage. Nous n'allons pas commenter le début des courbes, puisque pour un petit nombre de traces dans la phase de profilage, l'estimation d'entropie fournit des résultats erronés. En effet il n'y a pas assez de fuite. Cependant, nous remarquons que l'entropie conditionnelle tend vers une valeur limite.

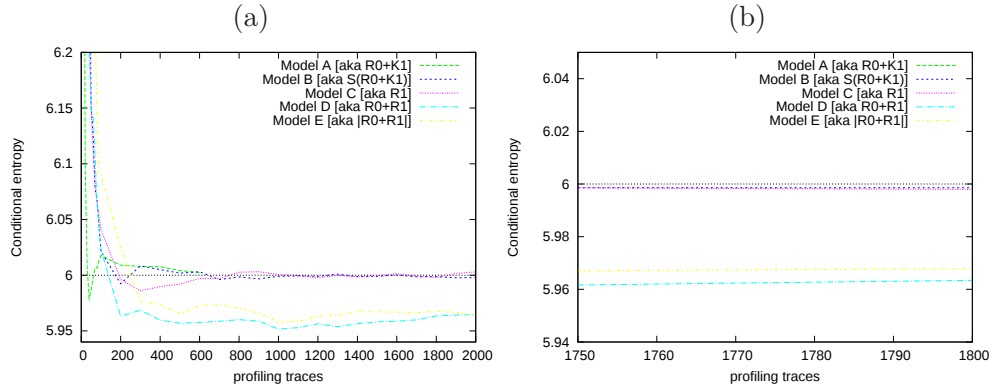


FIG. 5.8 – Comparaison des entropies conditionnelles (a) pour 2000 traces ; (b) zoom après convergence des estimations.

Comme prévu, cette valeur asymptotique est inférieure ou égale à 6 bits. En effet, les sous clés attaquées ont une entropie de 6 bits. Plus l'entropie conditionnelle pour un modèle choisi est basse, plus le circuit est vulnérable contre une attaque à l'aide de ce modèle.

Dans cette comparaison, nous avons veillé à avoir le même nombre de traces au cours de la phase de profilage. Mais ceci n'était pas possible pour le modèle A qui a besoin de plus des mesures puisqu'il a plus de classes (64) par rapport aux autres (16 ou 5 classes).

Par conséquent, le nombre de traces varie dans $[0,600]$ pour le modèle A et dans $[0,2000]$ pour les autres modèles. En comparant les différents modèles, il apparaît clairement que les modèles D et E sont plus favorables aux fuites par canaux auxiliaires. La figure 5.8(b) confirme qu'asymptotiquement, l'entropie conditionnelle pour les modèles D et E est plus petite que pour les autres modèles, et que D est meilleure que E pour une attaque. Par conséquent, le choix de la variable sensible est un élément important à prendre en compte dans une évaluation de la sécurité. Fondamentalement, nous observons que l'entropie conditionnelle classe les modèles dans le même ordre que le taux de succès.

Nous mettons en garde, qu'en théorie, le taux de succès et l'entropie conditionnelle ne sont certainement pas le même concept. Toutefois, dans cette étude, il semble qu'il y ait nécessairement une similitude entre eux. Cela est certainement dû au choix des modèles, et aussi à l'adéquation en termes de relation adversaire/sécurité. En effet, nous devons être conscients de certains pièges de l'interprétation, comme par exemple le risque que le circuit puisse sembler très bien protégé alors qu'en fait c'est l'attaquant qui n'est pas très astucieux.

Mais pour résumer, cette étude sur les différents modèles montre que l'entropie est en fait une bonne façon d'évaluer un circuit.

5.2.3 Modèles Invisibles

Soient $\mathcal{L}_1$ et $\mathcal{L}_2$ deux supposés modèles de fuite. Partons du principe que $\mathcal{L}_1$ est plus approprié que $\mathcal{L}_2$, si le taux de réussite d'une attaque en utilisant $\mathcal{L}_1$ est supérieur à $\mathcal{L}_2$. Ainsi dans le circuit SecMat que nous avons étudié, la distance entre les valeurs consécutives des variables sensibles qui sont dans les registres correspond à la fonction de fuite E. La découverte de ce meilleur modèle est toujours possible, à condition toutefois que l'attaquant ait une connaissance approfondie de l'architecture du circuit. Plus précisément, la valeur d'une variable sensible est accessible à toute personne en possession des spécifications de l'algorithme ; dans le cas contraire, la connaissance de la séquence de variables exige la mise en œuvre de l'algorithme au niveau du transfert du registre (RTL), comme par exemple sa description VHDL ou Verilog. Même en l'absence de ces éléments d'information, un attaquant peut toujours essayer de récupérer l'expression des variables sensibles (ou sa séquence) à partir des entrées ou sorties de l'algorithme, et ceci dépendra de l'attaque qui est soit à clair connu soit à chiffré connu. En effet, l'attaque agit comme un oracle : si elle ne parvient pas à récupérer la clé, cela signifie que la fonction de fuite n'est pas pertinente. Toutefois, dans les circuits, ces fonctions sont de degré algébrique élevé, notamment lorsque le niveau de pipeline suivant est un registre qui est un tour à l'intérieur de l'algorithme. Ce qui rend sa prédiction aléatoire, à moins d'avoir quelques indications sur les fonctions de fuites possibles. Quelques articles sur SCARE [58, 59] donnent un aperçu plus complet de ce problème.

Nous remarquons que les meilleurs modèles de fuite (les plus physiques) peuvent être connus par n'importe quel moyen. Le pipeline de l'algorithme peut être en effet très compliqué, intentionnellement ou non.

5.3 Amélioration des attaques grâce à la suppression du bruit par seuillage dans les profils de fuites

Comme déjà souligné dans la figure 5.3, un adversaire capable de comprendre la forme des vecteurs propres est plus susceptible de maîtriser le taux de succès de son attaque. De la même façon, un évaluateur de la sécurité peut exiger un vecteur propre idéal pour avoir une idée très claire de la sécurité de l'appareil expertisé.

Dans le cas d'un vecteur bruité, nous devons rechercher les meilleurs moments de fuite et aussi éliminer le bruit. Dans ce contexte, nous proposons d'améliorer le taux de succès ou le degré d'évaluation par la création d'un seuil $e \in [0, 1]$ appliqué au vecteurs propres. De manière générale, dans le cas où nous avons un nombre des traces infinies, le vecteur propre tend vers une courbe débruitée qui reflète parfaitement l'information temporelle de la fuite. Dans le cas de concrètes évaluations ou pour des scénarios d'attaques, nous sommes confrontés à des contraintes de temps et d'espace, et donc nous cherchons une meilleure façon d'affiner les vecteurs propres.

La figure 5.9(a) montre le gain important qui peut être attribué à un adversaire qui sait parfaitement exploiter les vecteurs propres.

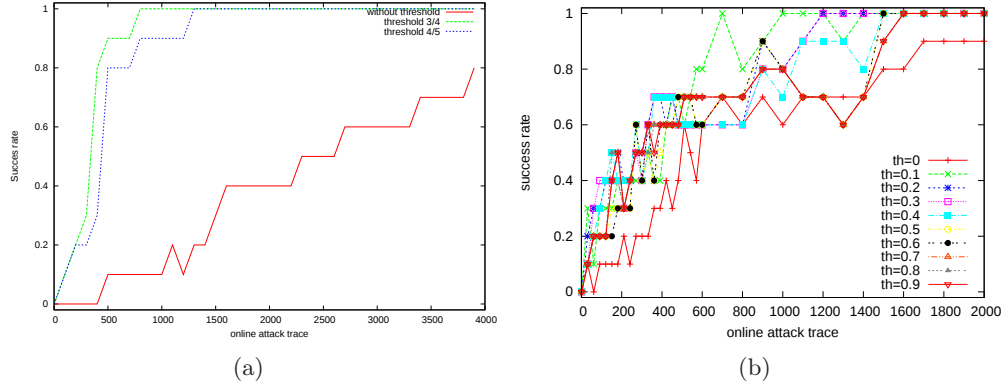


FIG. 5.9 – Comparaison du taux de succès (a) avec et sans seuillage pour le modèle A et (b) avec différents seuils pour le modèle C.

Cette expérience a été menée avec des traces bruitées différentes de celles utilisées dans les expérimentations précédentes, qui sont plus nettes. Le modèle choisi est le modèle A, qui montre une augmentation significative du taux de succès (environ 5 fois). Là, le seuil optimal th_{opt} est soit les trois quarts (3/4), soit les quatre cinquièmes (4/5) de la valeur maximale du vecteur propre.

En fait, nous voyons que pour un vecteur propre dont le bruit est éliminé, le taux de succès augmente.

Le choix du seuil est un compromis : quand il est trop élevé ($e \approx 1$), il peut y avoir un peu de bruit qui reste, alors que quand il est trop faible ($e \approx 0$), on peut éliminer des parties de l'information fournie par la phase de profilage. La figure 5.9(b) montre dans le cas du modèle C comment le taux de succès augmente en fonction du seuil choisi. Pour les modèles de la même famille (fondée sur la valeur plutôt que sur une distance), tels que A et C, on peut noter que l'adversaire peut se tromper sur le taux de succès, croyant qu'un modèle est meilleur qu'un autre.

D'autre part, si nous ne prenons pas de seuil, le modèle A paraîtrait meilleur que C. Toutefois, avec l'utilisation du seuillage, les conclusions sont inverses, comme représenté sur la figure 5.10.

De la même façon que pour le taux de succès, un évaluateur peut se tromper sur la sécurité d'un matériel. Un modèle peut sembler plus sûr qu'un autre, en particulier lorsque ces modèles sont liés. La figure 5.11 montre que la non-utilisation de seuillage nous conduit à l'erreur sur le modèle C, qui semble équivalent à A.

5.4 Discussion

Le but de cette section est de discuter de l'impact des techniques d'amélioration décrites ci-dessus dans le cadre théorique des analyses sur les canaux auxiliaires de [77].

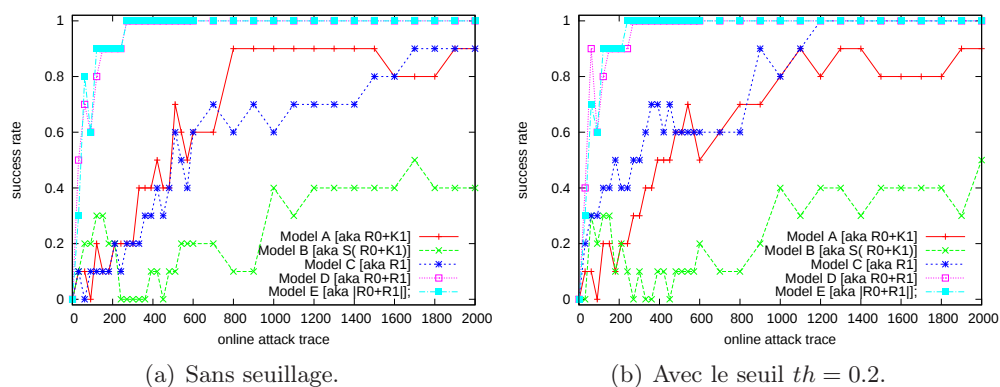


FIG. 5.10 – Amélioration du taux de succès avec le seuillage.

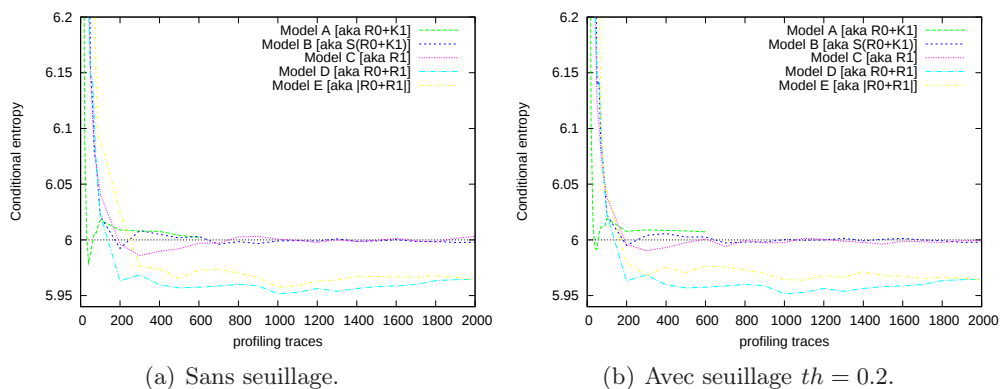


FIG. 5.11 – Amélioration de l'entropie conditionnelle avec le seuillage.

Fondamentalement, nous pensons que ces techniques peuvent amener à une évaluation erronée si elles présentent une dissymétrie plus avantageuse à l’attaquant au détriment de l’évaluateur. Pour illustrer cette argumentation, nous avons recours au même schéma que celui présenté dans [73]. L’évaluateur (noté E) calcule la position x alors que l’attaquant (noté A) calcule la position y . Ce type de diagramme peut être considéré comme un cas réel de résultats *à posteriori* en fonction des prédictions *à priori*.

Idéalement, il semble raisonnable que certains liens entre l’évaluation et les métriques sur les attaques soient mis en avant si celles-ci sont calculées sur le même circuit et avec les mêmes techniques de traitement du signal. Prenons l’exemple d’un matériel que nous avons l’intention d’examiner, dans le but de provoquer la fuite d’information. Typiquement, l’augmentation de la température T (ou sa tension d’alimentation) va augmenter sa consommation et donc la quantité de fuite. Dans ces expériences, nous nous attendons à ce que l’entropie et le taux de succès de l’attaque évoluent en parallèle. De plus, cette tendance devrait être indépendante du modèle de fuite choisi. L’attaquant n’a pas le choix de la température du circuit durant l’évaluation légale. Cette température était ce qu’elle était. Ces conclusions ne valent autant que la température de l’appareil pendant l’attaque est la même que celle pendant l’évaluation. Ceci est représenté dans la figure 5.12(a). Par conséquent, si l’évaluateur effectue l’évaluation à la température nominale, mais que l’attaquant est capable d’attaquer à des températures plus élevées ce dernier sera sans aucun doute favorisé. Ceci caractérise généralement une relation asymétrique entre l’attaquant et l’évaluateur ; nous pouvons également dire que l’attaquant triche ³ car il profite de degrés de libertés supplémentaires non disponible à l’évaluateur. Cette modification déséquilibre le rapport de force entre l’évaluation et l’attaque, comme le montre la figure 5.12(b). Dans des cas concrets, par exemple lorsque le circuit attaqué est une carte à puce inviolable, la température, la tension et toutes les conditions d’exploitation sont contrôlées. Par conséquent, *à priori*, ni l’évaluateur ni l’attaquant ne peuvent être favorisés.

Les contributions de ce chapitre montrent cependant comment créer une dissymétrie, même lorsque l’évaluateur et l’attaquant travaillent dans des conditions similaires (même mieux : exactement sur les mêmes traces [79]). La section 5.2 illustre une situation de dissymétrie dans les connaissances *à priori*, comme illustré dans la figure 5.13(a). Lorsque l’attaquant sait que $\mathcal{L}_2$ est plus approprié que $\mathcal{L}_1$, son attaque demandera en moyenne moins de traces pour retrouver la clé que celle de l’évaluateur qui se contente de $\mathcal{L}_1$. La section 5.3 montre l’effet d’une dissymétrie dans l’expertise sur les objets manipulés. Imaginez que pour un laboratoire, un CESTI⁴ par exemple, qui applique une attaque à partir des “manuels”, tels que présentées dans le chapitre 5.1. Ce CESTI court effectivement le risque de surestimation de la sécurité de sa cible d’évaluation s’il n’est pas au courant de la technique du “seuillage” par exemple.

En résumé, nous pensons que l’évaluateur peut se tromper en étant trop confiant sur la sécurité d’un circuit, ou ne pas prévoir une faiblesse dans le modèle de fuite ou

³Nous voulons dire que l’attaquant utilise une stratégie en dehors du modèle de sécurité.

⁴Centre d’Évaluation de la Sécurité des Technologies de l’Information

dans les étapes de traitement d'attaque. Ainsi les routines d'évaluation peuvent être nuisibles à la confiance qui pourra être accordée au circuit. Ce fait est illustré dans la figure 5.14. Si l'évaluateur a évalué de nombreux composants cryptographiques au même niveau commun d'évaluation de sécurité [16]), il limiterait son analyse à la mesure d'une métrique de la théorie de l'information. Toutefois, si un attaquant arrive avec plus de connaissances sur le matériel que l'évaluateur ou avec de meilleures techniques de traitement du signal, les prévisions du CESTI peuvent se révéler trop optimistes, puisque traditionnelles (considérant à tort que "tout le monde fait comme lui").

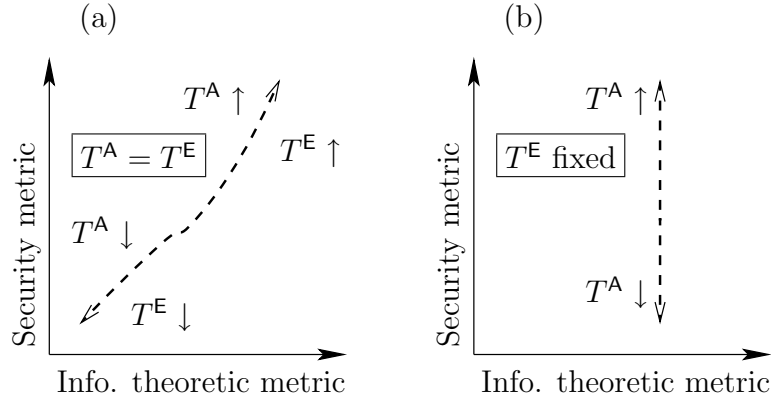


FIG. 5.12 – Évaluation *vs* Diagramme d'attaque dans une situation (a) symétrique et (b) asymétrique.

Ces techniques d'amélioration de l'attaque n'invalident pas le cadre théorique de [77], mais simplement avertissent que les notions qu'il introduit doivent être manipulées avec précaution. Sinon, des prédictions incorrectes peuvent être faites, ruinant ainsi la confiance qu'on peut avoir en ce cadre.

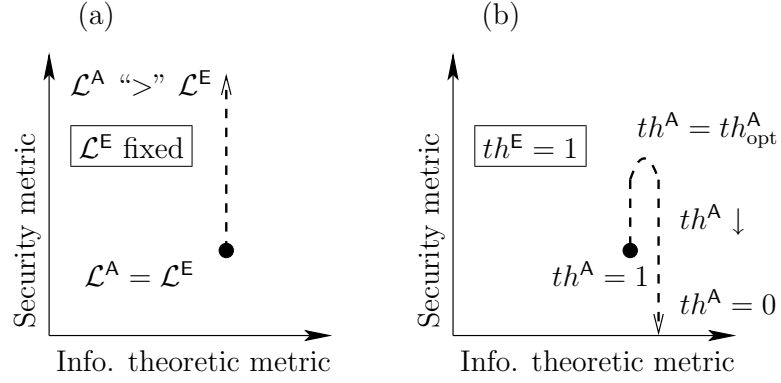


FIG. 5.13 – Fuite *vs* diagramme d’attaque pour les deux optimisations : (a) Modèle de fuite adéquat, (b) seuillage pour des échantillons non-informatifs.

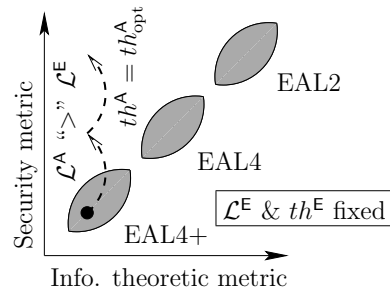


FIG. 5.14 – Les risques de confiance en d’anciennes expériences et en s’appuyant sur les évaluations formelles *vs* attaques sans fondement des lois empiriques.

Chapitre 6

Quelles attaques combinées ?

Nous avons vu, jusque là, que la plupart des attaques physiques commencent par tenter de partitionner les mesures acquises durant les chiffrements. Ces partitions sont indexées par des hypothèses sur les clés [75]. Ensuite, l'adversaire évalue la qualité de chaque partitionnement. Cette information est généralement résumée par un **facteur de mérite** qui peut être :

une différence de moyennes : DPA[38],

une corrélation : CPA [9],

une vraisemblance : attaques *template* [12]) ou

une information mutuelle : MIA [23],

pour ne citer que les attaques les plus répandues. Ces facteurs de mérite sont souvent appelés *distingueurs*, car ils réussissent à établir une distinction entre les clés candidates pour ne sélectionner que la clé utilisée durant le chiffrement. La comparaison de ces *distingueurs* sur les acquisitions est discutée dans [18, 49], [41, 24, 82] et dans le chapitre 5 de cette thèse. La fonction d'un *distingueur* est de donner une note à une façon de partitionner les traces, c'est-à-dire une hypothèse sur la clé. Cette note étant un nombre, il existe une relation d'ordre permettant de classer les dites hypothèses sur la clé. Si la vraie valeur de la clé est connue, les *distingueurs* peuvent être évalués selon les critères suivants :

- proportion de cas où le *distingueur* place la bonne clé en première position ou
- rang moyen de la bonne clé [24].

En outre, les conclusions dépendent de la cible, car la structure de fuite est inhérente à chaque dispositif. La construction de nouveaux *distingueurs* est un domaine de recherche très actif. En plus, tous les nouveaux *distingueurs* contribuent à alimenter une batterie d'attaques pouvant être lancées en parallèle contre un dispositif donné.

Une voie de recherche consiste à tenter de tirer le meilleur parti des *distingueurs* existants, selon les options suivantes :

- combinaison constructive d'attaques connues sur des traces de fuites communes
 - ou
-

- combinaison de différents échantillons de traces ou encore de différentes traces acquises de façon concomitante.

Ainsi différents types de combinaisons peuvent être envisagés. La plupart de ces combinaisons sont des problèmes qui, à notre connaissance, ne sont pas encore étudiés :

1. Divers distingueurs peuvent être combinés pour un même partitionnement. Quelques tentatives ont été faites dans la première édition du concours DPACONTEST [79] (contribution de Jung Hae-il) mais on ne sait pas encore comment différents distingueurs peuvent se renforcer mutuellement à partir d'un ensemble commun de données de fuites.
2. Un distingueur peut être évalué sur divers partitionnements ; il a déjà été observé dans le chapitre 5 qu'un même dispositif non protégé pourrait fuir différemment pour les différents partitionnements [19]. Dans ce chapitre, il est prouvé qu'il y a des circuits dont les fuites sont statistiquement indépendantes. Par conséquent, il serait opportun de les combiner car le résultat de l'attaque sera certainement amélioré par l'utilisation simultanée de plusieurs modèles.
3. La diversité peut également provenir de la multiplicité des échantillons généralement recueillis lors d'une campagne d'acquisition : l'attaque MMIA [22] exploite deux différents échantillons dans des traces de fuite dans le but de faire échouer une contre-mesure basée sur le masquage. L'idée est que la distribution conjointe de ces deux échantillons dépend du secret donc que la probabilité conjointe ou l'information mutuelle est un distingueur.
4. Ceci peut également résulter d'acquisitions multi-modales : une telle configuration peut également consister en plusieurs canaux auxiliaires. [2] explique la meilleure façon de combiner plusieurs canaux. Par exemple, les canaux peuvent être le résultat de la démodulation des signaux électromagnétiques sur plusieurs fréquences. Les acquisitions peuvent également être menées en parallèle en fonction de différents capteurs tels que, par exemple, la consommation en courant et le champ électromagnétique, comme suggéré dans [74]. Les multiples canaux peuvent également provenir de deux capteurs identiques mais qui enregistrent les émanations provenant de deux différents endroits sur la puce, ce qui est réalisable sur des circuits « VLSI ». La meilleure localisation des deux capteurs peut être déterminée en procédant à une cartographie [64] de l'appareil attaqué.

Une étude préliminaire de Thanh Ha Le ([40], chapitre 4) conclut qu'il y a peu d'avantages à gagner avec des acquisitions multiples en utilisant la même modalité, à savoir le champ magnétique, capturées à partir de différents endroits.

La même conclusion est tirée dans [43] par l'utilisation de l'analyse en composantes indépendantes (ACI), qui conduit à des matrices mal conditionnées et, par conséquent, à des systèmes d'équations numériquement insolubles.

Néanmoins, cette étude de cas a ciblé une carte à puce, qui est quasi-ponctuelle en ce qui concerne les longueurs d'onde d'intérêt¹. Des circuits plus grands, tels que

¹En effet, les longueurs d'onde d'intérêt sont supérieures à $c/f_{\max} \approx 5$ mm pour une chaîne d'acquisitions de bande passante $[0, f_{\max} = 3]$ GHz.

les FPGAs ou ASICs sur les PCBs², pourrait relancer l'intérêt de telles méthodes d'interférences constructives.

5. Il peut y avoir des situations où le partitionnement le plus approprié peut évoluer d'un échantillon à un autre sur une trace de fuite. Ce cinquième problème est lié au troisième : le modèle de fuite dépend de l'échantillonnage temporel. Cependant, nous envisageons dans cette partie des situations où la différence n'est pas artificiellement due à une contre-mesure mais provient de la distorsion au niveau de la communication entre la fuite dans le dispositif et le capteur du canal auxiliaire. Ce comportement est prévu par exemple dans les ondes électromagnétiques produites par un ECB. Nous illustrons cette situation sur un exemple concret où la fuite peut évoluer au cours du cryptage. Comme dans le cas de la deuxième question, nous soulignons qu'une recherche initiale des points d'intérêt en utilisant des méthodes générales nous aurait conduite à négliger la richesse de ce comportement. Cela rend ce problème d'autant plus intéressant du point de vue de caractérisation.
6. Pour être totalement exhaustif, mentionnons que d'autres types de combinaisons ont déjà été étudiés. Par exemple, [3] et [14] décrivent la combinaison de deux constructions d'attaques, à savoir une analyse passive (observation) et une analyse active (perturbation).

Nous proposons dans ce chapitre deux études sur les combinaisons :

- nous combinons des partitionnements dans le cadre d'attaques templates et
- nous combinons différents échantillons dans le cas d'une attaque CPA.

6.1 Attaques combinées et métriques basées sur des partitionnements multiples

Nous explorons dans cette section la combinaison de différents partitionnements sur les attaques *template*. En effet, quelques "comparaisons" d'attaques qui nécessitent un modèle physique de fuite échouent si la fonction de fuite ne correspond pas suffisamment à la modalité de fuite du circuit.

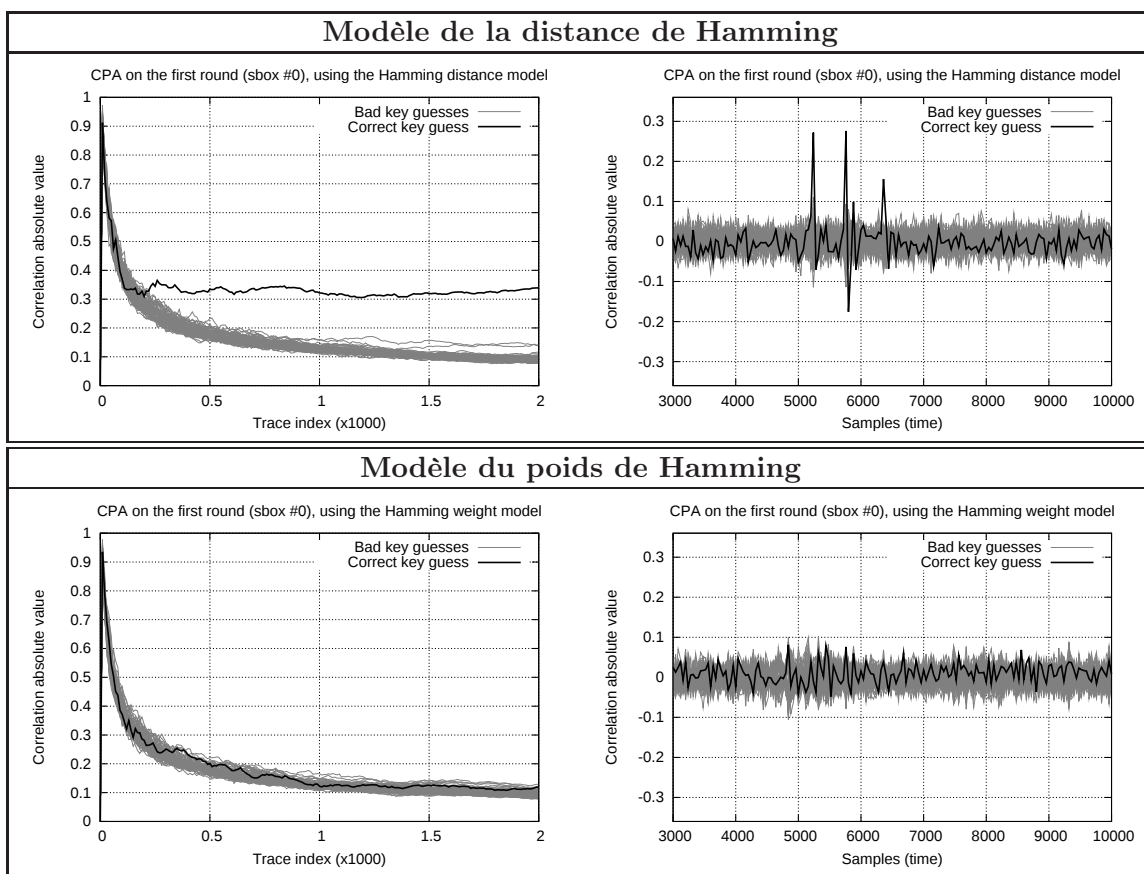
Par exemple, nous illustrons dans le tableau 6.1 le fait que :

- une CPA [9] réussit à récupérer la clé secrète sur un processeur DES non protégé si le modèle avec lequel les traces sont corrélées est la distance de Hamming. Ce qui est physiquement pertinent pour le dispositif en cours d'analyse ;
- une CPA avec une fonction de fuite moins adaptée, à savoir le poids de Hamming à la sortie de la boîte de substitution, échoue.

Cependant, nous remarquons que même si la bonne hypothèse de clé ne donne pas la plus grande corrélation avec le poids de Hamming, la forme du signal de corrélation correspondant à la bonne hypothèse sur la clé est différente de celles correspondants aux hypothèses incorrectes. Par conséquent, une attaque par la théorie de l'information serait appropriée pour aider à mieux distinguer entre les hypothèses.

²Abréviation de "Printed Circuit Board", c'est un support permettant de relier électriquement un

TAB. 6.1 – L'évolution des corrélations et le signal final après une estimation sur 2.000 traces.



6.1.1 Les variables sensibles et les attaques *template*

Dans le chapitre 5, notre étude consistait à choisir les meilleures variables sensibles adaptées pour un adversaire ciblant des traces, qui sont publiquement disponibles [79]. À Partir d’une comparaison entre les cinq différents modèles, il est démontré que le modèle le plus approprié pour le circuit ciblé est la distance de Hamming entre deux registres. Cependant, les “attaques par partitionnement” (dans le sens de [75]) sur différentes valeurs sensibles (telles que les entrées des fonctions linéaires et non linéaires) permettent également à un adversaire de récupérer la clé, mais avec beaucoup plus de traces. La connaissance de l’architecture du circuit fournit certainement beaucoup plus de détails sur la fonction de fuite. Dans ce chapitre, nous combinons ces modèles afin de récupérer la clé avec moins de traces, et observons le comportement de l’entropie en fonction du nombre de vecteurs propres retenus dans l’attaque.

Nous utilisons le même schéma d’attaque à savoir une phase de profilage qui va nous permettre de caractériser la fuite, et une phase d’attaque utilisée aussi dans la métrique estimant la force de l’attaquant.

L’analyse par composante principale a été utilisée pour réduire les dimensions des échantillons tout en gardant le maximum d’information.

6.1.1.1 Les modèles combinés

L’objectif est de combiner deux partitionnements. La sécurité du modèle composé résultant est évaluée par les attaques *template* ; De manière similaire, la robustesse du circuit est mesurée en tenant compte de ce nouveau modèle. Un adversaire qui combine des modèles peut-il être considéré comme un adversaire d’*ordre supérieur* [53] ? Est-il capable de récupérer la clé secrète plus rapidement ? L’expérience décrite dans cette section tente de répondre à ces questions. Considérons,

1. **le Modèle M1** correspondant à la valeur de sortie de la première S-box du premier tour. C’est un modèle sur 4-bit, et
2. **le Modèle M2** impliquant la transition du premier bit du modèle M1. Il s’agit d’un modèle mono-bit, appartenant à la catégorie générale des modèles de la “distance de Hamming”.

De ces deux modèles, on établit un troisième nommé **Modèle M3**. M3 combine le modèle de 4-bit M1 et le modèle M2 sur 1-bit. En d’autres termes, M3 est considéré comme une structure de bits où la valeur du bit le plus significatif (MSB) est le modèle M2. Les 4 autres bits correspondent au modèle M1. On peut voir aussi, M3 comme la concaténation de M1 et M2, et nous notons $M3 \doteq (M1, M2)$. Par conséquent M3 est un modèle sur $4 + 1 = 5$ bits, ce qui signifie que M3 sera basé sur 32 partitions. Autrement dit, le partitionnement de M3 est égal au produit cartésien de M1 et M2.

Une comparaison équitable entre ces modèles n’est pas une opération triviale.

ensemble de composants électroniques pour réaliser un circuit électronique complexe.

De manière générale, le nombre de *templates* pour les modèles M1, M2 et M3 est différent. Essentiellement, en ce qui concerne la phase d'entraînement (*c.-à-d.* la construction des *templates*) :

1. soit l'adversaire a un nombre égal de traces par classe.
2. soit l'adversaire dispose d'un nombre fixe de trace pour l'ensemble des classes.

Le choix va influencer le taux de succès comme nous le verrons dans la suite. Le premier cas est le plus réaliste : il consiste à dire que le temps de pré-caractérisation est presque illimité mais que les traces prises en ligne à partir de l'appareil attaqué sont limitées. Nous modélisons cette situation en prenant le même nombre de traces pour chaque partition. Par conséquent, au total, beaucoup moins de traces d'entraînement sont utilisées pour les modèles mono-bits ; mais cela représente t-il vraiment le cas où les modèles sont évalués dans des conditions identiques ? La seconde possibilité tient compte du cas où le coût de pré-caractérisation est non-négligeable. Selon cette hypothèse, l'avantage des attaques combinées est moins clair, puisque le nombre de traces permettant d'estimer chaque modèle est réduit. Ainsi, lors d'une attaque sur un modèle non combiné, la précision des *templates* compensera certainement la perte de profit de la combinaison.

6.1.1.2 Premier Choix : Évaluation avec attaque limitée.

Pour les besoins de cette évaluation, nous utilisons un nombre égal de traces par classe. Dans cette expérience, on prend 1.000 traces par classe pour les modèles **M1**, **M2** et **M3**. La comparaison est faite avec et sans l'utilisation de la méthode de seuillage présentée dans le chapitre 5. La figure 6.1 rappelle la méthode dans un schéma simplifié dans le cas de combinaison. Le principe du seuillage est de considérer qu'un échantillon de valeur inférieur à un seuil choisi tendrait vers zéro si on utilisait plus de traces donc il est négligeable. Le seuillage est important dans cette situation de combinaison, car un partitionnement avec beaucoup de bruit risque d'introduire artificiellement des erreurs. Idéalement, il faut faire un seuil pour chaque partie de la combinaison. Il y a bien sûr un compromis dans le choix du meilleur seuil possible. Un seuil trop petit garde un trop grand nombre d'échantillons non pertinents, tandis qu'un seuil trop grand filtre certains points d'intérêts qui peuvent apporter de l'information. Pour la mise en œuvre étudiée dans cette section, nous avons constaté que le choix d'une valeur de 40 % est un compromis équitable.

La figure 6.2 montre le taux de succès des attaques *template* suivant les trois modèles. Nous rappelons que plus le taux de succès est grand, meilleure est l'attaque. Nous remarquons sur la figure 6.2 que dans le cas de la non-application du seuillage, l'attaque *template* basée sur le modèle combiné est meilleure que celle sur les autres modèles. Le modèle combiné est nettement meilleur que le modèle **M1**, et légèrement meilleur que le modèle **M2**.

Par ailleurs, lorsque nous avons recours à un seuil, le modèle M2 et M3 sont équivalents et, évidemment, toujours meilleurs que M1.

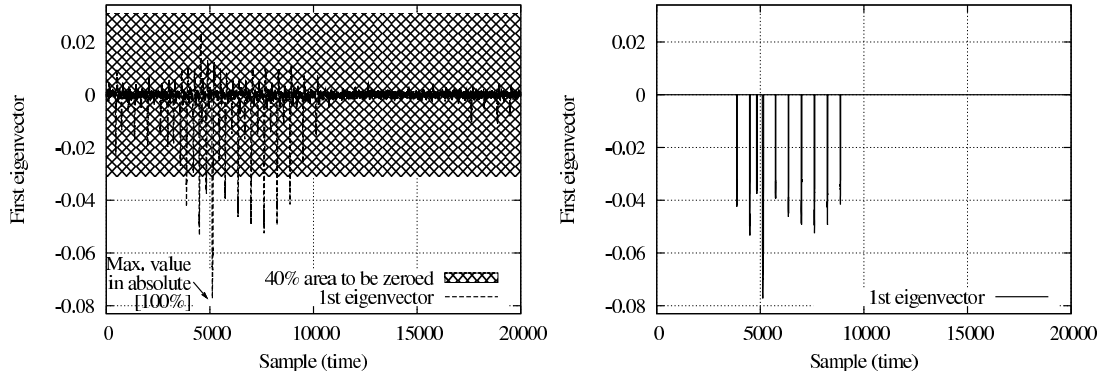


FIG. 6.1 – Premier vecteur propre sans seuillage (*gauche*), et le même avec un niveau de seuillage à 40% (*droite*).

Notre comparaison modélise d’une manière moins appropriée la fonction de fuite, puisque seul le premier vecteur propre de l’analyse par composante principale est utilisé dans la comparaison entre les modèles M2 et M3. En effet, d’autres vecteurs propres parmi les 31 possibles dans le cas du modèle combiné M3 contiennent également des informations, tandis que le modèle M2 a seulement une unique direction significative.

6.1.1.3 Second Choix : Évaluation avec entraînement limité.

Si on utilise la première option, on prend 32.000 traces en général. Ainsi, pour un nombre constant de traces par classe, nous avons $32.000/16=2.000$ traces par classe pour le modèle **M1** et $32.000/2=16.000$ traces par classe pour **M2**. Le modèle combiné **M3** correspond donc à un nombre de $32.000/32 = 1.000$ traces par la classe. Dans ce second cas, nous utilisons systématiquement 32.000 pour la formation de tous les modèles M1, M2 et M3. En conséquence, le modèle M2, qui a le plus petit nombre de partitions, est évalué avec plus de précision que M1 et M3.

Les deux schémas dans la figure 6.3 montrent que la combinaison des modèles n’améliore pas beaucoup l’attaque. En effet, le taux de succès du modèle M3 reste très proche du taux de succès du modèle M1.

6.1.1.4 L’entropie conditionnelle

L’entropie conditionnelle donne une idée sur la robustesse du circuit, indépendamment de toute attaque. La valeur de l’entropie conditionnelle tend vers une valeur limite en fonction du nombre de traces utilisées durant le profilage. Pour cette expérience, nous avons pris un grand nombre de traces au cours de la phase de profilage afin d’avoir une approximation de cette valeur limite. Cela nous permettra de comparer la robustesse du circuit contre des attaques en utilisant les modèles M1, M2 ou M3. Notre circuit est-il

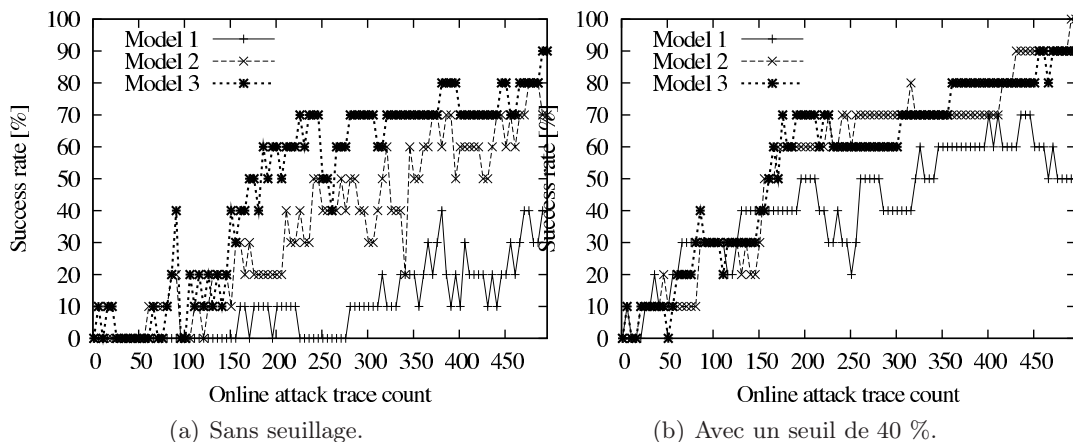


FIG. 6.2 – Comparaison du taux de succès entre le modèle M1, M2 et le modèle combiné M3 pour deux différents seuils et 1000 traces par classe.

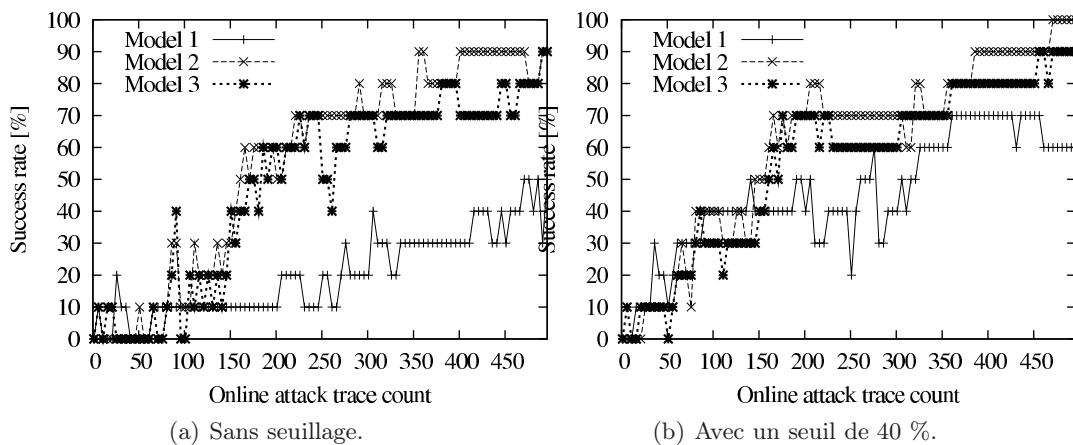


FIG. 6.3 – Comparaison du taux de succès entre les modèles M1, M2, et le modèle combiné M3 pour deux différents seuils et 32.000 traces en général pour l'entraînement (à diviser respectivement entre 16, 2 et 32 classes).

plus vulnérable face à un attaquant qui combine les modèles ? La figure 6.4 tente de répondre à cette question.

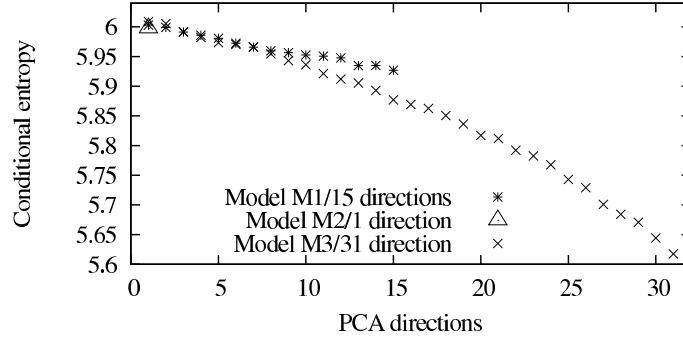


FIG. 6.4 – Comparaison de l'entropie conditionnelle entre les différents modèles.

Nous savons que le nombre de vecteur propres de l'ACP dépend de la cardinalité de la variable sensible. Par exemple, dans cette étude, nous avons 15 directions pour le modèle M1, 1 direction pour le modèle M2, et 31 directions pour modèle M3. La première direction résume un grand pourcentage de la variance des données. Une comparaison de la robustesse en utilisant uniquement cette première direction peut paraître satisfaisante, mais cette étude montre que plus on prend de directions, meilleure est l'estimation de la fuite donc plus petite l'entropie conditionnelle. Les modèles combinés sont donc l'occasion de découvrir de nouveaux modes de fuite, comme déjà indiqué pour la combinaison de multiples canaux cachés (puissance + EM) [74]. Ceci ajoute en fait un autre avertissement pour les évaluateurs de sécurité : la robustesse d'une mise en œuvre peut être surestimée si les modèles sont incomplets ou s'ils contiennent trop peu de partitions.

Dans cette section nous avons :

- combiné deux modèles dans le cas d'attaques *template*,
- illustré les difficultés rencontrés pour une comparaison entre les modèles singulier et le modèle combiné et
- mis en avant l'importance des composantes principales dans l'évaluation de la robustesse.

Dans la section suivante, nous nous concentrons sur la combinaison des points d'intérêt dans la cadre de l'analyse des corrélations.

6.2 Attaques par corrélation combinée

Pour améliorer les analyses sur les canaux cachés en présence d'importante quantité de bruit, la difficulté première consiste à identifier les échantillons qui fuient. Ce sont les *points d'intérêts POI*³ que nous avons commencé à présenter dans la section 3.4.1.

³Abrégé du terme anglais Points Of Interest

Ceux-ci correspondent à des moments où les données sensibles sont effectivement traitées et où il y a le plus de fuite. Comme déjà mentionné dans la section précédente lors de l'examen de la méthode de seuillage, le processus de sélection des POI implique un compromis : plus de points sont sélectionnés, plus d'information est recueillie mais aussi plus de bruit. La tâche difficile consiste à séparer le signal du bruit.

6.2.1 Des techniques pour retrouver les POIs

Plusieurs techniques ont été proposées pour identifier les points d'intérêt :

- la *somme des carrés des écarts* (ou SOSD dans [23] et SOST dans [25]),
- l'Information Mutuelle (IM [47]),
- l'*Analyse en Composantes Principales* (ACP [4]) et
- l'*Analyse Discriminante Linéaire* (ADL [74])

sont des exemples très répandus. Dans cette section, nous étudions ces méthodes et comparons leurs efficacités, en les appliquant sur deux séries de mesures, l'une à courte distance du circuit et une autre, avec plus de bruit, à 25cm du circuit. Pour ces expériences, nous avons utilisé une carte Sasebo-G [69] incorporant une mise en œuvre matérielle d'AES. Pour ces deux séries de mesures électromagnétiques $\mathbf{O}(t)$ nous remarquons que la CPA 1.5.3 peut être réalisée avec succès, en utilisant le modèle de la distance de Hamming entre l'avant-dernier et le dernier tour de l'AES.

6.2.1.1 Le sosd vs sost vs IM.

Le calcul de la métrique indicatrice de fuite sosd exige le calcul de la moyenne des traces dans un partitionnement donné. Dans la proposition initiale de [23], le partitionnement concerne les 256 valeurs d'un octet AES. La mise en œuvre sur Sasebo-G est connue pour des fuites en distance de Hamming entre l'avant-dernier et le dernier tour. En effet, nous réussissons la CPA pour les deux jeux de mesures dans ce modèle. Par conséquent, nous décidons de restreindre les valeurs de la fuite à l'*intervalle* $[0, 8]$, selon $\mathcal{L} = HW(\text{état}_9[\text{sbox}] \oplus \text{chiffré}[\text{sbox}])$, où $\text{sbox} \in [0, 16[$ est l'indice de la boîte de substitution. Si l'on note $o_i(t)$ les échantillons (t) de la $i^{\text{ème}}$ réalisation de l'observation $\mathbf{O}(t)$, alors les moyennes $\mu_j(t)$ dans chaque classe $j \in [0, 8]$ sont données par la moyenne de l'ensemble $\{o_i(t) \mid l_i = j\}$. Ensuite, leurs écarts quadratiques sont additionnés pour donner le sosd.

La sost est basée sur le T-Test, qui est un outil statistique standard pour distinguer les signaux bruyants. Cette méthode a l'avantage de considérer non seulement la différence entre les moyennes $\mu_j, \mu_{j'}$, mais aussi leur variance $(\sigma_j^2, \sigma_{j'}^2)$ par rapport au nombre d'échantillons $(n_j, n_{j'})$. Les définitions mathématiques du sosd et du sost sont données ci-dessous :

$$\text{sosd} \doteq \sum_{j,j'=0}^8 (\mu_j - \mu_{j'})^2 \quad \text{et} \quad \text{sost} \doteq \sum_{j,j'=0}^8 \left(\frac{\mu_j - \mu_{j'}}{\sqrt{\frac{\sigma_j^2}{n_j} + \frac{\sigma_{j'}^2}{n_{j'}}}} \right)^2.$$

La figure 6.5 montre les *sosd* et *sost* correspondants aux deux campagnes d'acquisitions électromagnétiques. Nous notons que, pour comparer la trace de corrélation, les courbes *sosd* et *sost* sont associés aussi à une mesure à 0 cm. Mais, bien que nous utilisions pour le partitionnement la même fonction de fuite $\mathcal{L}$ et bien que nous trouvons la bonne clé en utilisant une CPA sur la mesure à 25 cm, la courbe *sosd* ne met pas en évidence le bon échantillon, *c.-à-d.* le moment où la clé peut être récupérée par la CPA. La figure 6.5 montre que la métrique *sosd* n'est pas toujours une métrique efficace pour révéler les points d'intérêt. En effet, nous avons essayé d'exécuter la CPA sur les échantillons mis en évidence, mais toutes ces attaques ont échoué. En ce qui concerne la *sost* sur les mesures à 25 cm, plusieurs POIs sont révélés mais qui ne sont pas liés à des données secrètes. Ainsi *sost* n'est pas non plus un outil fiable pour identifier les points d'intérêt.

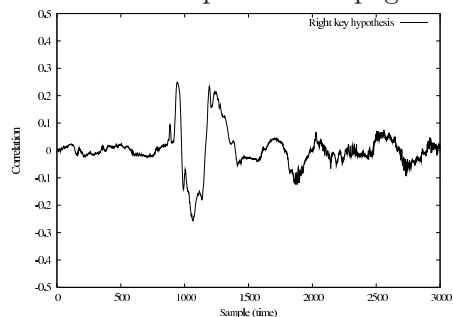
En ce qui concerne l'IM, également représentée sur la figure 6.5, on voit que les pics s'harmonisent bien avec le *sost* à de courtes distances, mais elle donne des pics sans informations (notamment les échantillons 441 et 975). Il n'est donc pas un outil fiable. La raison principale est que les densités de probabilité sont mal estimées en présence de grandes quantités de bruit.

6.2.1.2 L'ACP à distance.

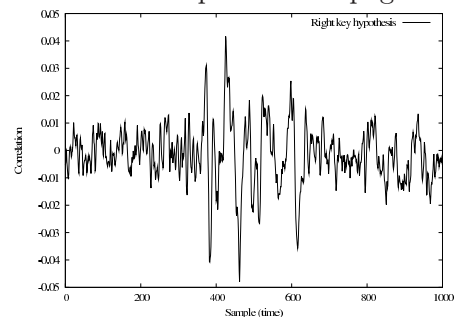
Comme expliqué précédemment dans la section 3.5.1, l'ACP vise à fournir une nouvelle description des mesures par projection sur le vecteur propre le plus important de la matrice de covariance empirique. Si l'on compare le taux de succès de la CPA, appliquée après une ACP, on peut remarquer que dans le cas de la campagne à distance, avec un niveau élevé de bruit, le vecteur propre correspondant à la plus grande valeur propre n'est pas nécessairement approprié. Le taux de succès de la CPA après une projection sur chacun des neuf vecteurs propres est donné dans les figures 6.6 et 6.7. À 25 cm, nous remarquons que la projection sur le premier vecteur propre n'est pas nécessairement la plus appropriée, car ce vecteur ne donne pas le meilleur taux de succès de l'attaque. D'une façon tout à fait surprenante la projection sur le troisième vecteur propre s'avère la plus efficace. À l'inverse, lorsque le niveau de bruit est faible et la sonde électromagnétique est positionnée à courte distance, la projection sur le premier vecteur est la plus efficace.

Ce phénomène peut être expliqué par le fait que le nombre de courbes dans les sous-ensembles correspondants à la distance de Hamming 0 et 8 sont dans la même proportion, néanmoins le niveau de bruit est plus élevé, car ils contiennent le moins de

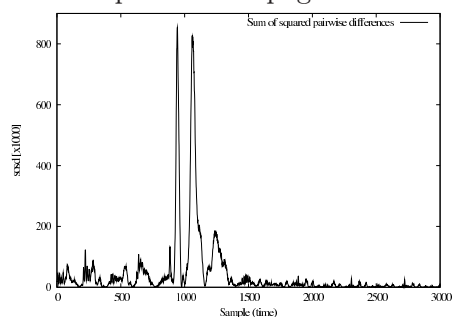
Trace de corrélation pour la campagne à 0 cm



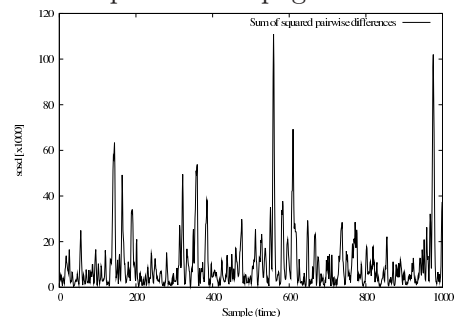
Trace de corrélation pour la campagne à 25 cm



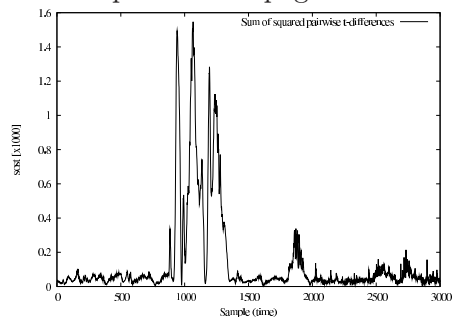
sosd pour la campagne 0 cm



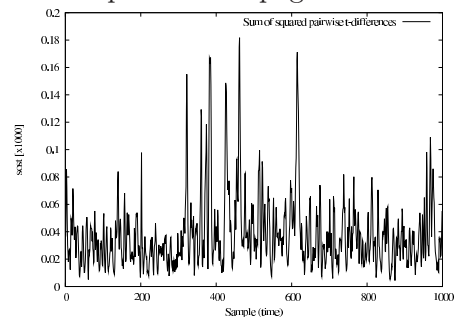
sosd pour la campagne 25 cm



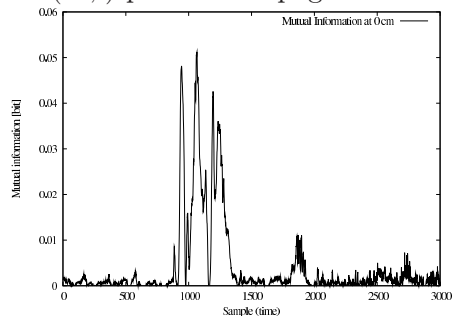
sost pour la campagne 0 cm



sost pour la campagne 25 cm



I(O;l) pour la campagne 0 cm



I(O;l) pour la campagne 25 cm

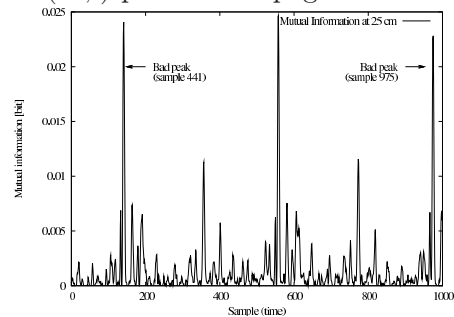


FIG. 6.5 – Les traces de corrélation, sosd, sost et IM obtenues pour les bonnes hypothèses de clés.

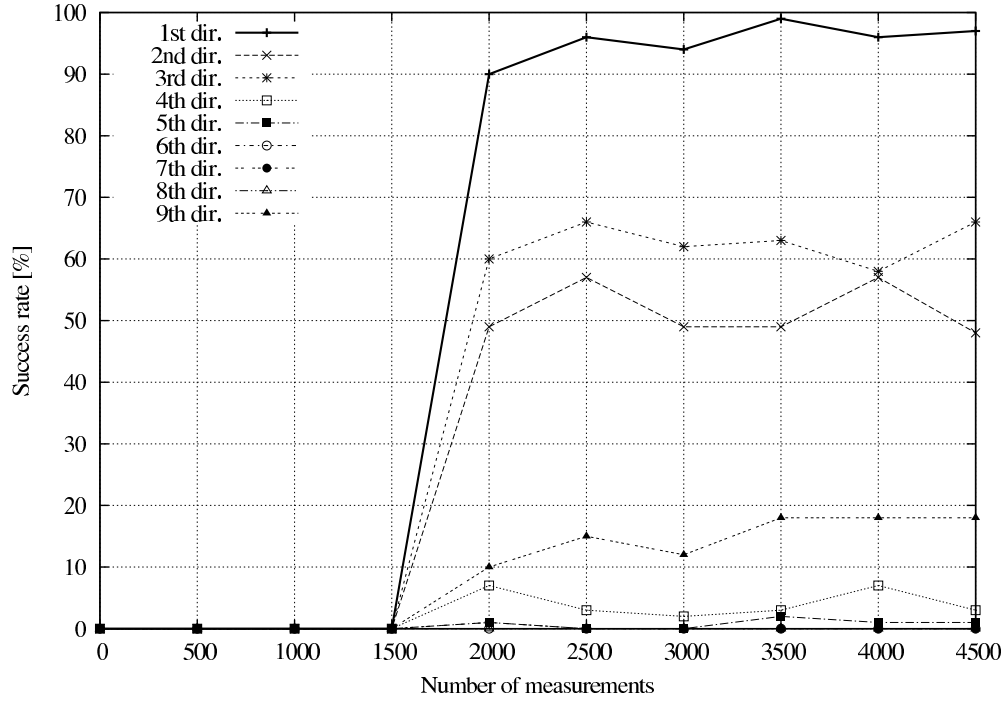


FIG. 6.6 – Taux de succès de la CPA après ACP à 0 cm.

traces. En effet, la proportion de traces disponibles pour la formation est égale à $\frac{1}{2^8} \cdot \binom{8}{l}$, qui est la plus basse pour $l = 0$ ou 8 . L'estimation de ces classes est donc moins précise.

Afin d'améliorer l'ACP, nous avons réduit le nombre de partitions de 9 à 7 sous-ensembles en fonction de la distance de Hamming $HD \in \{1, 2, 3, 4, 5, 6, 7\} = \{0, 1, 2, 3, 4, 5, 6, 7, 8\} \setminus \{0, 8\}$

Nous observons que de cette réduction fait que le meilleur taux de succès est obtenu par la projection sur le premier vecteur propre. Aussi, nous remarquons que le conditionnement de la matrice de covariance empirique se dégrade, ce qui confirme que les classes faiblement peuplées $l \in \{0, 8\}$ ont ajouté plus de bruit à la CPA. Étonnamment, cette approche est antinomique avec la DPA multi-bit de Messerges [54]. Si on fait une transposition de DES à AES, Messerges suggère au contraire de se débarrasser des classes $l = [1, 7]$ et de ne conserver que $L = \{0, 8\}$. Ces échantillons ont deux propriétés contradictoires :

- d'une part, ils véhiculent le plus d'informations, comme indiqué dans le tableau 6.2 mais
- d'autre part, ils sont aussi les échantillons les plus rares donc engendrent les coefficients les plus bruités de la matrice de covariance.

Comme Messerges ne fait pas usage de coefficients extra-diagonaux, son attaque n'est pas concernée par ce fait.

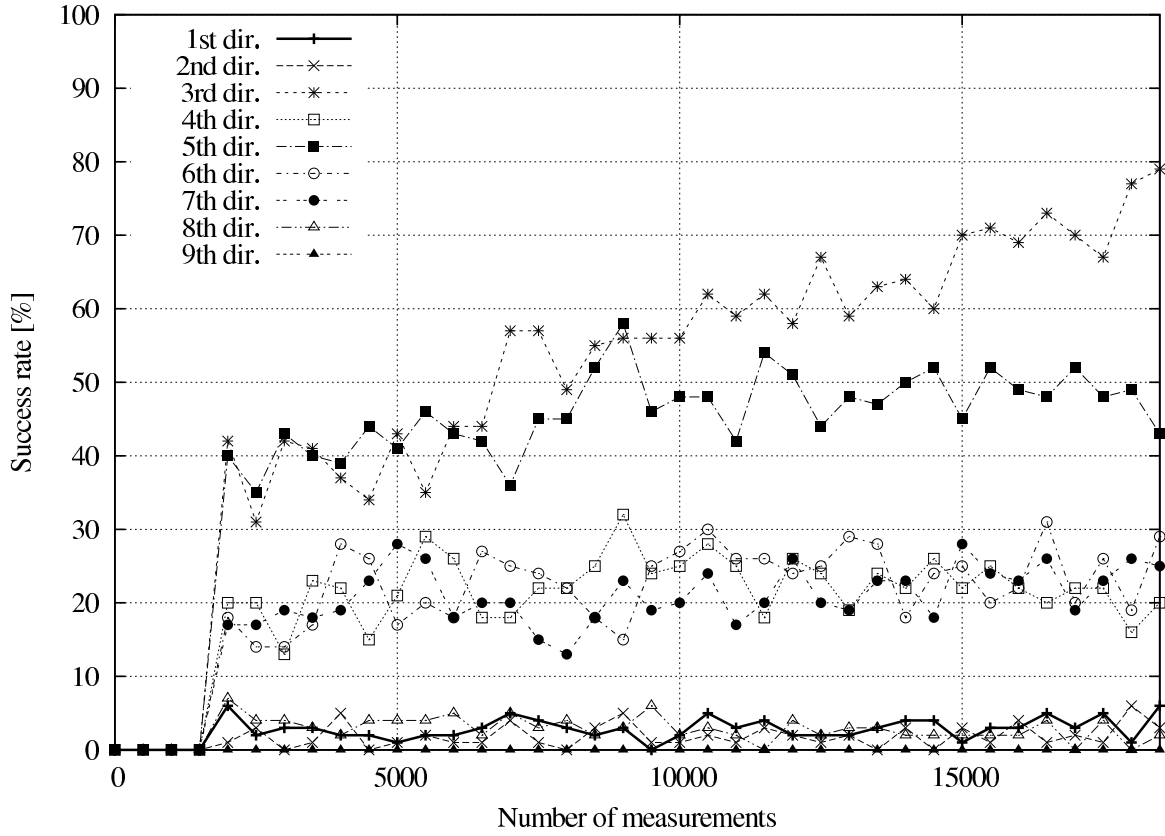


FIG. 6.7 – Taux de succès de la CPA après ACP à 25 cm.

6.2.2 Les échantillons combinés

6.2.2.1 Les observations

La trace de corrélation obtenue pour la bonne clé avec des mesures à distance est donnée dans la figure 6.8. Nous observons que les traces de corrélation sont extrêmement bruitées. En outre, pour certains échantillons dans le temps, identifiés par les chiffres $\{1,2,3,4\}$ sur la figure 6.8, l'amplitude de la trace de corrélation obtenue pour la bonne clé est nettement plus élevée que l'amplitude des traces de corrélation pour les mauvaises hypothèses de clés. Ces échantillons sont tous situés dans la même période d'horloge qui correspond au dernier tour de l'algorithme AES. Aux quatre instants indiqués, les échantillons sont sans aucun doute porteurs d'informations secrètes.

6.2.2.2 Combinaison principale d'échantillons et résultats.

Notre objectif est de montrer qu'il y a un gain considérable à combiner les fuites en provenance des quatre instants indiqués. Tout d'abord, nous confirmons que les quatre

TAB. 6.2 – L'information et la probabilité du poids de Hamming de la variable aléatoire uniformément distribué sur 8-bits .

Index de classes/	0	1	2	3	4	5	6	7	8
Information [bit]	8.00	5.00	3.19	2.19	1.87	2.19	3.19	5.00	8.00
Probabilité [%]	0.4	3.1	10.9	21.9	27.3	21.9	10.9	3.1	0.4

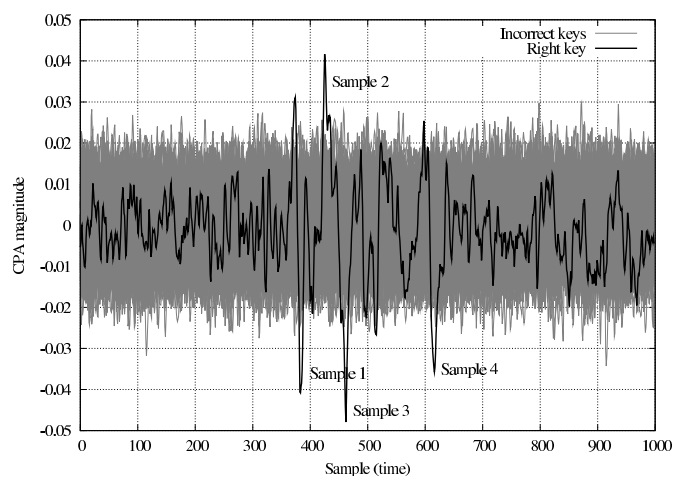


FIG. 6.8 – Traces de corrélations obtenues pour la bonne hypothèse de clé et pour une fausse hypothèse de clé à 25 cm.

échantillons de pics de la CPA sont en fait des POIs. Pour ce faire, nous réalisons des CPA avec succès sur ces échantillons dans le temps. Le résultat est montré dans la figure 6.9 : toutes les attaques sont réussies avec un taux de succès de 50 % après 12.000 traces. En second, nous proposons une méthode d'attaque qui exploite à la fois ces quatre échantillons. Des méthodes similaires ont déjà été mises en place dans le contexte de combinaison d'échantillons à fin d'attaquer des contre-mesures [63]. Dans [11], Chari *et al.* suggèrent d'utiliser le produit de deux modèles de fuite. Dans [36], Joye *et al.* recommandent de combiner deux échantillons avec la valeur absolue de la différence. Comme dans notre cas, où nous avons l'intention de combiner plus de deux échantillons, nous avons recours au produit comme fonction de combinaison. Nous l'appliquons à des coefficients de corrélation empiriques de Pearson $\hat{\rho}_t$, où t indique les quatres instants indiqués. Le nouveau distingueur dont nous faisons la promotion est donc :

$$\hat{\rho}_{\text{comb}} \doteq \prod_{t \in \text{Sample}\{1,2,3,4\}} \hat{\rho}_t. \quad (6.1)$$

Cette technique s'applique bien à des coefficients de corrélation de Pearson, qui sont déjà centrés par conception. Ainsi, il met en avant les coïncidences simultanées de forte corrélation, tandis qu'il rétrograde les hypothèses inexactes dont au moins un $\hat{\rho}_t$ est proche de zéro. Comme le montre la figure 6.9, le taux de succès de cette nouvelle attaque est supérieur à celui des attaques mono-échantillon. En outre, la figure 6.10 confirme que notre combinaison définie dans l'équation (6.1), bien que simple dans sa configuration, surpasse nettement une PCA après l'exécution de l'ACP.

Toutefois, nous avons seulement montré que si l'on connaît certains POIs de la courbe, une puissante attaque combinant plusieurs échantillons peut être montée. Maintenant, pour le moment, la seule méthode pour exhiber ces POIs a été d'appliquer une attaque réussie (une CPA dans notre cas). Par conséquent, une question ouverte est de localiser les points d'intérêt sans connaître la clé au préalable ou sans avoir procédé à une nouvelle attaque moins puissante. Nous proposons deux méthodes pour repérer les points d'intérêt : soit en ligne soit par pré-caractérisation sur un échantillon public en supposant que la position du POI ne dépend pas de la clé secrète.

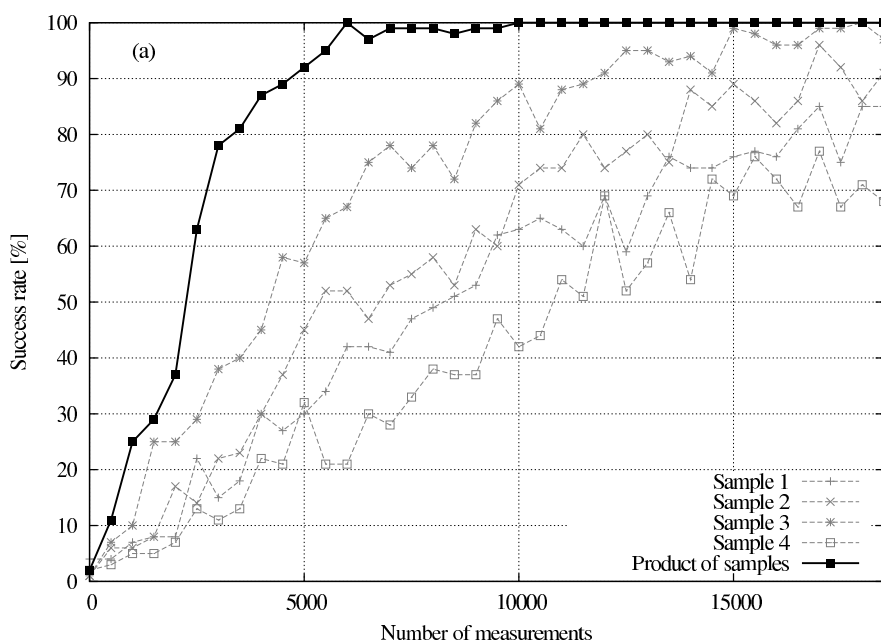


FIG. 6.9 – Taux de succès pour une attaque mono-échantillon, et attaque par produits de corrélations

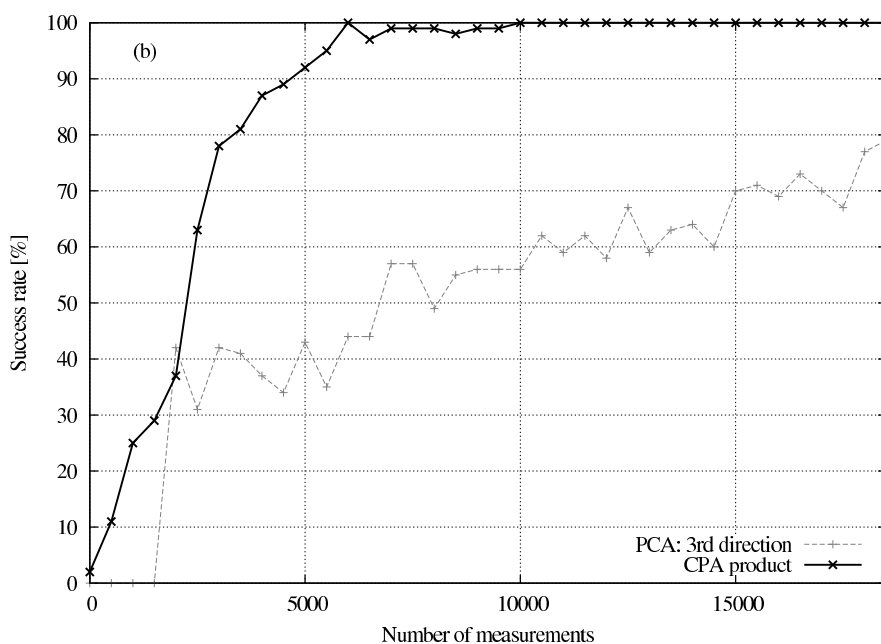


FIG. 6.10 – Taux de succès pour une CPA utilisant un pré-traitement par ACP et notre produit de corrélation

Chapitre 7

La portabilité des *templates*

À notre connaissance, la littérature des attaques *template* ne contient que la preuve de concept de ces dernières. Les traces destinées à l'entraînement et celles destinées à l'attaque sont acquises consécutivement sur le même circuit.

Ce cas est le plus favorable à l'attaquant, car les *templates* sont construits dans des conditions semblables à celles de l'attaque effective. Dans le cas d'une attaque réelle, un attaquant ne peut jamais faire l'entraînement et l'attaque sur le même circuit. L'effet du temps peut ajouter de nouveaux facteurs qui peuvent rendre l'attaque plus difficile. L'acquisition des traces, qui est un processus difficile et sujet à erreurs expérimentales, devrait idéalement être faite dans des conditions identiques durant le profilage et durant l'attaque en ligne. Chaque paramètre susceptible d'influencer les acquisitions, des caractéristiques de la résistance qui permet les mesures à la configuration de l'oscilloscope, devrait être similaire. Dans ce chapitre, nous étudions les conséquences (défavorables) pour l'attaquant du changement dans les condition d'acquisition. Parmi les problèmes identifiés, la dé-synchronisation des courbes en temps et la différence en amplitude. Puis, nous explorons quelque stratégies pouvant contourner ces problèmes pour un adversaire utilisant l'attaque *template*.

7.1 Méthodologie

Cette section vise à étudier les effets pratiques de possibles décalages entre les conditions expérimentales de l'entraînement et de la phase d'attaque sur le taux de succès et de l'entropie estimée [77]. Dans cette étude, nous nous concentrons sur l'effet du temps sur une attaque *template*. La reproductibilité des configurations de mesure est mise à l'épreuve.

Nous considérons trois ensembles de mesures acquises à partir d'une mise en œuvre de DES sur un même ASIC¹. :

¹Nous insistons sur le fait qu'il s'agit d'un unique ASIC et non pas de deux ASIC de même type.

Campagne A : 80000 mesures obtenues durant l'année 2006 avec une tension nominale (environ 1,2 V). Ces traces sont utilisées pour construire les *templates*.

Campagne B : 50000 mesures obtenues durant l'année 2010 avec une tension nominale (environ 1,2 V). Cette campagne de mesure est utilisée pour l'attaque en ligne de référence.

Campagne C : 50000 mesures obtenues durant l'année 2010 à une tension réduite (environ 1,0 V). Ces mesures sont utilisées pour l'attaque en ligne de test de variation, destinée à fournir une comparaison des deux campagnes (B et C) qui ont été réalisées à des moments proches mais avec tensions d'alimentation différentes, tous les autres paramètres étant identiques.

Plus précisément, les caractéristiques communes entre les campagnes A, d'une part, et B & C, d'autre part, sont énumérées ci-dessous :

- Le test du même ASIC soudé sur une carte d'évaluation.
- l'utilisation du même oscilloscope, avec exactement la même configuration (Voir la table 7.1 qui décrit plusieurs paramètres de l'acquisition).

Les différences entre les campagnes A et B/C sont :

- Le câblage entre la carte d'évaluation et de l'oscilloscope a été refait entre les campagnes A et B. Ce changement peut changer le temps de propagation de certains signaux, en particulier de celui qui déclenche l'acquisition par l'oscilloscope, entre la campagne A et les deux autres. il est différent en comparant A et B, A et C, mais il est le même pour B et C.
- L'ASIC a vieilli, et a donc subi une dégradation par injections de porteurs chauds [6] (effet difficile à quantifier sur un circuit qui n'a pas été conçu pour être testé contre le vieillissement).

Les statistiques de premier et de second ordre sur les trois campagnes sont détaillées dans les figures 7.1, 7.2 et 7.3.

Le chiffrement commence aux alentours de l'échantillon 5000 et s'arrête vers l'échantillon 15000.

L'augmentation de la consommation au cours des seize cycles d'horloge de cette coïncide avec les seize tours du chiffrement DES. Comme le montrent les figures 7.1(b), 7.2(b) et 7.3(b), la diminution concomitante de l'écart-type indique que la logique de contrôle devient déterministe pendant le chiffrement. En effet, dans l'ASIC ciblé, le *datapath* DES reste actif en permanence, ce qui provoque une activité électrique importante et incohérente en dehors de la fenêtre de chiffrement. À partir des courbes de variance, d'autres remarques intéressantes peuvent être déduites :

- Indépendamment de la campagne d'acquisition (A, B ou C), il y a un niveau de bruit constant (légèrement au dessous de 1 mV), qui représente le bruit d'acquisition intrinsèque provenant de la quantification des fluctuations thermiques et de sources de parasites externes.
- La hauteur des pics de variance aux fronts montants d'horloge augmente avec celle des pics en moyenne ; Ceci confirme que cette variance représente le bruit algorithmique.

TAB. 7.1 – Fichiers de configuration pour les trois campagnes étudiées dans ce chapitre – Ces données ont été extraites après les campagnes à partir des fichiers Agilent “.bin” (dont le format est décrit dans [1] au pages 409-413).

Campagne A	
<pre> \begin{center} ### Binary Header Format # Version: 10 # File Size: 80176 # Nbr of Waveforms: 1 ### Waveform Header # Header Size: 140 # Waveform Type: HORIZONTAL_HISTOGRAM # Nbr of Waveform Buffers: 1 # Nbr of Hits: 20003 # X Display Range: 7.4228448e-51 # X Display Origin: 2.32682e-06 # X Increment: 5e-11 # X Origin: 2.32676844e-06 # X Unit: 0 # Y Unit: 0 # Date: 8 APR 2006 # Time: 11:31:01 # Model: 54855A: # Label: channel 3 # Tag value: 0 # Index: 0 ### Waveform Data Header: # Waveform Data Header Size: 12 # Buffer Type: NORMAL_32 # Bytes Per Point: 4 # Buffer size (in bytes): 80012 \end{center} </pre>	
Campagnes B & C	
<pre> \begin{center} ### Binary Header Format # Version: 10 # File Size: 80176 # Nbr of Waveforms: 1 ### Waveform Header # Header Size: 140 # Waveform Type: HORIZONTAL_HISTOGRAM # Nbr of Waveform Buffers: 1 # Nbr of Hits: 20003 # X Display Range: 7.4228448e-51 # X Display Origin: 2.32682e-06 # X Increment: 5e-11 # X Origin: 2.32676844e-06 # X Unit: 0 # Y Unit: 0 # Date: 27 MAY 2010 # Time: 20:16:21 # Model: 54855A: # Label: channel 3 # Tag value: 0 # Index: 0 ### Waveform Data Header: # Waveform Data Header Size: 12 # Buffer Type: NORMAL_32 # Bytes Per Point: 4 # Buffer size (in bytes): 80012 \end{center} </pre>	

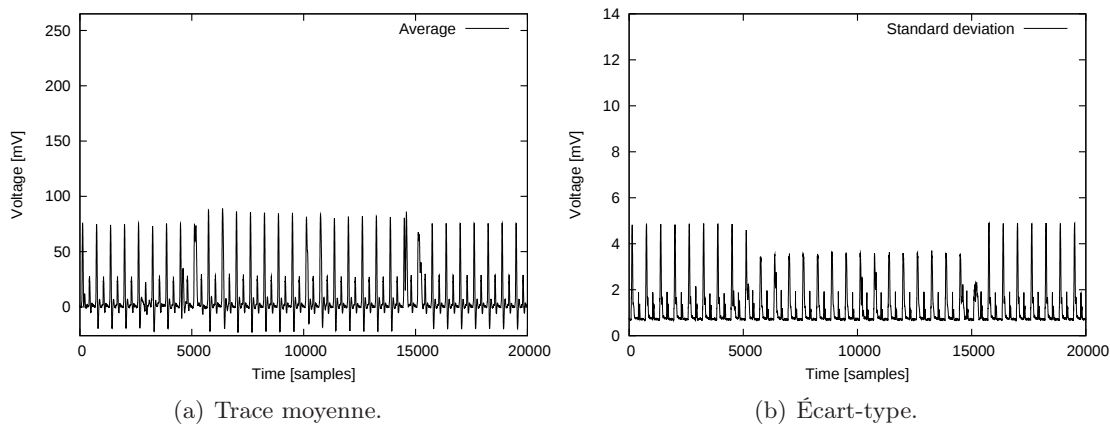


FIG. 7.1 – Trace Moyenne et écart-type pour la campagne A.

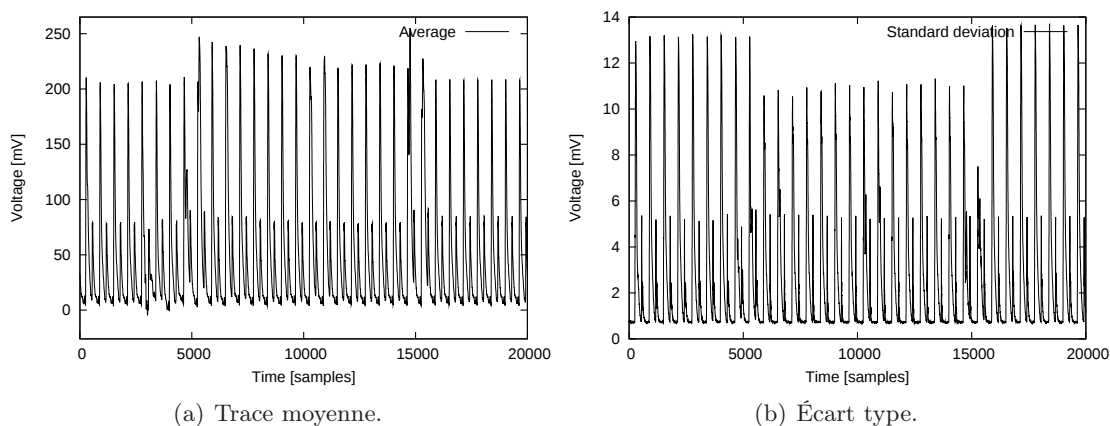


FIG. 7.2 – Trace Moyenne et écart type pour la campagne B.

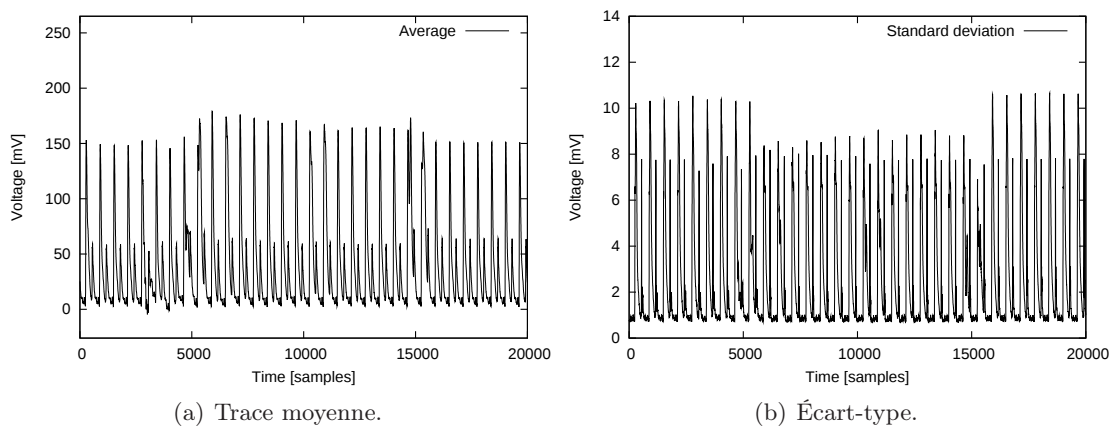


FIG. 7.3 – Trace Moyenne et écart type pour la campagne C.

La forme globale de ces trois campagnes est différente et, surtout, leur amplitude. D'autre part, l'examen des courbes montre qu'elles ne sont pas en phase. La figure 7.5(a), qui représente le premier tour de chiffrement, montre le décalage temporel entre la campagne A et les deux autres.

7.2 Re-synchronisation horizontale : POC et AOC

Le mauvais alignement temporel des traces est un problème habituellement rencontré dans l'analyse par canaux auxiliaires. En effet, dans le cas d'une attaque réelle, l'attaquant n'a pas facilement accès à une indication sur le début de l'opération à analyser. De plus, la plupart des systèmes embarqués réagissent en temps non-déterministe. Les raisons de ce non-déterminisme peuvent être internes au système (par exemple : la gestion d'un cache ou l'occurrence d'interruptions) ou être l'effet d'une contre-mesure (telle que, par exemple, la réorganisation plus ou moins aléatoire des instructions² [68] ou l'insertion de délais aléatoires entre des phases du programme³ [80, 17]. Plusieurs méthodes ont été proposées pour récupérer la meilleure synchronisation entre les signaux. Parmi celles-ci :

AOC⁴ dans la littérature anglo-saxonne. Cette méthode consiste en l'estimation du décalage entre deux traces par la minimisation de leur corrélation croisée.

POC⁵ dans la littérature anglo-saxonne. Cette méthode consiste en l'estimation du décalage entre deux traces par la minimisation de la corrélation croisée de leur phase.

Cependant, en général, quelque soit la source de la désynchronisation dans les mesures, cette dernière n'empêche pas les attaques de réussir. Ainsi, il a été montré dans [13] que le fait de moyenniser les signaux peut permettre à l'attaquant de contourner la désynchronisation des traces.

Supposons que la désynchronisation soit le résultat d'un déplacement du signal d'un certain nombre $n \in \llbracket 0, t \rrbracket$ où $t \in \mathbb{N}^*$ est la taille de la fenêtre de décalage. Dans le cas extrême où la désynchronisation est uniformément distribuée dans $\llbracket 0, t \rrbracket$ (ce qui a été possible dans [17]), la corrélation ρ entre les signaux désynchronisés et le modèle de fuite est égale à $1/\sqrt{t}$ fois celle existant entre les signaux synchronisés et le modèle. Il a aussi été vu que l'efficacité d'une analyse de courant par corrélation (CPA [9]) est directement liée à ces coefficients de corrélation. Plus précisément, d'après [48, 49] le nombre moyen de signaux nécessaires pour casser une mise en œuvre cryptographique est égale à :

$$3 + 8 \left(\frac{Z_{1-\alpha}}{\ln \left(\frac{1+\rho}{1-\rho} \right)} \right)^2 \quad (7.1)$$

où $Z_{1-\alpha}$ est le quantile d'une distribution normale pour l'intervalle de confiance avec une erreur de $1 - \alpha$. Pour de faibles valeurs de ρ , l'équation (7.1) est $\propto \rho^{-2}$. Par conséquent,

² *Instructions shuffling*, dans la littérature anglo-saxonne.

³ *Random delay insertion*, dans la littérature anglo-saxonne.

le nombre de traces pour casser une mise en œuvre cryptographique avec une fenêtre de décalage t est approximativement multiplié par $(1/\sqrt{t})^{-2} = t$. Cela montre que la contre-mesure se basant sur les décalages ne constitue pas un obstacle aux attaques.

Théoriquement, un signal X est un ensemble de valeurs réelles, *c.-à-d.* un élément dans $\mathbb{R}^{\mathbb{Z}}$. Nous considérons X_i l'échantillon de X à l'instant $i \in \mathbb{Z}$. Le signal mesuré Y est considéré désynchronisé par un décalage de k échantillons par rapport à X s'il satisfait :

$$\forall i, Y_i = X_{i-k}$$

En pratique, la référence (non décalée) X est inconnue et l'acquisition est limitée dans le temps. C'est pourquoi un signal sera défini comme un vecteur de $\mathbb{R}^n$. Lors d'une attaque réelle, le signal de référence X est inconnu. Le résultat de l'acquisition des traces est donc un ensemble de signaux mal alignés. L'attaquant choisit un des signaux comme référence et décale tous les autres de façon à ce que toutes les courbes soient en phase. La re-synchronisation d'un ensemble de signaux $S^1, S^2, \dots, S^M$ se ramène donc à M instances indépendantes du problème de la re-synchronisation d'un signal S^i avec un signal de référence, par exemple : S^1 .

7.2.1 AOC : La Corrélacion en amplitude

La corrélation croisée $X \star Y$ entre deux signaux X et Y est un signal, dont l'échantillon $i \in \llbracket 0, n \rrbracket$ est défini par :

$$(X \star Y)_i \doteq \sum_{j \in \mathbb{Z}_n} X_j \cdot Y_{j+i}$$

Dans cette notation, les indices de temps ne sont pas considérés dans un intervalle borné $\llbracket 0, n \rrbracket$, mais dans le groupe additif $\mathbb{Z}_n$. Nous avons choisi d'examiner les indices de l'échantillon modulo n afin de faciliter les calculs, comme pour l'identité impliquée dans l'équation (7.4).

La corrélation croisée peut être utilisée pour récupérer le bon décalage. Soit $\hat{k}$ le décalage qui maximise la corrélation croisée entre X et Y . Formellement :

$$\hat{k} = \arg \max_{k \in \mathbb{Z}_n} (X \star Y)_k \quad (7.2)$$

Notons ROR_k le décalage circulaire à droite des échantillons :

$$\forall i, \text{ROR}_k(X)_i = X_{i-k}$$

et $A \cdot B$ le produit coordonnée par coordonnée :

$$(A \cdot B)_i = A_i \cdot B_i$$

L'algorithme de re-synchronisation de l'équation (7.2) permet de retrouver le bon décalage quand Y est égal au signal de référence X décalé circulairement de k' :

$$\arg \max_{k \in \mathbb{Z}_n} (X \star \text{ROR}_{k'}(X))_k = \arg \max_{k \in \mathbb{Z}_n} \sum_{j \in \mathbb{Z}_n} X_j \cdot X_{j+(k-k')} = k'$$

Ce résultat découle de l'application du théorème de Cauchy-Schwarz à une auto-corrélation.

La corrélation croisée entre deux courbes peut être calculée de façon très efficace en utilisant les transformées de Fourier discrètes (TFD). La définition de la TFD et de la TFD inverse (TFDI), telles que mises en oeuvre dans la bibliothèque FFTW3 [21], est la suivante :

$$\begin{cases} \text{DFT}(X)_i & \doteq \sum_{j=0}^{n-1} X_j \cdot \exp(-2\pi j i \sqrt{-1}/n) , \\ \text{IDFT}(X)_i & \doteq \sum_{j=0}^{n-1} X_j \cdot \exp(+2\pi j i \sqrt{-1}/n) . \end{cases} \quad (7.3)$$

La définition de l'équation (7.3) n'est pas normalisée, puisqu'elle implique que :

$$\text{TFD} \circ \text{TFDI} = \text{TFDI} \circ \text{TFD} = n \times \text{Id}$$

Dans ces équations, les expressions sont des signaux, c.-à-d. des éléments de $\mathbb{R}^n$. Nous avons la propriété suivante :

$$\text{TFD}(X \star Y) = \overline{\text{TFD}(X)} \cdot \text{TFD}(Y)$$

Cela permet de réécrire la corrélation croisée :

$$X \star Y = \text{TFDI} \left(\overline{\text{TFD}(X)} \cdot \text{TFD}(Y) \right) / n$$

L'algorithme présenté dans cette section sera appelé AOC :

$$\text{AOC}(X; Y) \doteq X \star Y = \text{TFDI} \left(\overline{\text{TFD}(X)} \cdot \text{TFD}(Y) \right) / n \quad (7.4)$$

7.2.2 POC : Corrélation en phase

La corrélation en amplitude (AOC) peut être comparée à la corrélation en phase (POC), décrite dans [34, 56, 35].

Dans la corrélation en phase, les TFD du signal de référence X et du signal désynchronisé Y sont normalisées avant d'être multipliées. Le calcul est donc :

$$\text{POC}(X; Y) \doteq \text{TFDI} \left(\frac{\overline{\text{TFD}(X)} \cdot \text{TFD}(Y)}{|\text{TFD}(X)| \cdot |\text{TFD}(Y)|} \right) / n \quad (7.5)$$

Le POC permettra de synchroniser, puisque, si $Y = \text{ROR}_{k'}(X)$, alors :

$$\arg \max_{k \in \mathbb{Z}_n} \text{POC}(X; \text{ROR}_{k'}(X))_k = k' \quad (7.6)$$

En effet,

$$\text{TFD}(\text{ROR}_{k'}(X))_i = \text{TFD}(X)_i \cdot \exp(-2\pi k' i \sqrt{-1}/n)$$

Notons U le vecteur de composantes $U_i = \exp(-2\pi k' i \sqrt{-1}/n) \in \mathbb{C}$. Alors

$$\begin{aligned} \text{POC}(X; \text{ROR}_{k'}(X)) &= \text{TFDI}\left(\frac{\overline{\text{TFD}(X)} \cdot \text{TFD}(X) \cdot U}{|\text{TFD}(X)| \cdot |\text{TFD}(X) \cdot U|}\right)/n \\ &= \text{TFDI}(U)/n \end{aligned}$$

Le résultat de l'équation (7.6) vient du fait que :

$$\text{TFDI}(U)_i = \sum_{j=0}^{n-1} \exp(+2\pi j(i - k')\sqrt{-1}/n) = n \cdot \delta_{i-k'} \quad (7.7)$$

où δ est le symbole de Kronecker, qui satisfait : $\delta_i = 0$ si $i \neq 0$ et 1 sinon. Comparé au POC, AOC est capable de resynchroniser avec une résolution inférieure au taux d'échantillonnage.

7.2.3 AOC vs POC

Après la présentation de ces deux méthodes de re-synchronisation, dont une étude plus approfondie peut être consultée dans [31], nous comparons dans cette section leur efficacité.

Dans le cas des attaques *template*, si le partitionnement correct est connu (ce qui est vrai pour les *templates*, mais pas pour les traces utilisées pour l'attaque effective), nous pourrions comparer la capacité relative du AOC et du POC à récupérer le bon décalage en les appliquant sur le premier vecteur propre de l'ACP. Nous partons du principe que ceci est possible uniquement pour faire cette étude comparative.

En général, nous ne pouvons pas faire d'ACP sur les traces de l'attaque en ligne, car nous ne connaissons pas la clé. Nous pouvons utiliser la moyenne des traces des deux campagnes, ou une trace de chaque campagne pour estimer le décalage de synchronisation. Les performances sont respectivement illustrées dans les figures 7.4(b) et (c). Elles montrent que l'AOC contient nettement moins de bruit.

Par contre, dans le cas réaliste où nous n'avons pas de vecteurs propres, le POC semble plus adéquat : le pic maximal a un meilleur contraste pour cette méthode.

Nous avons donc utilisé le POC pour estimer le décalage des traces entre les campagnes A/B d'une part et A/C de l'autre. Nous continuons avec une re-synchronisation globale des courbes, représentées dans la figure 7.6(a) et 7.6(b). Après ce décalage dans le temps, les premiers vecteurs propres sont également en phase, comme illustré dans la figure 7.7.

Les valeurs exactes de la re-synchronisation sont données dans le tableau 7.2 ; Chaque échantillon représente $1/(20 \text{ Gsample}) = 0,05 \text{ ns}$. Ce tableau montre qu'*a priori*, la re-synchronisation sur les traces de consommation est légèrement différente de celle utilisant

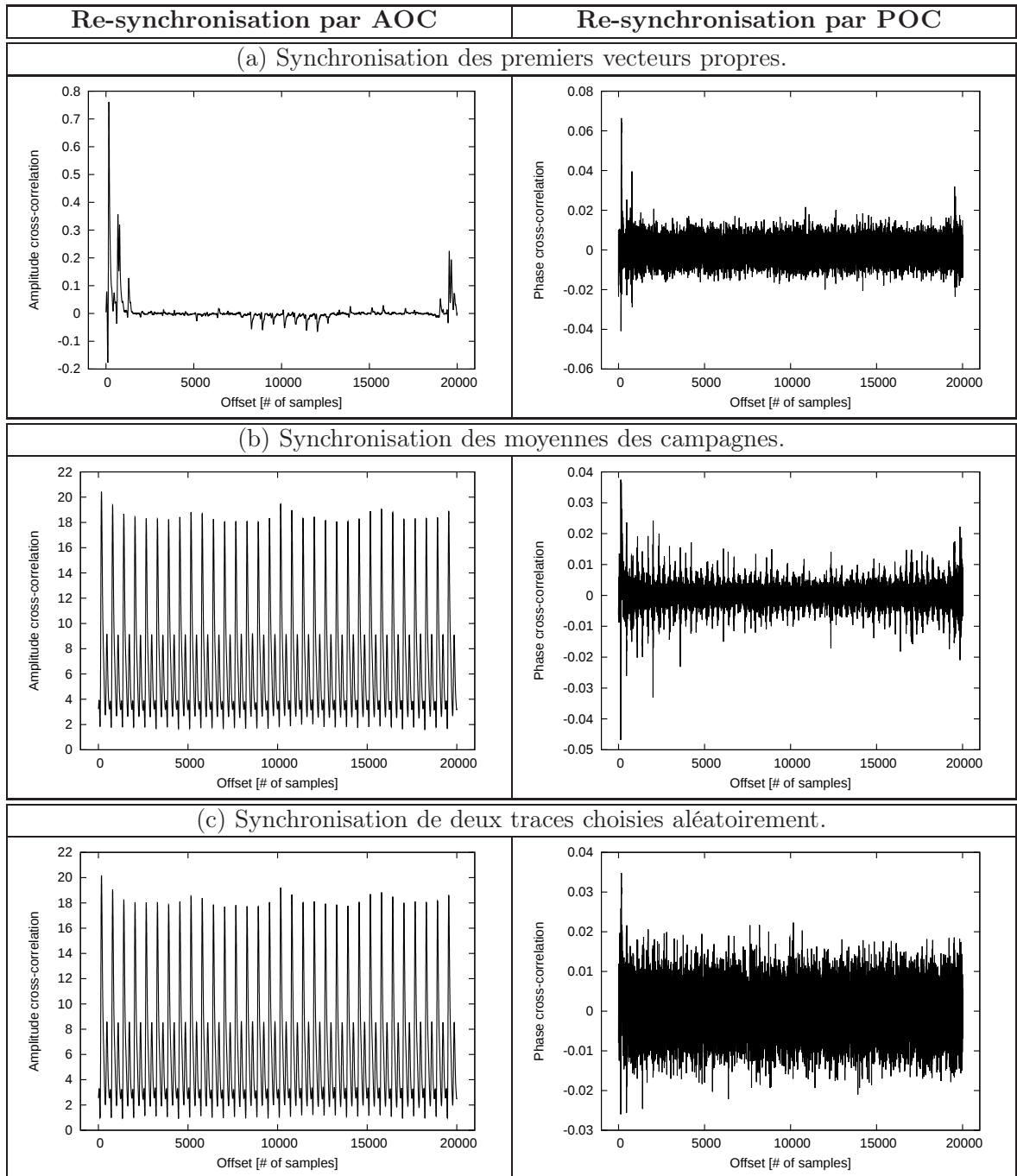
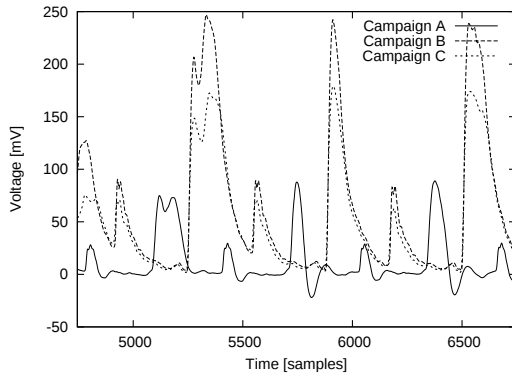
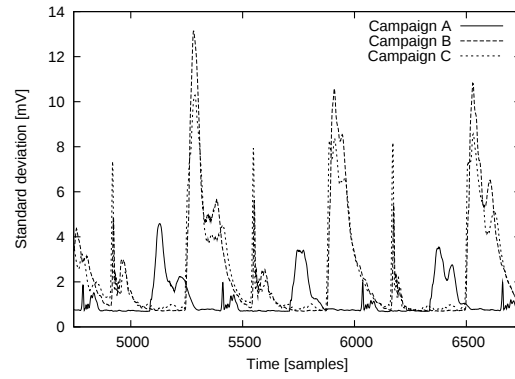


FIG. 7.4 – Résultats de la re-synchronisation entre (a) les vecteurs propres, (b) les moyennes des campagnes et (c) deux traces, pour AOC et POC.

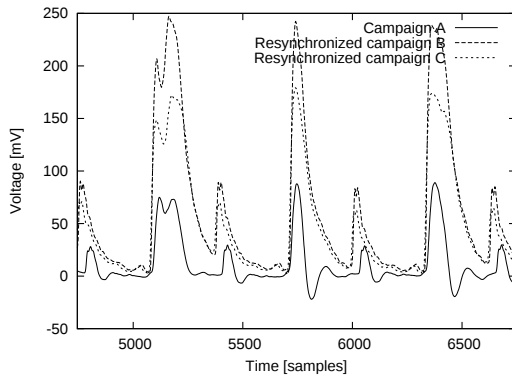


(a) Les moyennes

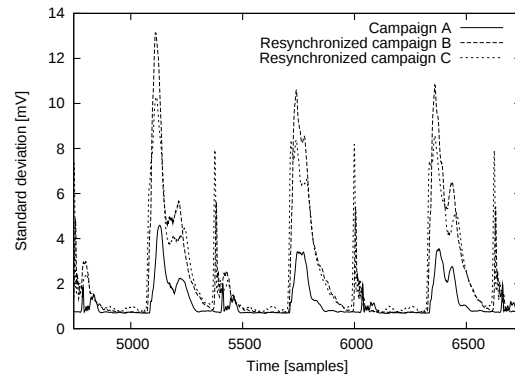


(b) L'écart-type

FIG. 7.5 – Comparaison des propriétés statistiques des campagnes A, B et C.



(a) Les moyennes



(b) L'écart-type

FIG. 7.6 – Comparaison des propriétés statistiques des campagnes resynchronisées A, B et C.

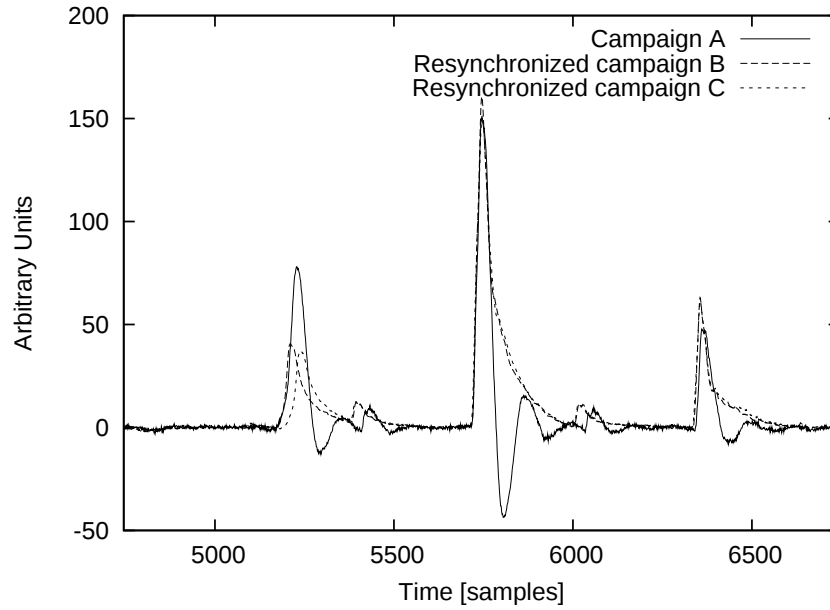


FIG. 7.7 – Comparaison des premiers vecteurs propres resynchronisés des campagnes A, B et C après l’ACP.

le premier vecteur propre de l’ACP. Néanmoins, nous nous appuyons dans la suite de cette étude sur ces valeurs pour faire la synchronisation des campagnes B et C, avec la campagne A. En effet, c’est plus réaliste dans le cas des attaques *template* car l’adversaire à besoin d’un minimum de traces par l’attaque en ligne.

7.3 Attaque avec de faux décalages

Nous avons testé les attaques *template* sur une grande fenêtre de décalages : $[0..1000]$, qui inclut les décalages trouvés précédemment et précisés dans le tableau 7.2. Le taux de succès et l’entropie estimée sont nos métriques de comparaison. Le taux de succès

TAB. 7.2 – Décalage temporel optimale trouvé par POC sur les campagnes A, B et C.

Base de synchronisation	Campagne		
	A	B	C
1 st vecteur propre (<i>Voir figure. 7.4(a)</i>)	0	166	166
traces brutes (<i>Voir figure 7.4(c)</i>)	0	170	172

de premier ordre est, par définition, le pourcentage de fois que la clé utilisée pendant le chiffrement occupe le premier rang parmi les 64 principales hypothèses. Dans la plupart des estimations, nous avons remarqué qu'un nombre croissant de traces réservées à l'attaque génère un taux de succès croissant. Par souci de représentativité, nous fixons le nombre de traces à 1.000 pour avoir assez de traces pour une estimation correcte des métriques. L'entropie estimée est également importante car elle illustre le classement de la vraie clé parmi les hypothèses de clés du chiffrement. Cette métrique est moins stricte et donc plus informative que le taux de succès quant à la tendance de l'attaque.

Cette première expérience consiste donc à vérifier si le taux de succès peut passer à 100% au décalage fourni par la synchronisation. Fondamentalement, nous voulons savoir si un attaquant a une marge d'erreur dans le processus de re-synchronisation.

Le premier résultat est que les décalages offrant un taux de succès ou une entropie estimée correcte ne sont pas ceux qui sont donnés par la re-synchronisation. En effet, les figures 7.8(a) 7.8(b) montrent que nous avons différents pics correspondant à différents décalages. ceci

D'autre part, en utilisant la définition stricte de l'entropie estimée [77] [équation (2)], nous faisons face à un problème pratique de calcul : les clés *ex aequo*. En fait, plus on ajoute de traces, plus les probabilités des clés tendent vers des valeurs limites. Ainsi, il est possible que quelque fois, la probabilité de certaines clés incertaines indiscernables (la vraie clé peut faire partie de cet ensemble) devienne nulle. Ce lot de clés aura la même probabilité, et donc la même entropie estimée.

Dans cet esprit, nous affinons le concept d'entropie estimée en ajoutant les notions de classements *pessimiste* et *optimiste*. Le classement est pessimiste (respectivement optimiste) si l'on considère le pire (respectivement le meilleur) classement des clés *ex aequo*. Dans nos figures, l'entropie estimée que nous représentons est la moyenne entre les entropies estimées pessimistes et optimistes. Ainsi, par exemple, lorsque l'attaque *template* estime que la bonne clé n'est pas celle qui correspond à la probabilité 1 (ce qui peut arriver en raison de la résolution finie des nombres à virgule flottante traités par les ordinateurs). Dans ce cas le classement pessimiste sera égal à 64 alors que le classement optimiste sera égal à 2. Par conséquent, nous optons pour un "compromis" d'entropie estimée de $(64 + 2)/2 = 33$.

Les figures 7.8(a) et 7.8(b) montrent une similitude entre le taux de succès et l'entropie estimée.

À ce niveau, nous pouvons déduire que l'attaquant peut récupérer la clé avec une bonne probabilité, même s'il ne synchronise pas les campagnes de profilage et celles pour l'attaque avec le bon décalage. Aussi, l'adversaire aurait remarqué que, pour certains décalages la clé est au dernier rang (*c.-à-d.* à la position 64 sur 64) en terme d'entropie estimée.

Nous conjecturons que ces erreurs sont causées par la différence d'amplitude entre les deux campagnes. C'est ce type de synchronisations que nous allons étudier dans la prochaine section.

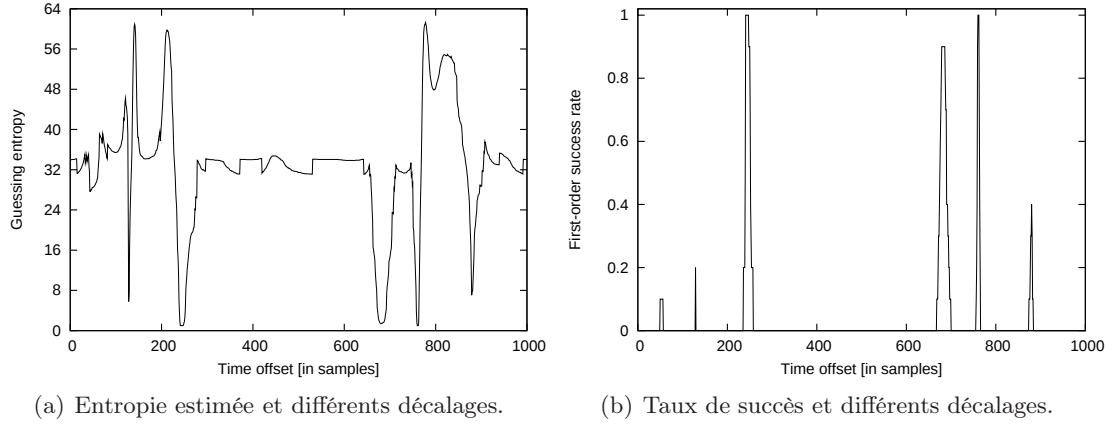


FIG. 7.8 – À la recherche du meilleur décalage possible pour la campagne B/A.

7.4 Homothétie verticale

Malgré notre souhait de faire des acquisitions de campagnes B et C proches de celles de la campagne A, nous avons été confrontés à des discordances des amplitudes. Dans le cas des simulations, il n’y a pas de disparité puisque la procédure est complètement sous contrôle (reproductible). Concernant les acquisitions réelles faites sur un circuit, de nombreux paramètres entrent en jeu, y compris le bruit.

Pour valider notre hypothèse, nous prenons des traces de la même campagne A, et les séparons en deux moitiés : la première pour le profilage et la seconde pour l’attaque. Nous déplaçons chaque trace cible dans une fenêtre de 2000 échantillons et examinons le taux de succès ainsi que l’entropie estimée, représentés dans les figures 7.9(a) et 7.9(b). Nous notons que le taux de succès croît jusqu’à 100% près du décalage zéro et reste à 0% aux autres décalages.

Comme nous pouvons le voir dans la figure 7.5(a), l’amplitude est significativement différente entre les moyennes. Ainsi, il est forcément difficile de tenter de casser une trace de la campagne B ou C en utilisant les *templates* construits à partir de la campagne A. Ceci est dû à la construction des *templates* qui est basée sur un modèle de consommation. Dans notre cas, le modèle choisi est la distance de Hamming, et donne cinq *templates* qui sont ordonnés selon le nombre de transitions dans le registre. La consommation représentée par la trace est plus élevée (de loin) que les moyennes des *templates*, et donc le maximum de vraisemblance donnera un mauvais résultat.

Cependant, les attaques *template* résistent mal aux homothéties, et donc les variations verticales peuvent certainement entraver l’attaque. L’effet d’une homothétie dans la tension est schématisée dans la figure 7.10.

Cette figure décrit un cas avec seulement deux *templates*, où la trace attaquée est inférieure à la tension nominale. Une analyse plus formelle sera donnée dans la section 7.5. Pour surmonter ce problème, nous effectuons une homothétie verticale sur les traces de

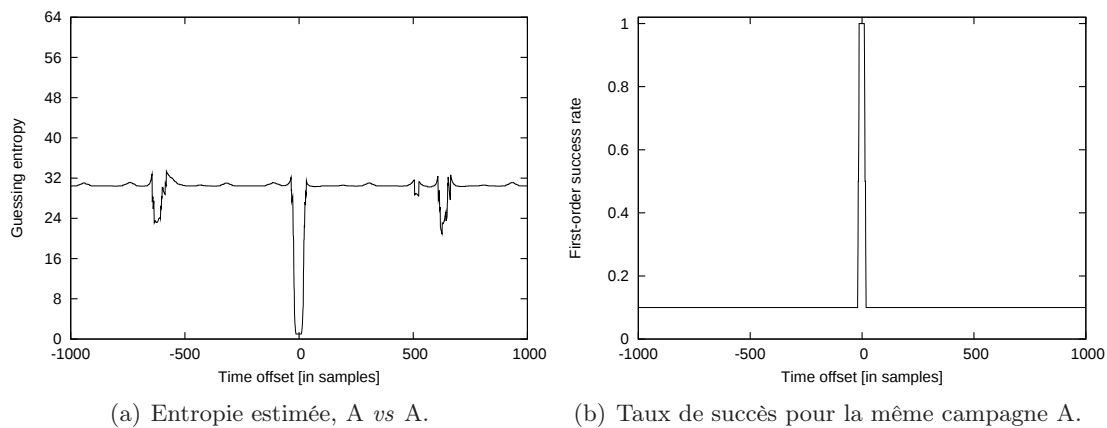


FIG. 7.9 – Taux de succès et Entropie estimée dans des conditions idéales (campagne A vs A).

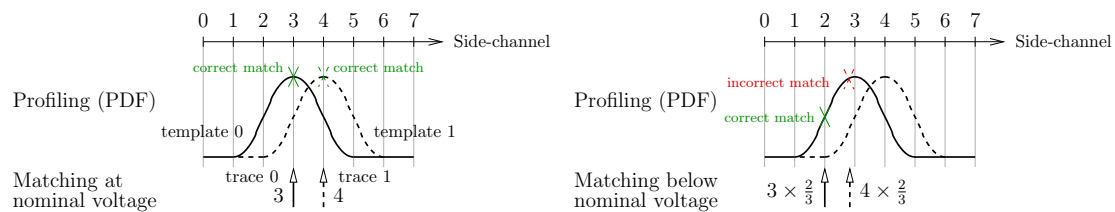


FIG. 7.10 – L'attaque en ligne peut donner des hypothèses erronées si les observations ont connu une homothétie. Dans cette illustration, il y a deux *template* (représentés avec des traits simples ou pointillés) et le facteur d'homothétie est de $2/3$.

l'attaque de façon à rapprocher leurs moyennes le plus près possible de celles des *templates*. Mais quel facteur adopter pour la remise à l'échelle ? La première expérience a été de multiplier par 0,5 chaque trace et de calculer le taux de succès et l'entropie estimée pour différents décalages dans le temps. Les résultats, illustrés dans les figures 7.11(a) et 7.11(b),r montrent que les décalages pour une attaque réussie sont entièrement différents de ceux obtenus sans tenir compte de l'amplitude. Aussi nous nous débarrassons des surprenants grands pics de la figure 7.8(b) et nous constatons que le taux de succès tend de plus en plus vers la valeur de 100% autour des décalages horizontaux prévus dans le tableau 7.2. Nous testons avec un autre facteur (0,47) obtenu en approchant le premier pic de la trace attaquée avec le premier pic de la moyenne générale des traces de profilage. Ce nouveau facteur améliore davantage l'entropie estimée et le taux de succès, surtout à côté du bon décalage.

Nous pourrions travailler séparément sur chaque point et calculer un coefficient de mise à l'échelle pour chaque point. Toutefois, pour automatiser cette procédure, nous suggérons de centrer et de normaliser toutes les traces : celles pour le profilage et également celles pour l'attaque en ligne. Cette normalisation harmonise les campagnes d'acquisition et réduit ainsi l'écart d'échelle entre elles. Cette méthode est mieux étudiée dans la section 7.6.

7.5 Modélisation pour des campagnes mal alignées

Dans cette section, nous tentons de prouver qu'il y a des situations où les *templates* sont déformés de façon imprévue. Par conséquent, un classement des clés peut être inversé dans certains cas. Par souci de simplicité, nous modélisons le taux de succès d'une attaque *template* dans le cas univariée avec seulement deux classes. Par ailleurs, nous supposons que les deux modèles ont la même variance σ^2 , et ne diffèrent que par leurs moyennes. Comme dans l'utilisation de l'ACP (qui n'en est pas moins exagéré dans les attaques univariées), nous supposons que les modèles sont centrés. Ainsi, nous décidons que les modèles ont une moyenne $\pm\mu$, qu'on peut encore simplifier à ± 1 . Formellement, cela signifie que la variable aléatoire conditionnelle $L \mid S = 0$ suit la loi $\mathcal{N}(-1, \Sigma^2)$, et que $L \mid S = 1 \sim \mathcal{N}(+1, \Sigma^2)$.

Dans cette situation, le taux de succès d'une attaque peut être estimé formellement [67]. Comme dans notre cas, où il y a seulement deux *templates*, il n'y a pas de notion de taux de succès à un certain ordre : la clé est soit classée au premier rang (et correcte) ou deuxième (dernière, et donc incorrecte). En effet, pour une mesure de fuites $L = l$ avec utilisation de la bonne clé $S^* = s^*$ (inconnue de l'attaquant), l'attaque est réussie si :

$$s^* = \arg \max_s P[S=s \mid L=l].$$

Or nous avons,

$$P[S=s \mid L=l] = P[L=l \mid S=s] \times \frac{P[S=s]}{P[L=l]}$$

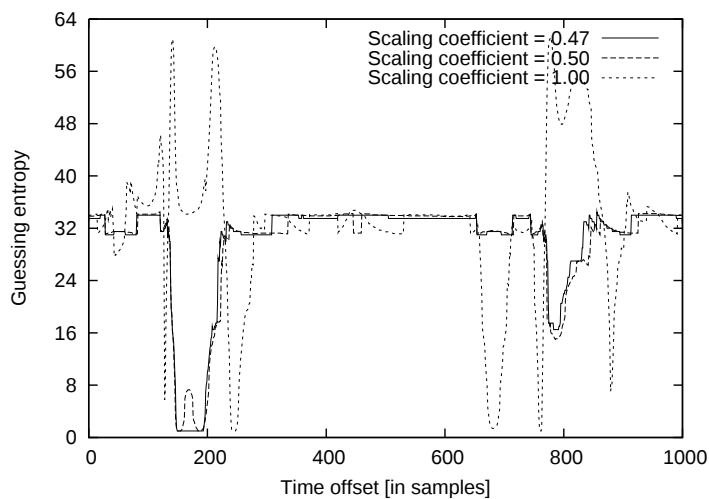
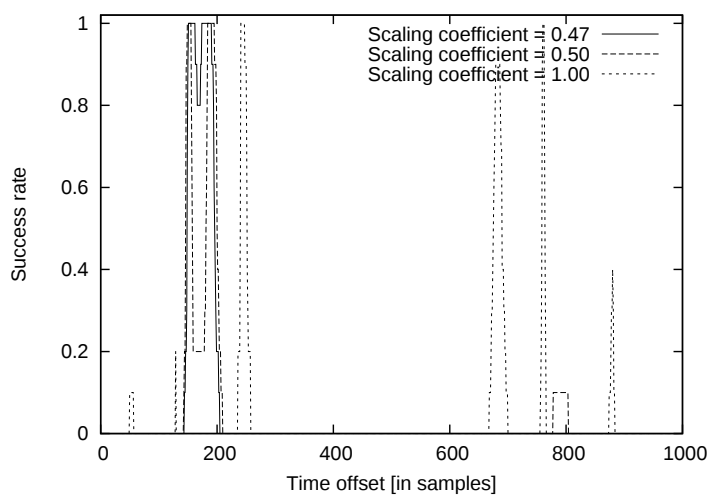
(a) Entropie estimée *vs* différents décalages.(b) Taux de succès *vs* différents décalages.

FIG. 7.11 – Comparaison des attaques avec différents facteurs en fonction du décalage de re-synchronisation. L'entraînement est fait sur la campagne A et l'attaque en ligne sur la campagne de B.

Comme les clés sont réparties uniformément, la probabilité $P[S = s]$ est égale à $1/2$ et ne dépend pas donc de la clé s . Ainsi

$$\arg \max_s P[S = s \mid L = l] = \arg \max_s P[L = l \mid S = s]$$

Sous cette forme, et puisqu'il n'y a que deux clés possibles ($s^* \in \{0, 1\}$), l'attaque sur les $L = l$, connaissant la bonne clé s^* est réussie si :

$$P[L = l \mid S = s^*] > P[L = l \mid S = \overline{s^*}]$$

En général, l'indicateur de succès est le résultat d'une expérience qui renvoie 1 ou 0, selon que la trace attaquée est classée avec succès ou non. Dans le cas de cette étude, l'indicateur du succès peut être également écrit : $\delta(l \geq 0)$ si $s^* = 1$ et $\delta(l \leq 0)$ si $s^* = 0$, indépendamment de Σ^2 .

Le taux de succès général est l'espérance sur toutes les fuites L et sur toutes les clés candidates S^* :

$$\begin{aligned} \text{SR} &\doteq \mathbb{E}_{S^*} \mathbb{E}_L \left(\arg \max_s P[L \mid S = s] \right) = S^* \\ &= P[S^* = 0] \int_{\mathbb{R}} \delta(l \leq 0) P[L = l \mid S^* = 0] dl \\ &+ P[S^* = 1] \int_{\mathbb{R}} \delta(l \geq 0) P[L = l \mid S^* = 1] dl \\ &= \frac{1}{2} \int_{-\infty}^0 P[L = l \mid S^* = 0] dl + \frac{1}{2} \int_0^{+\infty} P[L = l \mid S^* = 1] dl . \end{aligned} \quad (7.8)$$

Maintenant, nous supposons que les mesures ont subi un décalage vertical β et une homothétie de coefficient α . Si les mesures pour une attaque sont noyées dans du bruit centré de variance σ^2 , alors les variables aléatoires conditionnelles $L \mid S^* = 0$ et $L \mid S^* = 1$ suivent respectivement les lois $\mathcal{N}(\beta - \alpha, \sigma^2)$ et $\mathcal{N}(\beta + \alpha, \sigma^2)$.

Ainsi, en appliquant l'équation (7.8), nous obtenons :

$$\text{SR} = \frac{1}{2} + \frac{1}{4} \left(\operatorname{erf} \left(\frac{\beta + \alpha}{\sigma} \right) - \operatorname{erf} \left(\frac{\beta - \alpha}{\sigma} \right) \right) \quad (7.9)$$

où erf est la fonction d'erreur, définie par : $\operatorname{erf}(t) = \frac{2}{\sqrt{\pi}} \int_0^t \exp -t^2 dt$. Nous notons que erf est strictement croissante, symétrique (*c.-à-d.* $\forall t \in \mathbb{R}, \operatorname{erf}(-t) = -\operatorname{erf}(t)$), et bornée : $\lim_{t \rightarrow \pm\infty} \operatorname{erf}(t) = \pm 1$.

Nous remarquons que dans le cas nominal, $\beta = 0$ et $\alpha = 1$, le taux de succès commence à 1, lorsque le bruit est inexistant ($\sigma \rightarrow 0^+$), et diminue progressivement vers $1/2^+$ (la moitié, en arrivant par le haut) quand σ augmente. Cela signifie que la bonne clé est toujours trouvée, mais avec un succès qui diminue avec le niveau du bruit. Si $\beta = 0$ et α est arbitraire, mais positif, alors les mêmes conclusions peuvent être tirées.

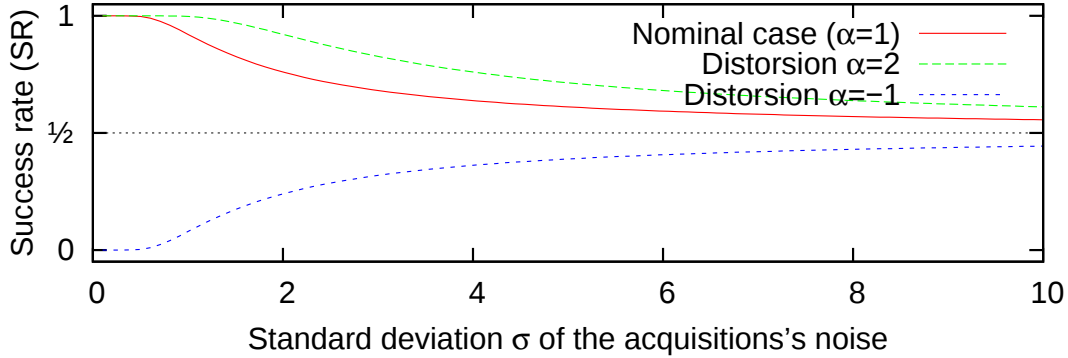


FIG. 7.12 – Les taux de succès en fonction du bruit dans les traces attaquées.

La figure 7.12 illustre cette situation. Cela peut paraître étonnant que le taux de succès soit d'autant meilleur que α est grand. Mais cette mention est artificielle : en pratique, nous nous attendons à avoir du bruit, notamment le bruit algorithmique, qui augmente si le signal augmente. En outre, avec plus de deux *templates*, le lien du taux de succès avec α est plus complexe et l'utilisation de α n'apporterait aucun avantage à l'attaquant.

La différence que nous voulions mettre en avant se produit lorsque $\alpha < 0$. Dans ce cas, le taux de succès est inférieur à $1/2$. Cela signifie que la mauvaise clé est toujours classée en premier. Les graphes tracés sur la figure 7.12 correspondent aux taux de succès.

Nous insistons sur le fait que $\alpha < 0$ est un cas "pathologique", qui n'est pas censé se produire en pratique. Cependant, il y a des situations délicates où effectivement $\alpha < 0$. Un attaquant peut inverser par erreur l'acquisition (cette fonctionnalité existe sur la plupart des oscilloscopes). Toutefois, si l'attaquant est assez prudent, le α devrait être positif, voir même proche de 1, comme le montre la figure 7.13(a). Cette figure représente la moyenne des fuites $L = l$ pour une clé donnée (en utilisant deux couleurs). Le problème causé par $\alpha < 0$ peut résulter d'un mauvais alignement temporel des *templates* avec les traces acquises pour être attaquées si elles ont des rebonds. Les rebonds se produisent si le capteur d'acquisition n'est pas adapté en impédance avec le circuit. Dans ce cas, la fuite est modulée par un sinus amorti, comme montré dans figure 7.13(b). Si l'acquisition décalée $t_1 - t_0$ est égale à la moitié de la fréquence d'oscillation, la courbe est en effet considérée comme inversée.

Les rebonds négatifs auront une amplitude négative $-1 < \alpha \leq 0$, quoique plus faible que l'amplitude maximale du signal, en valeur absolue. Cependant, si les traces d'attaque sont plus grandes en amplitude que les *templates*, comme c'est le cas de campagnes B et C, alors il est possible que le rebond négatif soit amplifié à l'amplitude de -1 , et donc inverser totalement le classement des clés durant la phase de l'attaque.

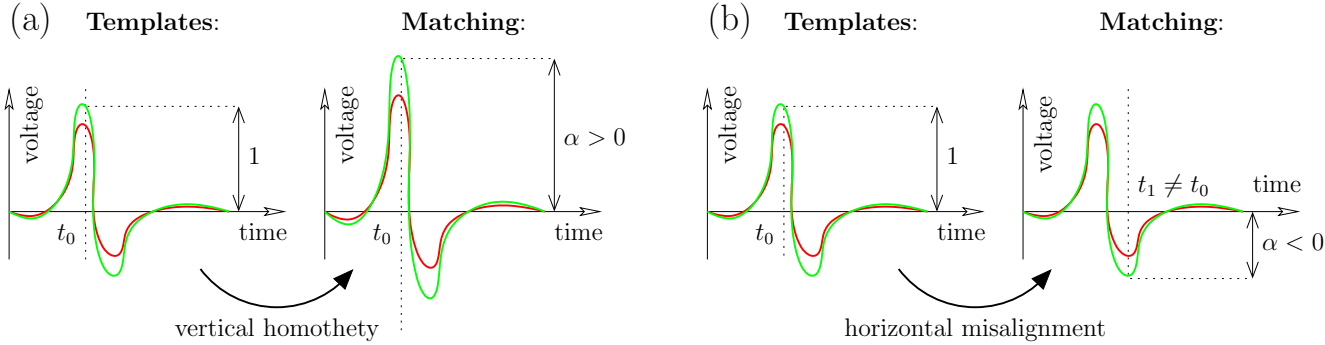


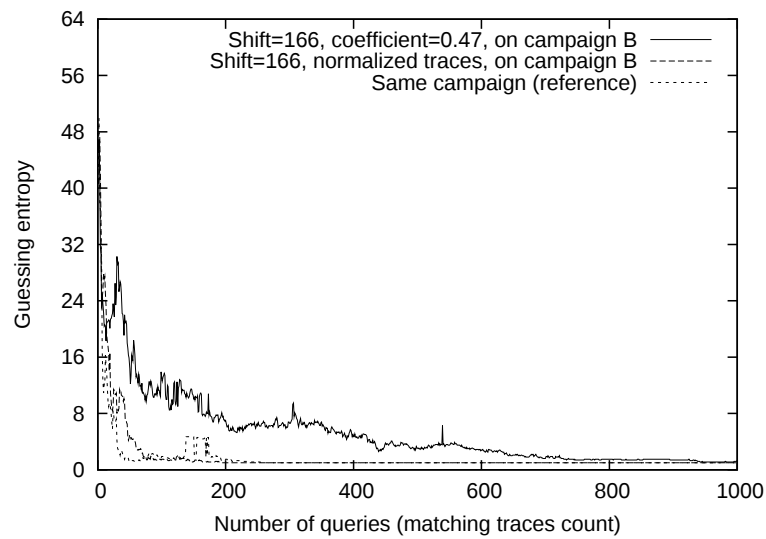
FIG. 7.13 – Illustration du taux de succès pour des coefficients (a) positifs et (b) négatifs.

7.6 Estimation de la portabilité

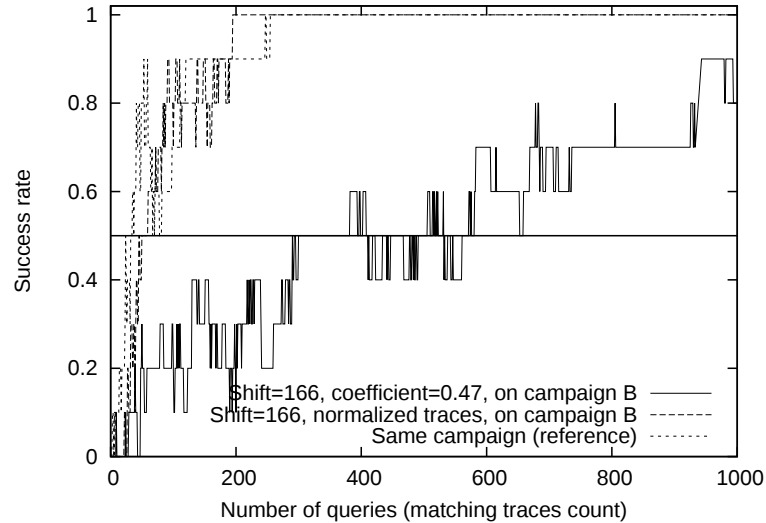
En utilisant le redimensionnement vertical (décrit à la section 7.4) et la re-synchronisation horizontale adéquate, nous comparons l'efficacité des attaques *template*. La métrique utilisée est le taux de succès et l'entropie estimée, calculés entre les *templates* (Campagne A) et des traces d'attaque prises juste après les traces du profilage, à partir des campagnes B et C.

Nous voyons clairement sur la figure 7.14(a) et 7.14(b) que, quand les traces sont prises avec une configuration différente de celle des *templates*, le taux de succès est plus faible que dans le cas idéal, où les campagnes d'entraînement et de l'attaque sont ressemblantes. Nous insistons, bien sûr, sur le fait que les traces sont bien séparées pour chaque expérience : il n'y a en effet pas d'intersection entre les traces d'attaque et celles du profilage. Bien que l'attaquant pourrait trouver la clé, il requiert néanmoins plusieurs traces pour y arriver. En fait, l'attaquant capable d'adapter globalement la campagne d'attaque (par un facteur de 0,47) n'a pas la même efficacité qu'un attaquant qui contrôle mieux l'acquisition de traces. La perte peut être estimée en termes de nombre de traces à 50% de taux de succès : comme nous pouvons le voir dans la figure 7.14(b), sans précaution, l'attaque nécessite environ 10 fois plus de traces.

Néanmoins, si les deux campagnes pour le profilage et pour l'attaque sont normalisées (chaque trace est remplacée par sa différence avec la trace moyenne de la campagne, et cette soustraction est elle-même divisée par l'écart global de la campagne standard), nous remarquons (aussi sur la figure 7.14(a) et 7.14(b)) que les métriques sont presque aussi bonnes que pour les attaques *template* sur la campagne de référence (la moitié pour l'entraînement par rapport l'autre moitié pour l'attaque). Ainsi, la normalisation des campagnes en collaboration avec la re-synchronisation dans le temps est un pré-traitement qui permet un portage réussi des *templates* de la campagne de A vers B ou C. On peut donc affirmer que, selon nos expériences, les attaques *template* (avec le pré-traitement indiqué) peuvent en effet être efficaces, même si des différences existent dans le temps ou en amplitude.



(a) Comparaison de l'entropie estimée.



(b) Comparaison du taux de succès.

FIG. 7.14 – Efficacité d'un attaquant utilisant les techniques de pré-traitement appropriées pour les attaques *template*, en utilisant la campagne A pour les *templates*.

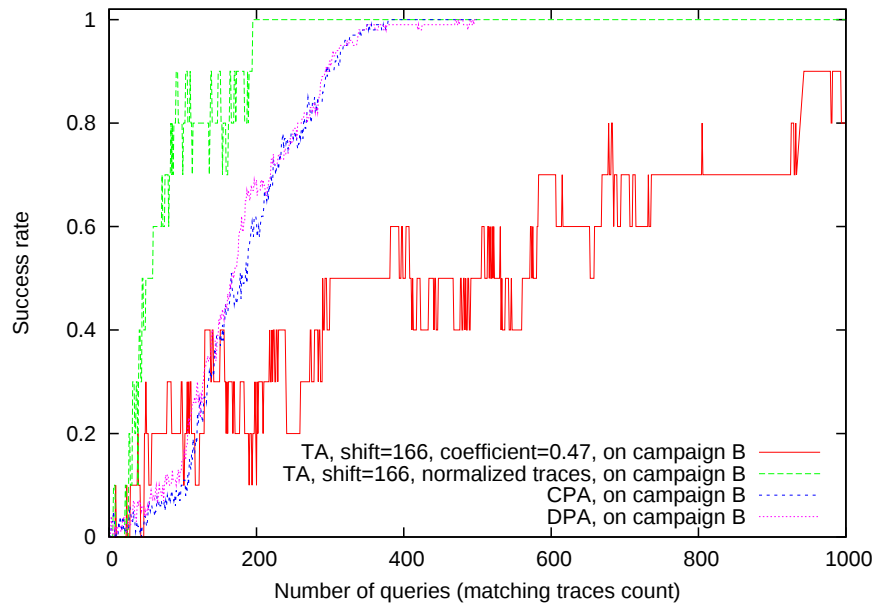


FIG. 7.15 – Comparaison des taux de succès des attaques *template* (brutes ou avec les techniques de pré-caractérisation proposées) et les attaques univariées traditionnelles (DPA and CPA).

Il est également intéressant de comparer les attaques *template* avec des attaques univariées (généralement DPA [8] et CPA [9]). Dans la figure 7.15, on peut voir qu'un attaquant voulant retrouver les clés de la campagne B avec une pré-caractérisation brute (et donc inadaptée) de la campagne A produit des résultats pires que pour la DPA ou CPA, mais les attaques *template* menées avec la re-synchronisation et la normalisation donnent de meilleurs résultats que ces dernières.

Chapitre 8

Les bienfaits du traitement du signal

Nous avons exposé dans le chapitre 5.4 des techniques qui peuvent améliorer une attaque *template*, comme le seuillage basé sur les vecteurs propres, et nous avons notamment discuté dans la section 5.4 de l'avantage que peut avoir un attaquant par rapport à un évaluateur qui ne connaît pas ces *astuces*.

Nous tenions aussi à signaler que les outils du traitement du signal peuvent être d'une importance primordiale quand l'attaquant est limité dans le temps et surtout en nombre de traces. L'analyse des signaux, en général, et des traces de consommation, en particulier, peut être effectuée soit dans le domaine temporel, soit dans le domaine fréquentiel.

L'analyse SCA dans le domaine temporel est plus efficace que l'analyse dans le domaine fréquentiel car nous disposons de l'ensemble la fuite. Néanmoins, une analyse de Fourier s'avère être particulièrement utile si des signaux acquis sont désynchronisés. L'inconvénient de la transformée de Fourier dans le cadre d'une analyse SCA est qu'elle est uniquement capable de récupérer le contenu fréquentiel global d'un signal donc elle détruit l'information temporelle. Cependant, la transformée de Fourier à court terme (TFCT), qui calcule la transformée de Fourier sur une fenêtre qui sera décalée au fur et à mesure sur le signal, donne le contenu temps-fréquence d'un signal avec une fréquence constante due à la longueur fixe de la fenêtre. Elle constitue donc un palliatif à l'inconvénient de la transformée de Fourier globale.

Pour les basses fréquences dans un signal SCA, souvent une bonne résolution fréquentielle est requise, alors que pour les hautes fréquences, la résolution temporelle est plus importante. Un outil alternatif avec quelques propriétés intéressantes est **la transformée en ondelettes**. Cette transformée utilise une fonction appelée fonction mère, qui est non forcément sinusoïdale (à la différence de la transformée de Fourier) pour analyser un signal. Cette fonction mère opère des translations et dilatations sur le signal pour avoir une analyse variée, sur une fenêtre variable. Bien que la théorie des ondelettes a été appliquée avec succès dans de nombreuses applications en sciences et en ingénierie,

elle n'a été utilisée qu'à deux reprises :

- pour éliminer le bruit d'une traces d'un canal auxiliaire [60] et
- pour estimer la fonction de densité de probabilité de la fuite d'information [66].

Dans ce chapitre, nous proposons de nouveaux moyens permettant à un adversaire de profiter des avantages de la multi-résolution fournis par l'analyse en ondelettes. Mais, tout d'abord, nous montrons comment les ondelettes peuvent être correctement utilisées pour améliorer la compression des traces en temps réel sans perte d'information.

Lors de l'acquisition de traces, nous avons besoin d'un taux d'échantillonnage élevé pour détecter les fuites cachées, en particulier lorsque la mise en œuvre est protégée. Par conséquent une mémoire de grande capacité de stockage est nécessaire.

Dans ce chapitre, nous proposons une amélioration de la méthode de Pelletier *et al.* [60] et nous l'utilisons filtrer les traces.

Nous discuterons du problème de filtrage du bruit dans le contexte des canaux auxiliaires et nous proposerons d'utiliser une approche utilisant la théorie de l'information comme outil complémentaire. Puis, nous mettons en évidence la capacité des ondelettes à détecter et extraire les motifs d'un processus de chiffrement lors de la réalisation d'une SPA (c.-à-d., une interprétation directe des traces acquises).

Finalement, nous montrons que, dans le cadre d'une attaque par canaux cachés, une analyse multi-résolution apporte des avantages, par rapport à une considération simple dans le temps ou à une résolution en fréquence.

En fait, pour des traces extrêmement bruitée, le gain en nombre de traces nécessaires pour récupérer la clé secrète est entre 30 % à 100 %, comparé à une attaque ordinaire. Par ailleurs, nous notons que cette analyse est générique car elle peut être considérée comme un plug-in utilisé pour améliorer les attaques existantes.

Nous détaillons dans la suite les avantages apportés par l'analyse multi-résolution à deux attaques : l'attaque *template* basée sur l'ACP et la CPA.

8.1 L'analyse multi-résolution

Lors d'une analyse par canaux cachés, les traces sont naturellement acquises dans le domaine temporel. Elles sont ensuite habituellement analysées dans le domaine temporel. Cependant, les informations comprises dans les signaux peuvent être traduites en différentes représentations. Grâce à ces différentes représentations, plus de détails concernant le signal, tels que le contenu fréquentiel, peuvent être utilisés pour mettre en évidence les dynamiques cachées liées au processus cryptographique. Les représentations les plus couramment utilisées pour l'analyse du contenu fréquentiel d'une signal sont :

- sa transformée de Fourier et
 - sa transformée de Fourier à court terme(TFCT).
-

8.1.1 Transformée de Fourier

La transformée de Fourier est une représentation mathématique qui décompose une fonction en de nombreuses composantes, dont chacune a une fréquence précise. L'ensemble des fonctions périodiques de période donnée, $2p$ est un espace vectoriel sur $\mathbb{R}$. De ce fait, toute fonction périodique $f(t)$, avec une période $2p$ décomposée sur la base constituée des fonctions $t \mapsto \sin(kt)$, $k = 1, \dots$ et $t \mapsto \cos(kt)$, $k = 0, 1, \dots$.

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} (a_n \cos(\frac{n\pi t}{p}) + b_n \sin(\frac{n\pi t}{p})). \quad (8.1)$$

où

$$\begin{aligned} a_n &= \frac{1}{p} \int_{-p}^p f(t) \cos(\frac{n\pi t}{p}) dt; n = 0, 1, 2, \dots, \\ b_n &= \frac{1}{p} \int_{-p}^p f(t) \sin(\frac{n\pi t}{p}) dt; n = 1, 2, \dots \end{aligned} \quad (8.2)$$

$$(8.3)$$

Ces séries de sinus et cosinus sont appelées *séries de Fourier*.

Les *séries de Fourier* peuvent être généralisées aux nombres complexes pour calculer la transformée de Fourier. Ainsi, la transformée de Fourier du signal $f(t)$ peut être exprimée par :

$$\hat{f}(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} f(t) e^{-i\omega t} dt. \quad (8.4)$$

Par ailleurs, le signal d'origine $f(t)$ peut être reconstruite à l'aide de la transformée de Fourier inverse comme suit :

$$f(t) = \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{i\omega t} dt. \quad (8.5)$$

En pratique, les signaux acquis expérimentalement ne sont pas continus dans le temps, mais échantillonnés dans des intervalles de temps discrets δT . Dans ce cas, l'analyse en fréquence est réalisée en utilisant la *transformée de Fourier discrète* (TFD). Cependant l'utilisation de la transformée de Fourier ne révèle pas comment le contenu fréquentiel du signal varie avec le temps et cela peut limiter le mérite de transformée de Fourier surtout quand nous avons besoin aussi de l'information temporelle.

8.1.2 Transformée de Fourier à Court Terme

Afin de contourner les limitations de la transformée de Fourier, on peut introduire une fenêtre d'analyse d'une certaine longueur, qui se déplace suivant l'axe du temps du signal pour effectuer une *transformée de Fourier localisée*. Ce concept appelé *Transformée de Fourier à Court Terme* ou (TFCT) permet de récupérer la fréquence et l'information

temporelle à partir d'un signal. Techniquement, la TFCT emploie une fonction de fenêtre glissante $g(t)$, centrée à l'instant τ et peut être exprimée par l'équation suivante :

$$f_{TFCT}(\tau\mu) = \int_{-\infty}^{+\infty} f(t)g^*(t - \tau)e^{-i2\pi\mu t} dt \quad (8.6)$$

Lorsque la fenêtre est déplacée par τ , une transformée de Fourier localisée dans le temps est effectuée sur la sur la fenêtre du signal original $f(t)$. De cette façon, la TFCT décompose un signal temporel en une représentation de deux dimensions temps/fréquence.

L'analyse avec la TFCT dépend essentiellement de la fenêtre choisie $g(t)$. Cependant il n'est pas possible d'obtenir à la fois une bonne résolution temporelle et une bonne résolution fréquentielle avec la TFCT et un compromis est nécessaire :

- une petite fenêtre sera adaptée à la détection des fréquences élevées, et offrira une bonne résolution temporelle, alors que
- une fenêtre adaptée à la détection des fréquences basses, fournira une résolution temporelle inférieure, mais une meilleure résolution en fréquence.

8.1.3 Transformée en ondelettes

L'analyse en ondelettes corrèle le signal original $f(t)$ avec un ensemble de fonctions obtenues à partir :

- d'une mise à l'échelle (c.-à-d. dilatation et contraction) et
- de décalages (c.-à-d. translations le long de l'axe du temps)

d'une fonction d'ondelette Ψ , appelée *l'ondelette mère*. En comparaison avec la transformée de Fourier, la transformée en ondelettes offre plus de souplesse puisque la fonction d'analyse peut être choisie avec plus de liberté, sans la nécessité d'utiliser des formes sinusoïdales. Contrairement à la TFCT, où la taille de la fenêtre glissante est fixe, la transformée en ondelettes permet d'avoir des tailles de fenêtres variables pour analyser le contenu fréquentiel d'un signal.

Fondamentalement, lorsqu'il s'agit de l'analyse en ondelettes, deux types de transformations peuvent être utilisées :

CWT : transformée en ondelettes continue¹ et

DWT : transformée en ondelettes discrète².

8.1.3.1 Transformée en ondelettes continu (CWT)

La CWT est la somme dans le temps, des versions réduites et décalées de la fonction d'ondelettes Ψ . Lorsqu'elle est appliqué sur le signal original $f(t)$, la CWT est exprimée comme suit :

¹Continuous **W**avelet **T**ransform, dans la littérature anglo-saxonne.

²Discrete **W**avelet **T**ransform, dans la littérature anglo-saxonne.

$$CWT_f(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} f(t) \Psi^*\left(\frac{t - \tau}{s}\right) dt. \quad (8.7)$$

Le calcul de la CWT sur un signal donne de nombreux *coefficients*, qui sont des fonctions du paramètre de décalage τ et du paramètre d'échelle s . En fait, τ est proportionnel aux informations temporelles. Il indique l'emplacement de l'ondelette dans le temps ; en faisant varier τ l'ondelette peut être déplacée sur le signal. Le facteur de dilatation s est inversement proportionnel à la fréquence de l'information. La variation de s modifie non seulement la fréquence centrale de l'ondelette, mais aussi la longueur de la fenêtre. Bien qu'il fournisse une grande précision dans l'analyse du signal, le calcul de la CWT est redondant et très coûteux en temps. Habituellement, le calcul de la CWT est réalisé en prenant les valeurs discrètes pour le paramètre de dilatation s et du paramètre de décalage τ .

8.1.3.2 Transformée en ondelettes discrète (DWT)

La DWT est la version discrétisée du CWT avec un calcul discrétisé des paramètres de dilatation et de décalage. Elle fournit en plus les informations nécessaires pour analyser le contenu de manière fiable d'un signal et aussi d'une manière beaucoup plus rapide comparée aux calculs de CWT.

La DWT peut être appliqué à différents niveaux. Le niveau 1, est essentiellement composée de ce que nous appelons *les ondelette filtres*, qui impliquent deux concepts fondamentaux : les bancs de filtres et un haut-bas échantillonnage.

Les bancs de filtres visent à changer la résolution en séparant le signal en bandes de fréquences.

En fait, un signal discret est d'abord filtré par les filtres L et H qui séparent le contenu fréquentiel du signal analysé en bandes de fréquences de largeurs égales. Les filtres L et H sont donc respectivement des filtres passe-bas et passe-haut. Lorsque le signal passe à travers :

- le filtre passe-haut, seule l'information en haute fréquence est transmise,
- le filtre passe-bas, seule l'information en basse fréquence transmise.

Par conséquent, le signal est effectivement décomposé en deux sous-signaux appelés respectivement :

les détails ou coefficients d'ondelettes de détails et

les approximations ou coefficients d'ondelettes d'approximation.

Les approximations correspondent aux basses fréquences et les détails aux hautes fréquences. Notons que chaque sous-signal contient la moitié du contenu en fréquence, mais un nombre d'échantillons égal à celui du signal original.

Pour une décomposition de niveau p , le contenu des fréquences comprises par les approximations et les détails, notés respectivement par f_{Approx} et f_{Det} , peuvent être déterminés comme suit :

$$f_{Approx} = [0, \frac{f_s}{2^{p+1}}], f_{Det} = [\frac{f_s}{2^{p+1}}, \frac{f_s}{2^p}]. \quad (8.8)$$

tel que f_s est la fréquence d'échantillonnage du signal d'origine. Finalement, le signal d'origine peut être représenté par l'approximation finale relative au dernier niveau, et les détails accumulés correspondants à tous les niveaux.

Le processus peut être élargi à un niveau arbitraire, selon la résolution souhaitée. Une illustration d'une DWT 3-niveaux est montrée dans la figure 8.1.3.2.

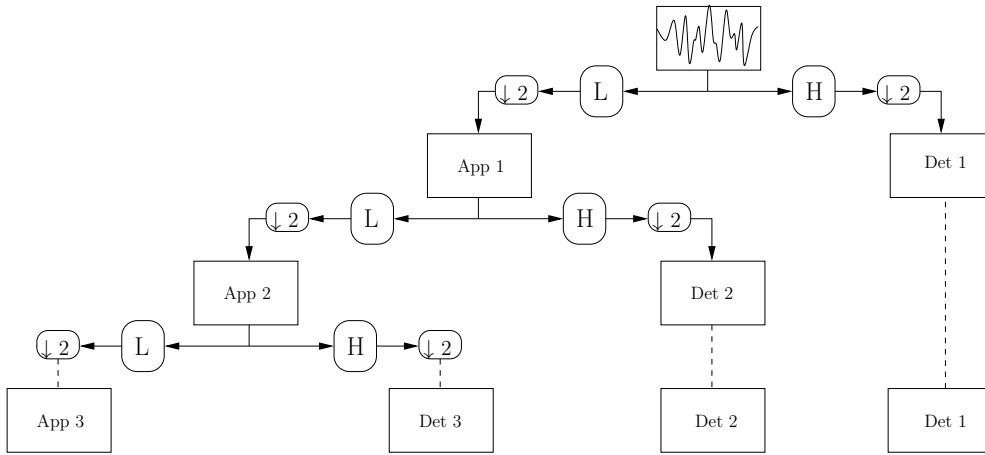


FIG. 8.1 – Une illustration d'une décomposition DWT à 3 niveaux

8.2 Applications aux SCAs

Dans ce qui suit, nous allons montrer comment l'analyse en ondelettes peut être très utile en particulier pour un attaquant par canaux auxiliaires.

8.2.1 Les ondelettes pour la détection des motifs cryptographiques

Dans le domaine des SCAs, le principal défi de l'attaquant est de mieux analyser et d'exploiter les dépendances entre les données manipulées et la fuite. En pratique, il analyse un ensemble de traces dont chacune comporte un bloc d'opérations survenant lors d'un processus cryptographique. En général, les bits de données chiffrées sont divisés en petits blocs de bits, appelés blocs d'exploitation, selon la spécification de l'algorithme cryptographique.

Par exemple, les chiffrements par blocs, comme l'Advanced Encryption Standard (AES), sont liés aux différents modes de fonctionnement (*par exemple* ECB, CBC, CFB,

OFB) qui visent à diviser les données chiffrées et de gérer le mode de fonctionnement des blocs obtenus.

Par ailleurs, dans le cadre des SCAs, l'alignement des traces est une grande préoccupation, car l'attaquant doit être capable de détecter le début et la fin de chaque bloc d'opération. Malheureusement, en pratique, il est presque impossible de recueillir des traces parfaitement alignées en raison de nombreux facteurs.

En particulier, lors d'une attaque réelle, l'attaquant n'a jamais à sa disposition de signal synchronisé avec le chiffrement lui permettant de déclencher l'acquisition de façon synchronisée.

En plus, les traces acquises sont très souvent perturbées par la présence de bruit. Dans une telle situation, nous proposons d'utiliser le CWT pour révéler les informations globales impliquées par le processus cryptographique analysé. À cette fin, nous avons enregistré un signal impliquant l'activité des trois chiffrements AES-128bit en mode CBC.

La figure 8.2.1 est une capture générée à partir de MATLAB (*wavemenu toolbox*). En haut de cette figure, nous avons le signal d'origine acquis par l'oscilloscope. On constate que le signal est perturbé par une quantité élevée de bruit et aucune information sur le processus de chiffrement ne peut être révélée.

Au centre de la figure, nous pouvons voir la représentation en deux dimensions de la CWT. On constate maintenant que, grâce à cette représentation, nous pouvons facilement détecter les limites des trois blocs AES mesurés.

Nous avons ajouté des lignes pointillées pour mettre en évidence ces limites. Notons que ces limites sont détectées pour des échelles élevées (basses fréquences) de CWT. Par ailleurs, nous pouvons révéler plus de détails sur le contenu de l'information du signal, tels que le nombre de tours composants chaque bloc de l'AES.

Cela peut être très utile à l'attaquant, spécialement lorsque l'algorithme analysé est inconnu.

En outre, comme indiqué au bas de la figure, l'outil MATLAB génère une forme approximative (c.-à-d. extrait l'information globale) du signal analysé sur la base des coefficients de CWT.

8.2.2 L'information mutuelle et les ondelettes pour filtrage des traces

8.2.2.1 Filtrage des traces avec les ondelettes

L'utilisation de la transformation en ondelettes dans des procédures de débruitage des canaux auxiliaires a été proposée par Pelletier *et al.*[60]. Les auteurs ont proposé une méthode très générale, basée sur la transformée en ondelettes discrète, pour supprimer le bruit des signaux sans explications détaillées.

Si les coefficients de détails sont assez petits, ils peuvent être omis sans affecter sensiblement les principales caractéristiques du signal. Comme ces petits détails sont

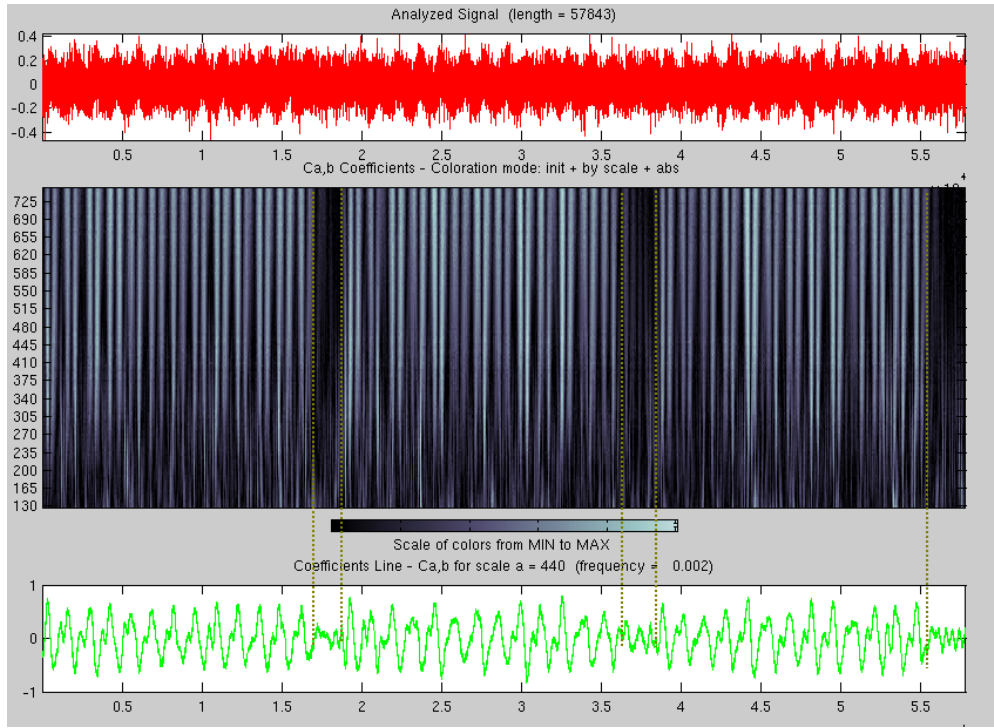


FIG. 8.2 – Illustration de la CWT sur un triple chiffrement AES.

souvent ceux associés au bruit leur omission revient essentiellement à enlever le bruit. Cela devient l'idée principale du *seuillage* de toutes les fréquences qui sont inférieures à un seuil particulier. La valeur du seuil est estimée en utilisant la formule universelle de Donoho [45]. Le seuil λ_{Donoho} de Donoho, peut être calculé comme suit :

$$\lambda_{Donoho} = \sigma \sqrt{2 \log(L)}, \sigma = \frac{\text{median}|\text{details}|}{0.6745} \quad (8.9)$$

$$Th_{hard}(x) = \begin{cases} x & \text{if } |x| > \lambda_{Donoho} \\ 0 & \text{if } |x| \leq \lambda_{Donoho} \end{cases} \quad (8.10)$$

où σ et L sont respectivement la variance du bruit et la longueur des coefficients de détails. Ce seuil est appliqué seulement sur les coefficients de détails pour un niveau d'échelle en ondelettes spécifique. Un nouveau seuil est calculé pour chaque échelle et utilisé dans une fonction de seuillage.

En effet, il y a deux types de fonctions de base de seuillage :

un seuillage fort Th_{hard} , défini dans l'équation (8.10), qui fixe à zéro tous les coefficients de détails qui sont en dessous de la valeur seuil λ_{Donoho} , et

un seuillage lisse Th_{soft} , défini dans l'équation (8.11), pour lequel les coefficients de détails avec des amplitudes plus petites que λ_{Donoho} sont mis à zéro.

Mais les coefficients retenus sont aussi réduits par la valeur du seuil, afin de diminuer l'effet du bruit.

$$Th_{soft}(x) = \max(|x| - \lambda_{Donoho}, 0) \quad (8.11)$$

En comparaison avec notre méthode de seuillage proposée dans la section 5.3, celle-ci est basée sur les coefficients de détails des ondelettes alors que notre méthode est basée sur les vecteurs propres.

8.2.2.2 Amélioration du filtrage du bruit

Dans le but d'utiliser le seuil de Donoho dans le contexte SCA, une idée simple consiste à l'appliquer sur les vecteurs propres obtenus après un traitement par une transformée en ondelettes comme ce qui a été fait dans la section 5.3. Une autre alternative serait de l'appliquer sur les courbes d'information mutuelles. Dans cette optique, le seuil de Donoho peut être utilisé comme un outil complémentaire afin d'améliorer l'attaque.

Par exemple, la nouvelle information mutuelle, basée sur le seuil de Donoho, $\lambda_{IM_{Donoho}}$, peut être exprimé comme le produit du seuil de Donoho λ_{Donoho} par la valeur de l'information mutuelle $IM(O; L)$, calculé sur l'ensemble des traces acquises. Ainsi $\lambda_{IM_{Donoho}} = \lambda_{Donoho} \cdot IM(O; L)$. De toute évidence, le nouveau seuil, $\lambda_{IM_{Donoho}}$ vise à écarter les échantillons de temps les plus bruyants et favoriser uniquement les échantillons qui sont liés à l'information sensible. Une illustration du calcul de l' IM est montrée dans la figure 8.2.2.2. L' IM est calculée sur la décomposition de premier niveau DWT d'une mise en œuvre non protégée du DES (500 traces utilisées).

8.2.3 Les ondelettes pour la compression des traces

Pratiquement, la plupart des algorithmes de chiffrement peuvent être cassés par des attaques si l'adversaire a assez de temps et de ressources. Par conséquent, un objectif plus réaliste de contrer les SCAs serait de faire de l'obtention des informations un travail trop intensif. Dans la littérature des SCAs, la majorité des mesures de sécurité sont basées sur le nombre de traces pour évaluer la robustesse d'une application cryptographique contre les attaques par canaux auxiliaires. Pour ce faire, l'adversaire est tenu de recueillir le plus possible de traces, pour avoir une meilleure corrélation entre les traces et les variables sensibles [48, 49]. En pratique, l'acquisition des traces est généralement effectuée par un oscilloscope et exige une haute capacité de stockage, en particulier lorsque des contre-mesures sont mises en œuvre. Aussi, il serait préférable, lors d'attaques en aveugle, d'avoir toute la quantité d'information divulguée par le procédé de chiffrement. Ceci exige une haute capacité de stockage lors de la prise en compte de toute la longueur des traces acquises. Par ailleurs, avec un taux d'échantillonnage élevé plus de ressources de mémoire, de temps, et de complexité de calculs sont nécessaires. Une des solutions

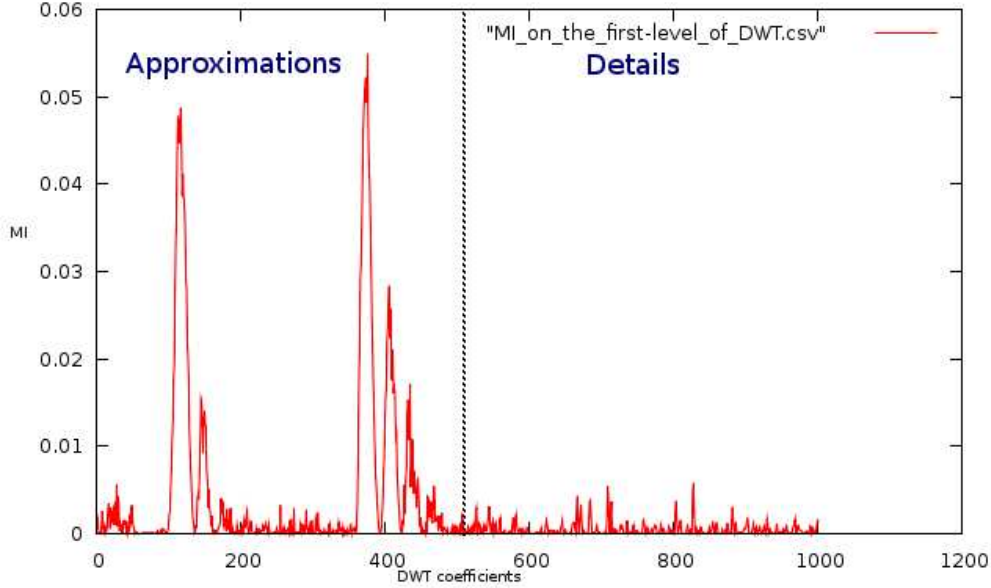


FIG. 8.3 – Calcul de l'IM avec DWT au premier niveau.

les plus efficaces est d'utiliser la transformée en ondelettes discrète pour la compression des signaux contenant une grande quantité d'informations. Ceci permettra d'éliminer les redondances du signal et de ne garder que les informations utiles en temps réel (*c. – d.* avant stockage). Dans la littérature, la compression des signaux basée sur les ondelettes est essentiellement réalisée à l'aide de la notion de *seuil*, détaillée précédemment.

Du point de vue SCA, nous proposons de ne garder que les coefficients d'approximation et de rejeter les détails. En effet, pour les mises en œuvre cryptographiques modernes, les informations sensibles sont traitées au niveau des périodes d'horloge, dont l'activité est représentée par les basses fréquences du signal (*c. – d.* les coefficients d'approximation). Pour un niveau de décomposition p , le nombre d'échantillons conservés pour chaque trace est $\frac{l}{2^{p+1}}$, où l est la longueur de la trace. Par conséquent, la compression des traces est effectuée sans perte d'information lorsqu'on considère uniquement les approximations. En effet, il a été souvent signalé que la perte d'information ne peut arriver que lors de la reconstruction du signal. Dans cette perspective, il est fortement recommandé de faire un pré-traitement des signaux grâce aux ondelettes avant de passer à l'étape du choix des points d'intérêt.

8.2.4 Les ondelettes pour retrouver les clés

En SCA, l'intérêt est de maximiser l'efficacité de l'attaque, sachant que les traces sont une ressource rare. Comme indiqué précédemment, les ondelettes font une séparation dans le contenu du signal pour le représenter par un enchaînement des approximations du dernier niveau avec les détails. Nous proposons d'effectuer une analyse par canaux auxiliaires directement sur les coefficients d'ondelettes, qui sont les approximations et les détails. En d'autres termes, la fuite n'est plus représentée par les échantillons temporels des signaux acquis mais par la valeur des coefficients d'ondelettes. De cette façon, nous pouvons améliorer l'efficacité de l'attaque.

8.2.4.1 Applications des ondelettes et résultats

La première expérimentation a été de comparer l'efficacité d'un adversaire utilisant cette technique basée sur les ondelettes (1 seul niveau) en complément d'une attaque *template* avec celle d'un adversaire utilisant une attaque *template* brute. L'ACP est toujours utilisée pour le choix des points d'intérêt. Ainsi, nous avons utilisé des traces issues d'une campagne non protégée d'un chiffrement DES, et dont le niveau de bruit est minimal. Nous avons choisi le modèle de la distance de Hamming déjà testé dans le chapitre 5. Le but est de savoir s'il y a un gain significatif en nombre de traces, dans ces conditions idéales. Nous avons aussi tenu à inclure les résultats d'une attaque *template* basée uniquement sur les fréquences en utilisant la FFT. Les métriques utilisées sont le taux de succès et l'entropie estimée. Les figures 8.4 et 8.5 montrent que l'attaque basée sur les ondelettes est nettement meilleure que les attaques basées sur la FFT, ou l'attaque brute. Le taux de succès et l'entropie estimée deviennent rapidement stables. Aussi nous remarquons que l'attaque basée sur la FFT donne de moins bons résultats par rapport à l'attaque brute.

Nous tenons à signaler que ces résultats sont obtenus dans des conditions où le bruit est minimal.

Pour voir l'effet du traitement sur des traces bruitées, nous avons ajouté du bruit gaussien aux mêmes traces utilisées dans la première expérimentation, et avons tenté de comparer les taux de succès et les entropies estimées pour différents niveaux de décompositions, et en faisant l'attaque sur ces coefficients. Aussi nous voulions savoir à partir de quel niveau il n'y a plus d'apports en termes d'efficacité.

Les figures 8.2.4.1 et 8.2.4.1 montrent que l'utilisation de ces techniques apporte un gain réel. Nous remarquons aussi que l'analyse en ondelettes surpasse l'analyse standard sur les traces originales et ce pour tous les niveaux de décomposition. En effet, sans l'utilisation des ondelettes, nous ne pouvons pas classer la vraie clé après utilisation de 1000 traces alors qu'avec l'utilisation des ondelettes nous arrivons à un taux de succès de 100% avec le niveau 4 seulement avec 900 traces. Par ailleurs, les performances de l'analyse s'améliorent avec l'augmentation du niveau. En principe, plus de niveaux sont utilisés, plus nous avons de performances.

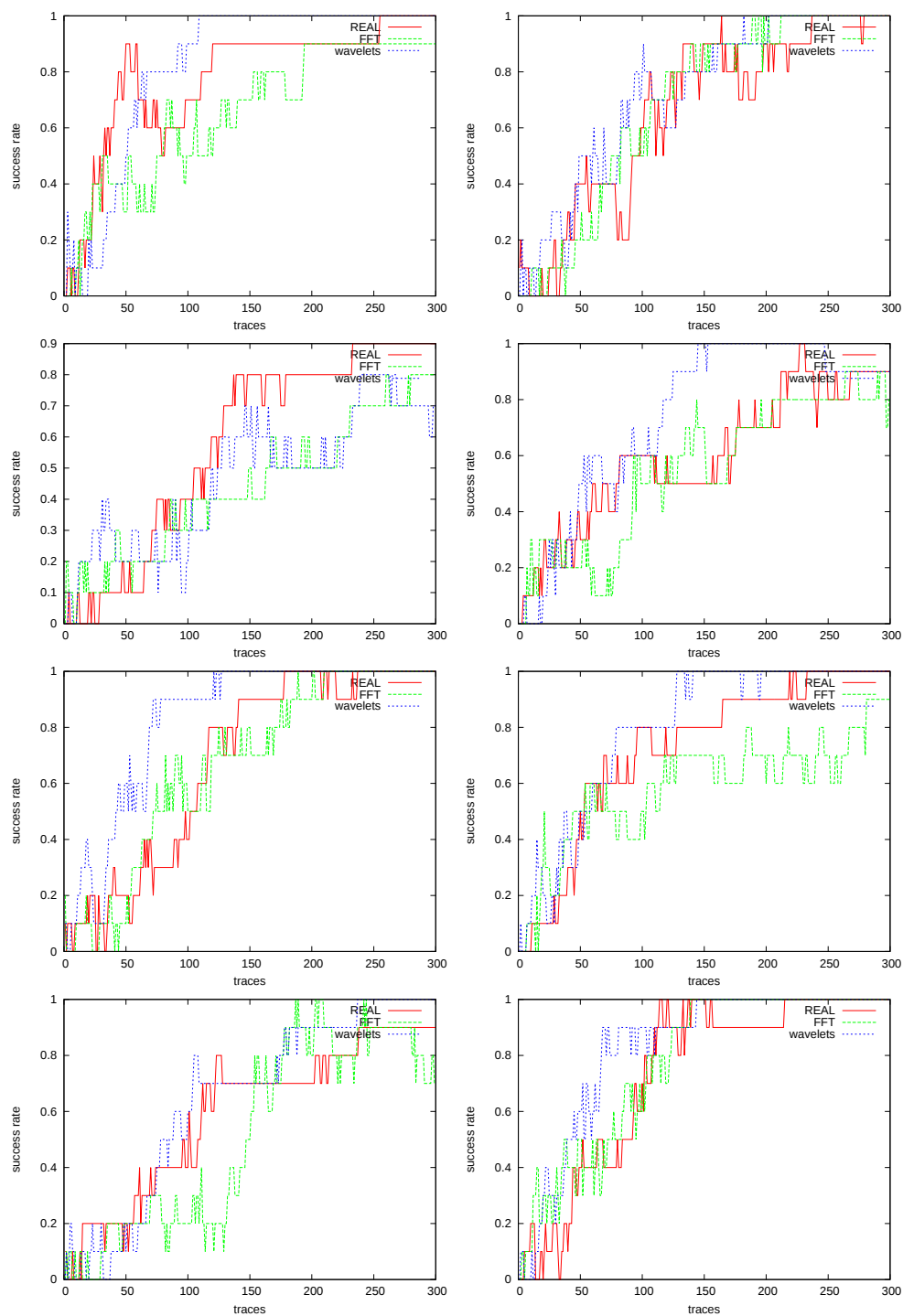


FIG. 8.4 – Comparaison du taux de succès pour les 8 S-box

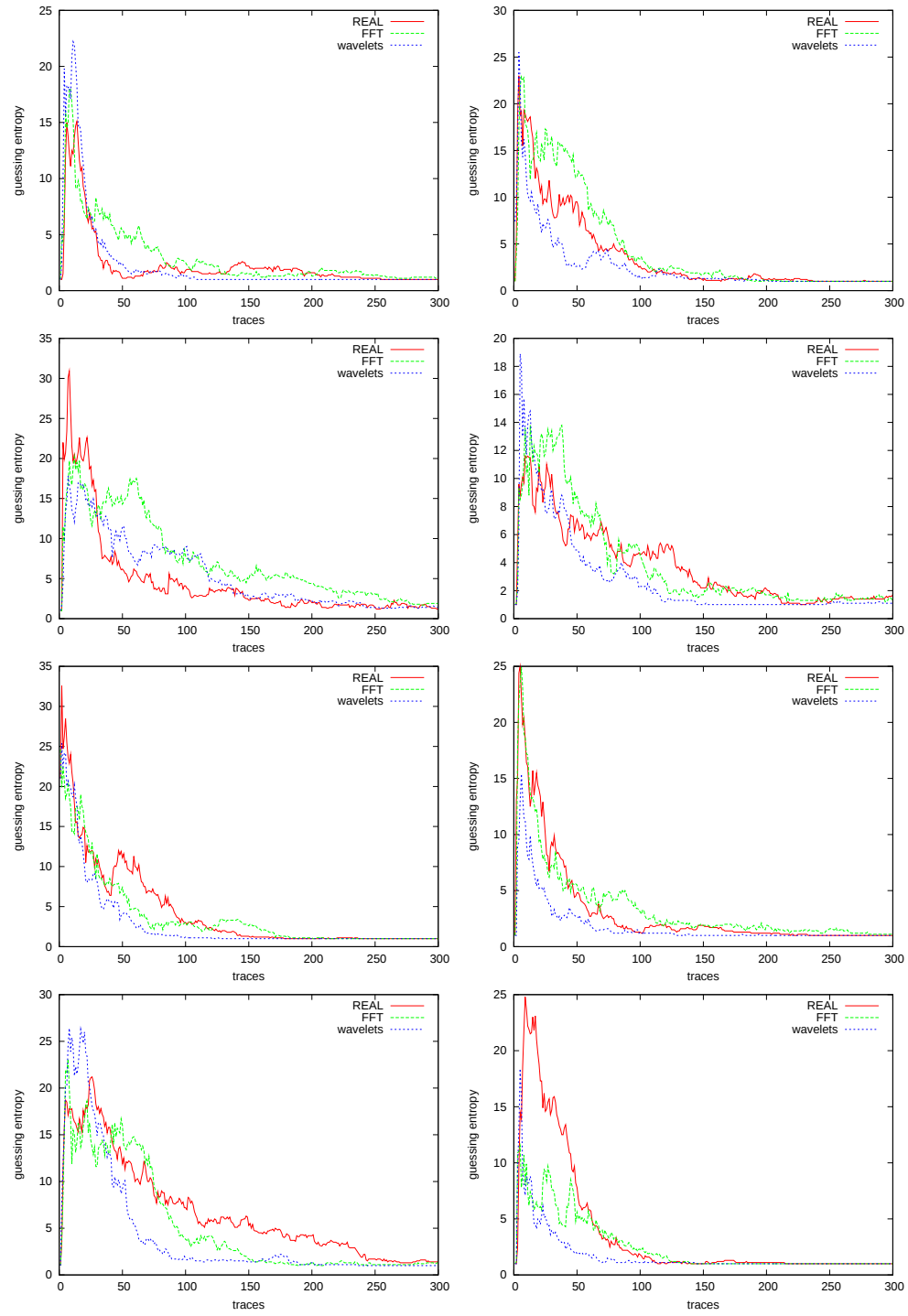


FIG. 8.5 – Comparaison de l'entropie estimée pour les 8 S-box

Néanmoins, l'énergie ne peut être centralisée à l'infini. Ainsi, il y a toujours un niveau limite de décomposition pour la transformation en ondelettes. Au-delà de ce niveau, la performance ne peut être augmentée. Les figures montrent qu'il n'y a pas une grande différence entre le niveau 4 et 5.

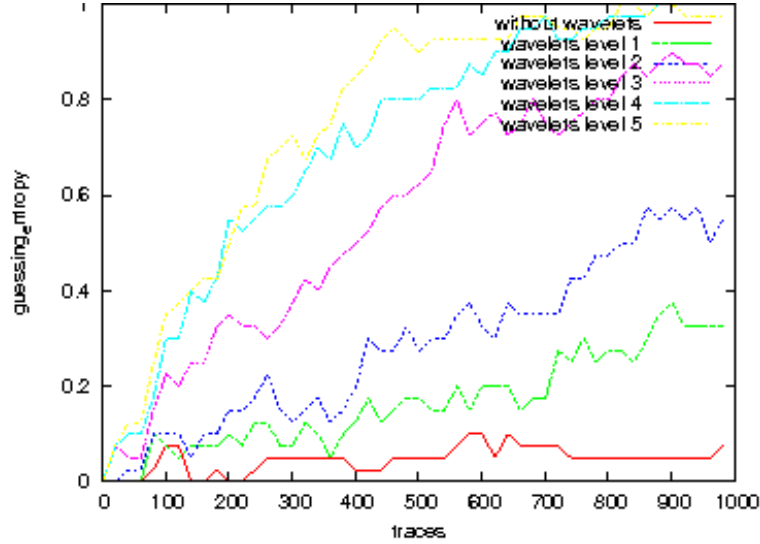


FIG. 8.6 – Taux de succès et attaques *template* avec ondelettes

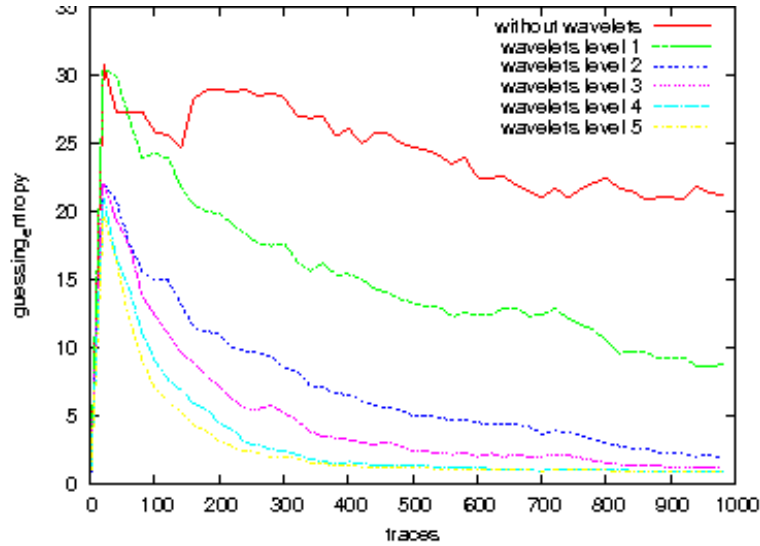


FIG. 8.7 – Entropies estimées et attaques *template* avec ondelettes

Dans le but de voir si nous pouvions obtenir les mêmes améliorations pour d'autres

attaques, nous avons testé une CPA effectuée sur des traces réelles. Ce sont les mêmes traces utilisées dans les figures 8.4 et 8.5.

Cette fois, nous avons appliqué la CPA sur le premier niveau de décomposition en ondelettes et voulions voir le comportement des métriques avec ou sans utilisation des *détails* des ondelettes. Ainsi, nous avons utilisé les approximations et les détails (*AppandDet*), d'une part, et, d'autre part, uniquement les coefficients d'approximation (*app*).

Le taux de succès et l'entropie estimée sont représentés sur les figures 8.2.4.1 et 8.2.4.1, qui montrent que l'analyse des traces à l'aide des ondelettes est clairement plus efficace que l'analyse de base dans le domaine temporel. Ces résultats sont assez ressemblants avec ceux obtenus avec les attaques *template*.

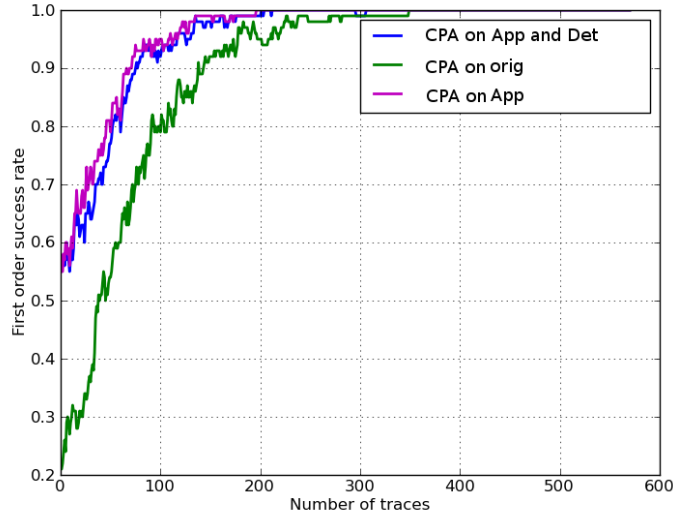


FIG. 8.8 – Taux de succès de la CPA et ondelettes

Dans la perspective d'évaluer de manière fiable la robustesse des dispositifs sécurisés contre les attaques par canaux auxiliaires, quatre aspects de l'analyse sont essentiellement pris en considération :

- l'acquisition des traces,
- le pré-traitement des traces,
- la détection et l'extraction de la structure de chiffrement à partir des traces pré-traitées, et
- la récupération d'informations secrètes.

Dans ce chapitre, nous fournissons à l'évaluateur de nouvelles applications de la transformée en ondelettes dans le contexte SCA. Nous avons montré comment la transformée en ondelettes peut être utile pour réduire la capacité de stockage avec la compression. Ceci, tout en gardant la maximum d'informations. Par ailleurs, nous avons mis en évi-

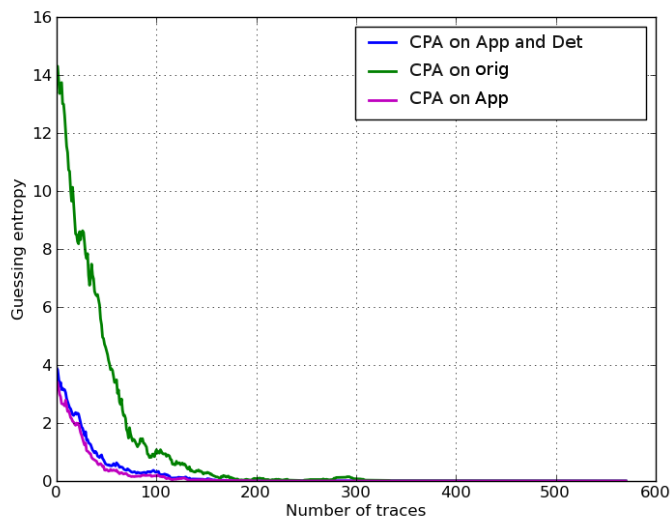


FIG. 8.9 – Entropies estimées de la CPA et ondelettes

dence comment détecter et d'extraire les motifs du processus cryptographique analysé. Nous avons aussi montré que l'analyse en ondelettes est non seulement utilisée pour des fins de pré-traitement mais aussi dans le cœur même de l'attaque. La figure 8.2.4.1 montre les possibles implications des applications proposées dans tous les aspects SCAs.

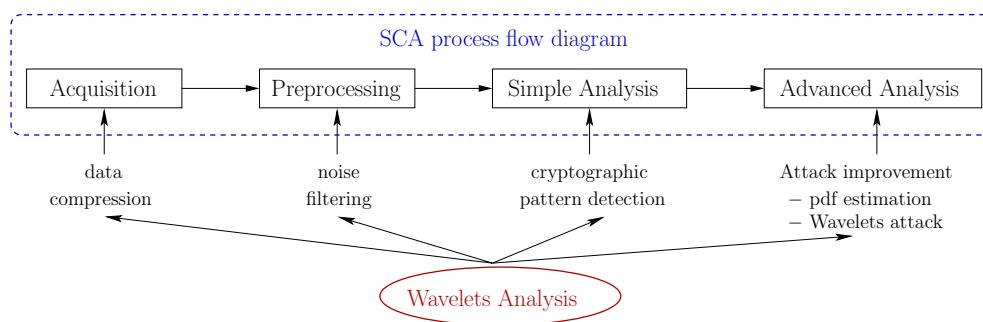


FIG. 8.10 – Les ondelettes et les SCAs

Chapitre 9

Attaques aveugles et différents obstacles

Nous décrirons dans ce chapitre les difficultés techniques que nous avons rencontrées. Nous essayons d'aller plus loin dans nos analyses sur les fuites que nous appelons « exotiques ». EN particulier, nous étudions la possibilité de les voir exploitées pour des attaques.

9.1 Attaques sur *DPA contest* 2

9.1.1 Présentation du *DPA contest*

Le *DPA contest* est un concours international organisé par le groupe COMELEC de Telecom-Paristech. Le but de ce concours est de permettre aux chercheurs de comparer leurs attaques. Ce concours a été officiellement ouvert en 2008 à la conférence CHES. Depuis, il a été renouvelé chaque année sous différentes formes, et avec différents défis.

9.1.2 Plate-forme d'attaque *template*

Les résultats exposés dans ce document ont été le fruit de l'élaboration d'une plate-forme spécialisée dans les attaques *templates*. Grâce à cette plateforme, qui a été améliorée régulièrement, nous avons pu évaluer différents circuits à partir de traces issues d'une première étape d'acquisitions. Les traces ont été de différents types (EM ou consommation) et le mécanisme consiste à tester tous les modèles connus de consommation, en poids ou en distance de Hamming.

Notre plateforme peut examiner pratiquement tous les niveaux des algorithmes de chiffrements DES, 3-DES et AES. Durant, cette thèse, nous avons eu l'occasion de tester les performances de nos attaques dans le cas du *DPA contest*. En effet, l'initiative de ce concours a été l'occasion de se mesurer face à d'autres assaillants.

9.1.3 Résultats de l'attaque template au concours *DPA contest*

Les acquisitions à partir d'une mise en œuvre d'AES-128 ont été réalisées sur une carte Sasebo GII dont le design a été entièrement fourni.

Notre modèle le plus convaincant fut la distance de Hamming entre les deux derniers tours. Ce modèle est défini dans l'équation suivante :

$$\mathcal{L} = HW(\text{état}_9[b] \oplus \text{chiffré}[b]) \text{ où } b \in [0, 16[\text{ est l'indice de la boîte de substitution.} \quad (9.1)$$

L'attaque *templates* ayant bénéficié d'un long profilage sur un million de traces partait favorite. Malgré cet avantage, nous n'arrivions pas avoir un taux de succès général dépassant les 80%. En effet, le taux de succès général est défini par la capacité de l'adversaire à récupérer les 128 bits de la clé entière. Le taux de succès partiel correspond aux sous-clés aux entrées des *S-boxes*. Les résultats des métriques sont détaillés dans les tableaux A.3 et A.2 et la figure A.1 et expliquent que notre difficulté venait des sous clés entrant dans les *S-boxes* $\{1, 5, 9 \text{ et } 13\}$, qui résistaient à nos différentes tentatives. Le point *positif* venait des autres *S-boxes* qui étaient facilement cassables. Ces expérimentations ont apporté beaucoup d'enseignements. Elles ont, entre autres, mis en évidence le fait que des *S-boxes* identiques dans leur fonction peuvent ne pas l'être pour ce qui est de leurs caractéristiques face à une attaque SCA.

9.1.4 Comparaison avec l'attaque stochastique

9.1.4.1 L'attaque stochastique

L'équipe de recherche de CASCADE à Darmstadt a contribué au *DPA contest* en présentant une attaque dite *attaque stochastique*[? , ???] Cette attaque a été classée première au niveau du taux de succès général.

Comme les attaques *templates*, les attaques stochastiques [71] comportent deux étapes : une étape d'entraînement et une étape d'attaque proprement dite. Cependant, à la différence des attaques *template*, la phase d'entraînement de l'attaque stochastique ne nécessite qu'une seule clé. Le modèle de consommation à un instant t est donné dans l'équation suivante :

$$W_t(x, k) = h_t(x, k) + \mathcal{B}_t \quad (9.2)$$

où :

- x est le texte clair variable,
- k est la clé fixe,
- h_t est la partie déterministe de la consommation en courant (qui dépend de x et k) et
- $\mathcal{B}_t$ est un bruit aléatoire avec une espérance nulle ($\forall t, \mathbb{E}(\mathcal{B}_t) = 0$).

L'étape du profilage consiste en une approximation de h_t , suivie par une estimation

de $\mathcal{B}_t$ grâce à h_t . Soit $\tilde{h}_t$ la meilleure estimation de h_t calculée de la façon suivante :

$$\tilde{h}_t(x, k) = \beta_0 + \sum_{i=1}^u \beta_{it} g_i(x, k) \quad (9.3)$$

où :

- les g_i sont des fonctions de base choisie, qui dépendent de x et k et
- $\{\beta_{it}\}$ sont des coefficients, qui estiment l'influence de la fonction g_i associée dans la consommation.

C'est le choix de fonctions de base qui définit le degré des modèles stochastiques. Un modèle linéaire prend seulement des bits particuliers alors qu'un modèle de degré supérieur considère plusieurs bits pour chaque coefficient. g_0 peut être supposé égal à 1.

La deuxième étape de la phase du profilage consiste à caractériser le bruit. D'abord, il faut sélectionner des instants pertinents par des méthodes comparables à celles exposées au chapitre 5. Le bruit est caractérisé par la construction de la fonction de densité de probabilité de la distribution normale multivariée, utilisant une matrice de covariance (calculée avec une variable aléatoire de bruit associée à chaque point d'intérêt).

De même que celle de l'attaque *template*, la phase d'attaque en ligne de l'attaque stochastique utilise le principe du maximum de vraisemblance.

9.1.4.2 Comparaison des attaques *template* et stochastique

En se basant sur les métriques de comparaison et malgré l'avantage visible de l'attaque stochastique pour ce qui est du taux de succès général, notre attaque fut la meilleure pour 12 sous-clés parmi les 16, comme le montre la figure 9.1, dans laquelle nous comparons notre attaque (nous montrons ici uniquement deux *S-boxes*) au niveau du taux de succès partiel.

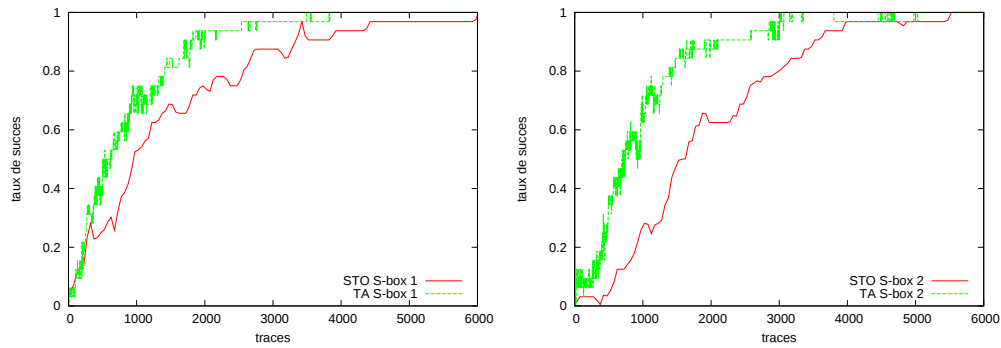


FIG. 9.1 – Comparaison des taux de succès entre l'attaque *template* et l'attaque stochastique pour les *S-boxes*[s] 2 et 3

Notre interprétation est que les 4 *S-boxes* résistantes ont bénéficié d'une mise en œuvre différente des autres *S-boxes* car elles sont toutes sur la première ligne dans la

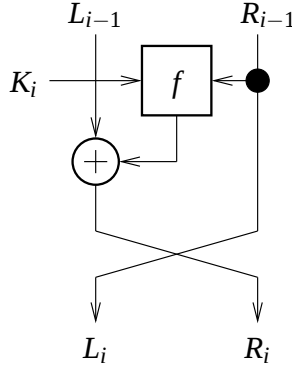


FIG. 9.2 – Bloc Feistel

matrice d'état. Cette ligne n'est pas concernée par la transformation ShiftRows comme expliqué dans le chapitre 1.3.

Tous nos résultats et les résultats des autres participants au concours sont visibles sur [79]().

Parmi nos perspectives, nous tentons d'améliorer le taux de succès général et d'expliquer les raisons techniques de cette contrainte.

9.2 Attaques et bruit exotique

Les équipement sécurisés doivent être protégés contre les attaques par canaux auxiliaires. Néanmoins, les fuites involontaires de certains appareils peuvent être relativement inattendues dans la pratique. Dans cette section, nous commençons par illustrer plusieurs modèles de fuite inattendues que nous appellerons fuites *exotiques* (basés sur des mesures réelles).

9.2.1 Comment situer la fuite ?

Comme déjà discuté dans ce rapport, la fuite se produit en général en distance de Hamming. Néanmoins, comme observé dans [19], des modèles de fuite beaucoup plus simples, basés sur des valeurs des variables intermédiaires, peuvent être exploités dans des attaques. Par exemple, pour un schéma de Feistel 9.2, nous pouvons localiser la fuite en distance de Hamming de la manière suivante :

$$\mathcal{L}(LR_0, LR_1) \sim \text{HW}(LR_0 \oplus LR_1) + \mathcal{N}(0, \sigma^2)$$

où σ^2 est la variance du bruit gaussien ajouté à la fuite déterministe.

La fonction non linéaire de DES est donnée par $f(R_0, K_1) = P \circ S(E(R_0) \oplus K_1)$; alors la fuite en distance de Hamming est déduite par :

$$\mathcal{L}(LR_0, LR_1) \sim \sum_{i=1}^{32} L_i \oplus R_i \quad (9.4)$$

$$+ \sum_{s=1}^8 \text{HW} (S_s(E(R_0) \oplus K_1) \llbracket 6(s-1) + 1, 6s \rrbracket) \quad (9.5)$$

$$\oplus P^{-1}(L_0 \oplus R_0) \llbracket 4(s-1) + 1, 4s \rrbracket + \mathcal{N}(0, \sigma^2) \quad (9.6)$$

On peut voir clairement d'après cette équation que 9.4 n'est pas une variable sensible puisqu'elle ne dépend pas de la clé, et que la partie sensible est 9.5.

En nous concentrant sur une *S-box* $s \in \llbracket 1, 8 \rrbracket$ nous avons :

$$\mathcal{L}(S\text{-box } s) \sim \text{HW} \left(\begin{array}{c} S_s(E(R_0) \oplus K_1) \llbracket 6(s-1) + 1, 6s \rrbracket \\ \oplus P^{-1}(L_0 \oplus R_0) \llbracket 4(s-1) + 1, 4s \rrbracket \end{array} \right) + \mathcal{N}(60/2, 60/4 + \sigma^2)$$

Le dernier terme vient du fait que l'espérance d'un bit aléatoire B est égale à $1/2$ et que sa variance est donc :

$$\text{sum}_{b \in \{0,1\}} P[B = b] \times (b - 1/2)^2 = 1/2 \times (1/4 + 1/4) = 1/4$$

Ainsi le bruit peut être résumé comme suit :

- $(64 - 4)/4 = 60/4$: bruit algorithmique
- σ^2 : bruit de mesures

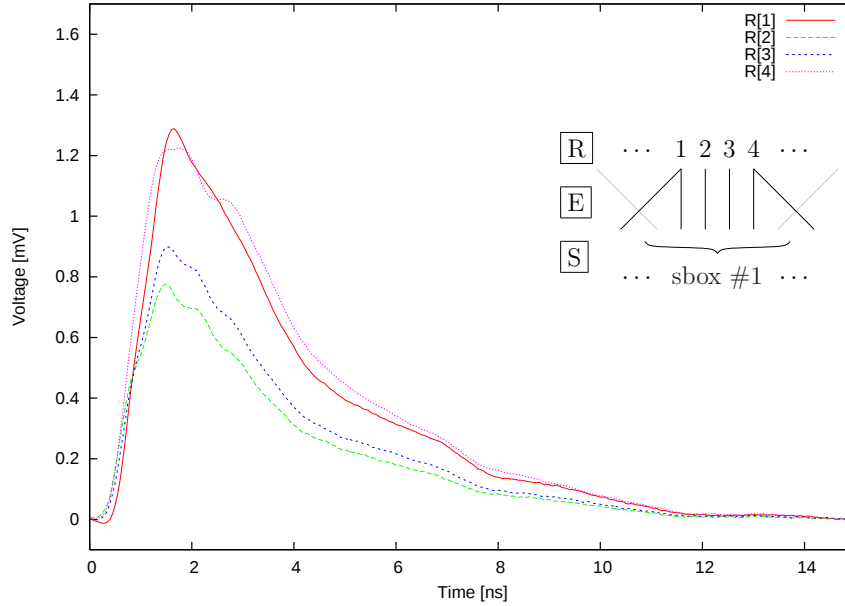
Pour un ASIC, $60/4 \gg \sigma^2$ alors que pour un FPGA, $60/4 \ll \sigma^2$

- Dans cet exemple, le modèle de fuite utilise $4 \times 2 + 6 = 14$ **variables** provenant du texte clair (4 initiales & 4+6 finales) :
 - $R_{\{32,1,2,3,4,5\}}$ d'un côté, et
 - $R_{P^{-1}\{1,2,3,4\}}$ et $L_{P^{-1}\{1,2,3,4\}}$ de l'autre. Ils sont aussi égaux à $R_{\{9,17,23,31\}}$ et $L_{\{9,17,23,31\}}$.
- Ainsi, ces variables vont contribuer dans la fuite du modèle de la distance de Hamming.

Dans cet environnement, une caractérisation entière de la fuite du chiffrement DES prendra $2^{64} \times 2^{56} \times n$ traces, où n est le nombre de chiffrements nécessaires pour tenir compte d'une fuite donnée.

9.2.2 Les disproportions des bits de données

Le chemin de données peut influencer les fuites d'une mise en œuvre. En effet, tous les bits du registre cible ne sont pas identiques, du point de vue consommation. Par exemple, les bits d'entrée d'une *S-box* DES n'ont pas la même intensité. Comme expliqué dans [30], la fonction d'expansion du DES induit un *fan-out* supplémentaire pour les bits $R[1]$ et $R[4]$ en entrée de la première *S-box*.

FIG. 9.3 – Consommation des quatre bits de sortie de la *S-box* #1

Ceci est illustré sur la figure 9.3, qui correspond aux traces différentielles pour chacun des quatre bits $j \in \{1, 2, 3, 4\}$ de sortie de la première *S-box* de DES. Ces valeurs sont issues de traces de consommations où la clé est fixée à $\{0x01\}^8$. Le choix de cette clé permet d'annuler l'activité qui correspond au *key schedule*. Par conséquent, l'activité du chemin de données est proprement isolée de celle due à l'accès à la clé.

La connaissance de ce détail de l'architecture du DES, peut avantager un attaquant et l'aider à peaufiner son apprentissage.

9.2.3 Investigations approfondies sur les bits d'entrée d'une *S-box* DES

La figure 5.4 du chapitre 5 montre que la fuite dépend du modèle choisi. La distance de Hamming localise mieux la fuite alors que les modèles basé sur les valeurs des entrées ou sorties des *S-box* offrent des fuites répétitives sur les tours. Quelques questions se posent alors :

- L'attaquant peut-il tirer profit de ces fuites ?
- D'où viennent elles ?
- Dans le cas où elles dégradent le taux de succès de l'attaque, peut on en déduire des contremesures ?

Nous avons vu tout au long de cette thèse que l'ACP permet de donner une caractérisation générale des fuites dans un ensemble de traces. Les vecteur propres, notamment

ceux qui correspondent à la plus grande valeur propre, peuvent être utilisés par un adversaire.

Dans le but d'analyser cette fuite étalée sur les cycles d'horloge, nous éliminons la partie qui correspond au chiffrement et *key schedule* dans les traces et gardons uniquement la fenêtre à étudier ([7500 :10500]) pour une attaque *template*. Le modèle choisi est la valeur d'entrée de la première *S-box* et il permet de donner, pour les deux cas les vecteurs propres suivants 9.4 :

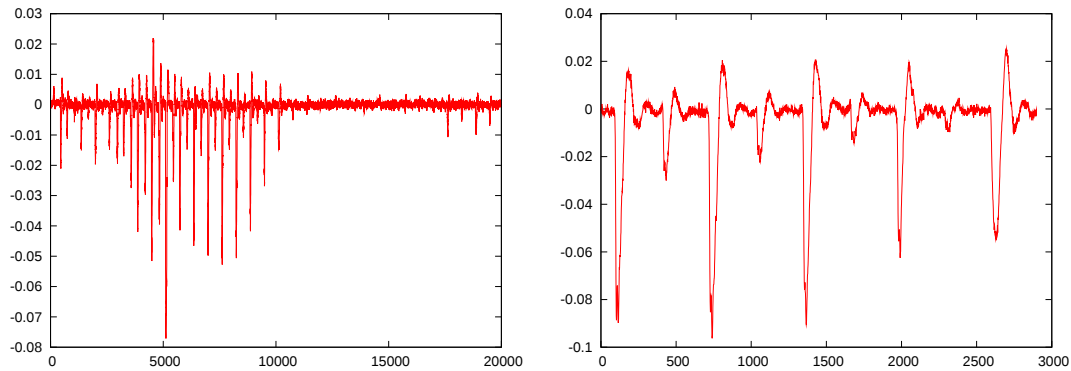


FIG. 9.4 –]

Vecteurs propres : trace entière (à gauche) et fenêtre [7500 :10500] (à droite)

Une attaque utilisant uniquement ces premiers vecteurs propres donne les résultats de l'entropie estimée suivants :

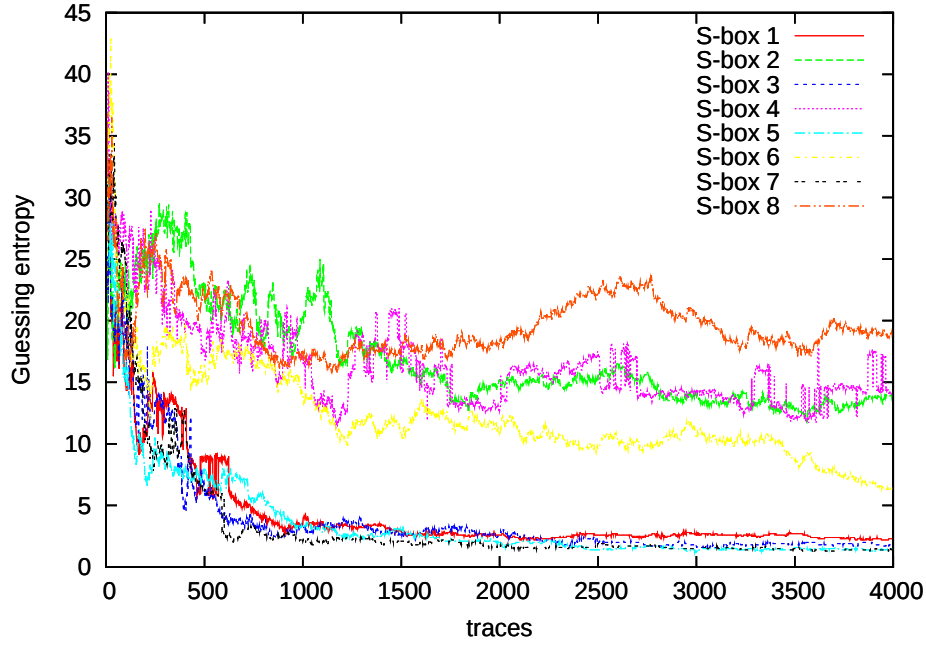
Dans la figure 9.2.3 nous montrons l'effet de l'attaque sur toutes les *S-boxes* et ces résultats montrent que les fuites en dehors des périodes de chiffrement peuvent être exploitées par la moitié des *S-boxes*. Cette fuite est donc liée à une variable sensible dans les *S-boxes* impaires $\{1,3,5,7\}$. Une question se pose alors : cette disparité entre ces *S-boxes* est-elle la conséquence d'une imprudence dans la mise en œuvre ou due à la conception mathématique des fonctions ?

Pour avoir une idée plus précise, nous approfondissons notre analyse à chaque bit en entrée des *S-boxes*. Les vecteurs propres dans les figures A.2 et A.3 précisent que les bits 1 et 2 de la *S-box* 1 sont porteurs de la même information donnée par les bits 5 et 6 de la *S-box* 2. La même observation peut être faite pour toutes les *S-boxes*. Elle est inhérente à la structure de la fonction d'expansion de l'algorithme DES.

On remarque également qu'il y a :

- une fuite répétitive sur 8 cycles d'horloge **avant** le début du chiffrement pour les bits 6 (*S-boxes* impaires) et les bits 2 (*S-boxes* paires), et
- une fuite répétitive sur 8 cycles d'horloge **après** le début du chiffrement pour les bits 5 (*S-boxes* impaires) et les bits 1 (*S-boxes* paires)

Ces fuites peuvent être expliquées par l'activité dans le registre "IF" (voir la section 2.1) qui charge et décharge les données à partir de/vers la mémoire de 8 bits. En effet,



les 8 cycles qui précèdent le chiffrement, nous avons :

$M_2 = L_8$	remplace $KEY_{58} \mapsto CD0[9] \mapsto CD1[8] \mapsto K[18]$
$M_4 = L_{16}$	remplace $KEY_{60} \mapsto CD0[25] \mapsto CD1[24] \mapsto K[4]$
$M_6 = L_{24}$	remplace $KEY_{62} \mapsto CD0[37] \mapsto CD1[36] \mapsto K[46]$
$M_8 = L_{32}$	remplace $KEY_{64} \mapsto$ Parity
$M_1 = R_8$	remplace $KEY_{57} \mapsto CD0[57] \mapsto CD1[56] \mapsto K[40]$
$M_3 = R_{16}$	remplace $KEY_{59} \mapsto CD0[17] \mapsto CD1[16] \mapsto K[18]$
$M_5 = R_{24}$	remplace $KEY_{61} \mapsto CD0[45] \mapsto CD1[44] \mapsto K[18]$
$M_7 = R_{32}$	remplace $KEY_{63} \mapsto CD0[29] \mapsto CD1[28] \mapsto K[8]$

Alors, pendant les 8 cycles qui suivent le début du chiffrement :

$M_{58} = L_1$	est remplacé par une constante.
$M_{60} = L_9$	est remplacé par une constante.
$M_{62} = L_{17}$	est remplacé par une constante.
$M_{64} = L_{25}$	est remplacé par une constante.
$M_{57} = R_1$	est remplacé par une constante.
$M_{59} = R_9$	est remplacé par une constante.
$M_{61} = R_{17}$	est remplacé par une constante.
$M_{63} = R_{25}$	est remplacé par une constante.

Nous pouvons voir que des bits du message viennent régulièrement remplacer des bits de la clé. Ces remplacements provoquent des consommations dépendantes des valeurs de

ces bits, qui sont visibles sur les figures avant ou après le chiffrement. Or, les bits de clé écrasé n'ont rien à voir avec ceux utilisés dans les *S-boxes* ciblées. Ceci est étonnant, car nous arrivons quand même à avoir de bons taux de succès. En perspectives, nous essayerons de comprendre les raisons de ce succès inattendu.

Les taux de succès montrent que les bits à l'entrée des *S-boxes* DES ne sont équivalents ni au niveau conceptuel ni au niveau de la mise en œuvre. En effet, chaque *S-box* a deux bits de clé faciles à casser, trois bits moyennement résistants, et un bit difficile à casser.

9.3 Attaques et masquage

face à la puissance des attaques SCA, il est vital de mettre en œuvre des contre-mesures pour sécuriser les circuits sensibles. Deux grandes catégories de contre-mesures ont été proposées dans la littérature :

- conception de circuits électroniques dont la consommation est indépendante des données,
- masquage de la fuite par l'ajout d'éléments aléatoires dans le calcul, qui ne sont pas dans la définition formelle de l'algorithme.

Dans cette section nous proposons l'étude d'une attaque *template* durant laquelle le profilage a été fait à partir d'une campagne de traces non masquée et l'attaque en ligne sur des traces masquées. Le but est de savoir si nous pouvons exploiter les fuites *exotiques* présentées dans la section précédente dans ce genre d'attaque contre les traces masquée par la duplication.

9.3.1 Méthode de duplication

Le but de cette méthode est de construire une implementation d'un algorithme cryptographique qui ne serait pas vulnérable aux attaques SCA.

L'idée de Goubin et Patarin [26] consiste à remplacer chaque variable intermédiaire V , observée durant le chiffrement et dépendante des entrées (ou sorties), par k variables $V_1, \dots, V_k$ qui permettront de retrouver V par la suite. En d'autres termes, pour garantir la sécurité d'un algorithme, ils préconisent de choisir une fonction f telle que $V = f(V_1, \dots, V_k)$ et qui satisfait les conditions suivantes :

Condition 1 : La connaissance de v et d'une valeur de i fixée, $1 \leq i \leq k$, ne permet pas de déduire une information sur l'ensemble des valeurs v_i tel qu'il existe un $(k-1)$ -uplet $(v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_k)$ qui satisfait l'équation : $f(v_1, \dots, v_k) = v$

Condition 2 : La fonction f peut être implémentée sans calculer V (la transformation se fera sur $V_1, \dots, V_k$ au lieu de V).

Pour le cas du DES, ils ont divisé chaque variable intermédiaire v , en deux variables v_1 et v_2 ($k=2$) et ont choisi la fonction $f(v_1, v_2) = v_1 \oplus v_2$. Cette méthode décrite ci-dessus

permet d'immuniser un circuit contre une attaque SCAs *de premier ordre*. Cependant, des attaques d'ordre supérieur (>1) ont été présentées :

- Kocher *et al.* [38] proposent de combiner plusieurs échantillons d'une trace de fuite pour *casser* un circuit protégé par la duplication avec deux variables de masquage. D'un point de vue général, quand une méthode de duplication est utilisée pour protéger une implémentation avec d variables de masquage, on peut toujours faire une attaque de DPA d'ordre d dans le but de reconstruire la dépendance entre la fuite et la variable sensible. Cependant, la difficulté de réaliser une attaque DPA croît en fonction de l'ordre. En effet Chari *et al.* ont montré dans [11] que la complexité (en termes de mesures) d'une telle attaque est de $\Theta(\sigma^d)$, où σ est l'écart-type des fuites de bruit.
- Une nouvelle approche de masquage a été développée par S. Nikova [57] qui se base sur le partage de secret. Cette méthode consiste à prouver la résistance du circuit contre les attaques d'ordre quelconque. Ce nouveau masquage prend en compte la faiblesse des circuits causée par les 'Glitches', et utilise néanmoins plus de valeurs aléatoires durant la procédure, que les autres méthodes de masquages.

9.3.2 Exploitation des fuites exotiques

Les besoins de cette étude sont les suivants :

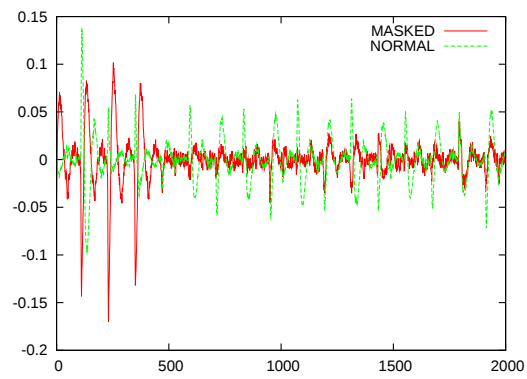
- Une campagne A de 30000 traces non protégées.
- Une campagne B de 30000 traces masquée par une duplication de deux variables.

Le but de cette étude est de chercher les synergies entre ces deux campagnes et de tester des attaques *template* avec le profilage sur A et l'attaque sur B et inversement. Le modèle choisi est la valeur des bits d'entrée dans les *S-boxes* et dans ce cas la PCA donne 1 seul vecteur propre, et deux *template*. Les vecteurs propres associés aux bits 5 et 6 de la première *S-box* et ceux associés aux bits 1 et 2 de la seconde *S-box* sont présentés sur la figure 9.3.2.

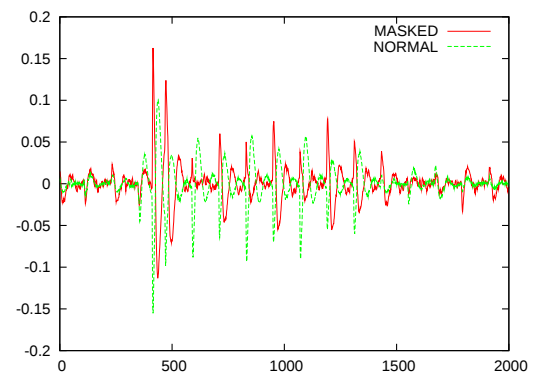
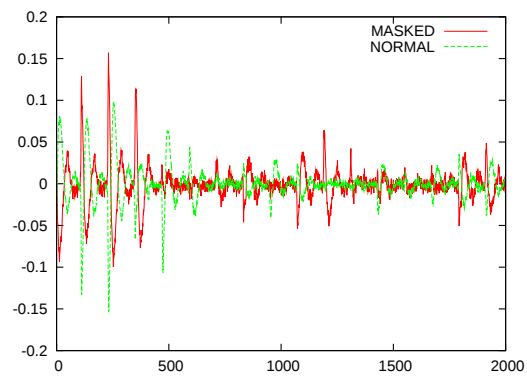
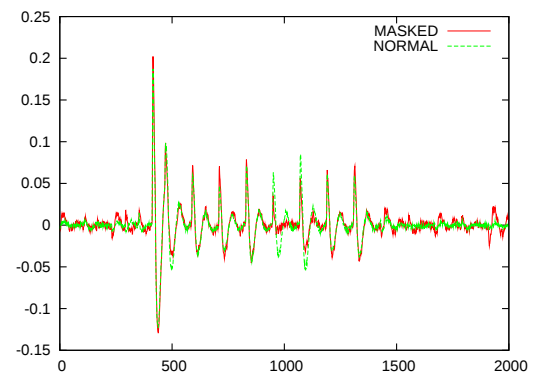
Nous remarquons clairement des similitudes entre les vecteurs propres correspondant aux bits 5 pour la *S-box* 1 et du bit 2 de la *S-box* 2. Ceci veut dire que nous avons la même fuite, qui pourrait simplement être exploitable par une attaque SCA. Or nos métriques n'ont pas été concluantes et le taux de succès a été souvent à 50% en moyenne. Ceci est expliqué par la normalisation des vecteurs propres et aussi par les différences d'amplitude des traces. En effet, les traces masquées ont une amplitude légèrement supérieure à celles des traces non protégées du fait de la présence de la variable de masquage, et ceci est décrit dans la figure 9.3.2.

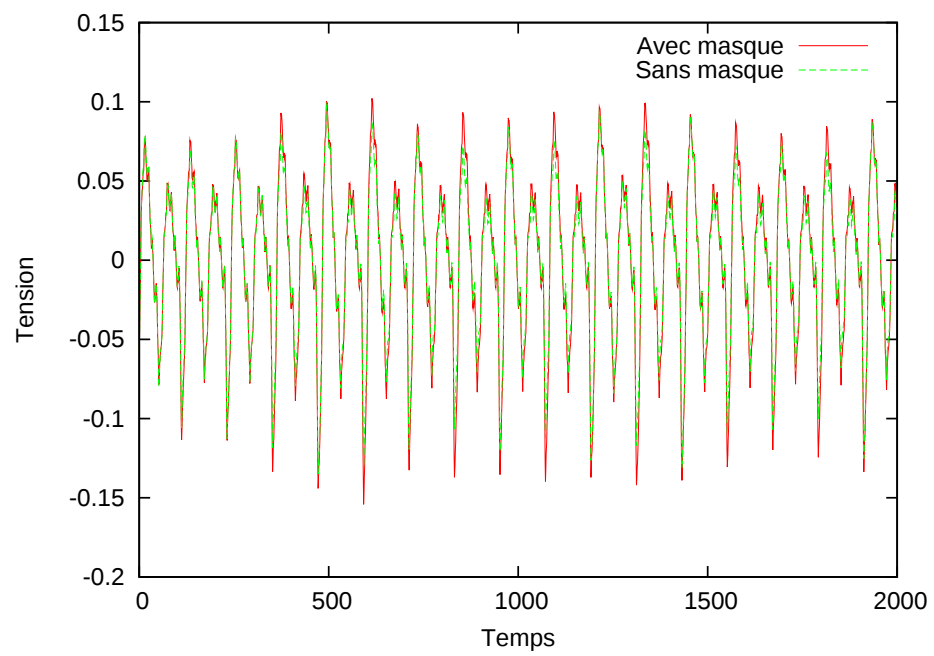
Le problème pourra donc être résolu grâce à la synchronisation des traces comme expliqué dans le chapitre 7. Comme perspectives, nous utiliserons les méthodes proposées pour retrouver la clé dans la campagne masquée.

Fuite avant



Fuite après





Chapitre 10

Conclusions et perspectives

Dans ces travaux de thèse, nous avons fait des investigations techniques sur les attaques *templates*. Nous avons ainsi montré que ces attaques, contrairement aux autres méthodes de cryptanalyse physique, sont très puissantes. Elles peuvent toucher à n'importe quelle partie de l'algorithme du chiffrement, à commencer par le cadencement de la clé. Les boîtes de substitutions ne sont pas l'unique point faible.

Ainsi, ce serait imprudent de faire des efforts de masquage uniquement sur les boîtes de substitutions, ou sur les premiers ou derniers tours, comme il est parfois suggéré dans la littérature.

Il est vrai que ce sont les éléments les plus dissipatifs du chemin de données, et aussi de part leur non-linéarité, elles contribuent à distinguer la bonne clé des autres suppositions.

Les attaques *templates* peuvent attaquer n'importe où, mais avec différents degrés d'intensité. Elles peuvent utiliser uniquement les valeurs/hypothèses de clés pour trouver la meilleure vraisemblance. Contrairement à la DPA qui se concentre, en général, sur le premier ou le dernier tour, celle-ci consiste à la dérivation d'un modèle de consommation pour chaque message et pour une hypothèse clé.

Nous avons montré aussi que le fait d'introduire un modèle de consommation augmenté de plusieurs ordres de grandeur des valeurs propres dans l'analyse par composantes principales. Dans notre analyse, nous mettons l'évaluateur, qui est en charge de quantifier la fuite d'informations, face à l'attaquant, en charge de l'exploitation de la fuite pour extraire le secret. Tout est fait pour qu'ils soient sur un pied d'égalité en ce qui concerne les compétences expérimentales : les deux utilisent la même campagne d'acquisition. Nous avons profité de ces résultats afin de tester une asymétrie dans le couple évaluateur/attaquant, à savoir le rôle de la connaissance initiale. Nous avons illustré concrètement que toute connaissance sur l'architecture du matériel cible ou sur la structure de fuite dans le temps peut favoriser l'un de ces acteurs. Du point de vue de l'attaquant, nous avons proposé deux améliorations par rapport aux attaques traditionnelles.

Tout d’abord, nous avons montré que le soin avec lequel la variable sensible est choisie, influence qualitativement les résultats de l’attaque. Dans l’accélérateur matériel étudié, le choix de la variable sensible appropriée est relié à son architecture RTL. Deuxièmement, nous avons montré que si l’attaquant sait que la fuite est localisée dans quelques échantillons parmi le grand nombre d’échantillons qui composent les traces, il peut deviner précocement (avant que l’attaque n’ait convergé vers un clé candidate) que certains échantillons seront principalement porteur de bruit au lieu d’informations utiles. Grâce à une technique de seuillage innovante, nous avons montré qu’en excluant ces échantillons, l’attaquant peut en effet accélérer l’attaque en terme de métriques. Enfin, nous concluons quant à l’utilité du cadre théorique présenté dans [77] par rapport aux prédictions de sécurité lorsque l’évaluateur et l’attaquant peuvent jouer avec les mêmes degrés de liberté. Toutefois, nous avertissons que l’attaquant, en particulier dans la cryptanalyse, peut toujours venir avec des stratégies innovantes qui pourraient lui donner un avantage imprévu par rapport à l’évaluateur. D’autres directions de recherche consisteront notamment à évaluer la question de l’existence d’un modèle de fuite optimale.

- Quelle est la meilleure attaque/évaluation, qui peut être obtenue à partir d’une analyse impliquant simultanément plusieurs modèles de fuite ?
- Mais dans ce cas, quelle technique est la plus appropriée pour combiner de manière cohérente des modèles de fuite différents ?

En effet, plusieurs techniques qui permettent d’exploiter les informations par canaux auxiliaires à partir de multiples canaux ont déjà été présentées. Par exemple, différents EM sont combinés dans [2]. Aussi, dans [74], la connaissance simultanée de la puissance du courant et des fuites EM sont mis à profit pour réduire le nombre d’interactions avec le dispositif cryptographique attaqué. Toutefois, dans ces deux exemples, le même modèle de fuite est supposé.

Cela nous a conduit à étudier deux exemples de combinaisons dans les attaques SCAs. La première contribution est la démonstration d’un multi-partitionnement constructif de l’attaque.

Nous avons montré qu’un minimum de partitionnement peut améliorer la convergence des taux de succès à cent pour cent ; comme nous bénéficions d’une pré-caractérisation exhaustive, le nombre de *templates* sera augmenté. La durée/longueur d’entraînement est en effet le produit de la phase d’entraînement pour chaque partitionnement.

Nous avons aussi souligné comment la fuite de chaque échantillon peut être combinée, et donner de nouveaux points d’intérêt, meilleurs que ceux des méthodes de réduction des fuites (sosd, sost ou ACP). Cette amélioration provient du fait que chaque échantillon comporte une fuite de nature différente qui peut être exploitée individuellement, ce qui est hors de portée des techniques globales qui consistent à identifier les instants avec de grandes variations. Notre distingueur de combinaison amélioré consiste à multiplier les coefficients de corrélation de Pearson de plusieurs points d’intérêt. Bien que cette attaque fait bénéficier de taux de succès meilleurs que d’autres attaques utilisant différentes techniques de pré-traitement. Nous pensons qu’elles peuvent être encore plus renforcées par une autre méthode d’identification des points d’intérêt avec précision, même lorsque

les observations par canaux auxiliaires contiennent beaucoup de bruit.

Aussi, nous avons vu, malgré les faveurs accordées à un adversaire, comme l'entraînement illimité sur un périphérique clone, il peut être confronté à des difficultés si les traces ne sont ni à l'échelle, ni synchronisées. Nous nous sommes basés sur des expériences réelles, et nous avons fait remarquer que l'amplitude verticale peut changer entre les acquisitions sur le circuit clone et les mesures ciblées du circuit attaqué. Il peut aussi y'avoir une dé-synchronisation dans le temps, qui peut induire des erreurs en particulier dans le choix des points d'intérêt.

Nous avons étudié ce genre de situations, dans le cas où un adversaire utilise l'ACP afin de réduire la dimensionnalité des traces par canaux auxiliaires. Pour notre étude de cas, nous avons réalisé des acquisitions à des dates (années 2006 et 2010) différentes, et nous avons conclu que, malgré tous nos efforts pour maintenir les mêmes conditions expérimentales, l'apparence des traces n'a jamais été sauvegardée de manière idéale.

Dans cette situation, nous avons recommandé que l'adversaire ajoute une phase de traitement entre le profilage et les attaques en ligne. Nous avons montré comment resynchroniser les traces en amplitude et dans le temps pour être en mesure de récupérer la clé. Le ré-alignement dans le temps est très simple ; mais de l'autre coté il est difficile de trouver un coefficient consensuel pour modifier l'amplitude de chaque trace. Ainsi, nous avons introduit la normalisation de traces, et ce pré-traitement semble efficace : il permet, en effet, de garder un taux de succès équivalent comparé à une attaque faite sur la même campagne d'acquisition.

Nous avons aussi exploré des techniques de traitement du signal comme la FFT ou la transformé en ondelettes, et nous avons pointé la grande contribution qu'apporte le prétraitement des traces dans l'élimination du bruit, ou le choix des points d'intérêt.

Ces travaux, pourront orienter vers des contremesures en se basant sur les nouvelles connaissances apportées par cette thèse, et qui seront mises à la disposition d'un adversaire utilisant les attaques *templates*.

En effet, nous avons observé que si le profilage ne se fait pas avec une clé variable, alors il peut y avoir une dépendance avec des données utilisées hors cadre cryptographique. Cela guide la PCA vers des zone "de non-intérêt". En forgeant une fuite dépendant de la donnée sensible, mais pour une autre clé constante, on pourrait effectivement forcer la PCA à s'intéresser à des échantillons "leurre". Dans ce cas l'adversaire pensera à tort qu'il est entrain de manipuler une donnée sensible.

Annexe A

A.1 Les valeurs de la clé et du registre CD au premier tour

Index des templates	Clés de 64-bit : K	Contenu initial du registre CD	CD
0	0101010101010101	00000000000000	0
1	0101800101010101	04000000000000	1
2	0101010101018001	40000000000000	1
3	0101800101018001	44000000000000	2
4	0101010101010110	00000080000000	1
5	0101800101010110	04000080000000	2
6	0101010101018010	40000080000000	2
7	0101800101018010	44000080000000	3
8	0101010140010101	00100000000000	1
9	0101800140010101	04100000000000	2
10	0101010140018001	40100000000000	2
11	0101800140018001	44100000000000	3
12	0101010140010110	00100080000000	2
13	0101800140010110	04100080000000	3
14	0101010140018010	40100080000000	3
15	0101800140018010	44100080000000	4
16	0101010101012001	00004000000000	1
17	0101800101012001	04004000000000	2
18	010101010101a101	40004000000000	2
19	010180010101a101	44004000000000	3
20	0101010101012010	00004080000000	2
21	0101800101012010	04004080000000	3
22	010101010101a110	40004080000000	3
23	010180010101a110	44004080000000	4
24	0101010140012001	00104000000000	2
25	0101800140012001	04104000000000	3
26	010101014001a101	40104000000000	3
27	010180014001a101	44104000000000	4
28	0101010140012010	00104080000000	3
29	0101800140012010	04104080000000	4
30	010101014001a110	40104080000000	4
31	010180014001a110	44104080000000	5
32	0140010101010101	00020000000000	1
33	0140800101010101	04020000000000	2
34	0140010101018001	40020000000000	2
35	0140800101018001	44020000000000	3

Index des templates	Clés de 64-bit : K	Contenu initial du registre CD	CD
36	0140010101010110	00020080000000	2
37	0140800101010110	04020080000000	3
38	0140010101018010	40020080000000	3
39	0140800101018010	44020080000000	4
40	0140010140010101	00120000000000	2
41	0140800140010101	04120000000000	3
42	0140010140018001	40120000000000	3
43	0140800140018001	44120000000000	4
44	0140010140010110	00120080000000	3
45	0140800140010110	04120080000000	4
46	0140010140018010	40120080000000	4
47	0140800140018010	44120080000000	5
48	0140010101012001	00024000000000	2
49	0140800101012001	04024000000000	3
50	014001010101a101	40024000000000	3
51	014080010101a101	44024000000000	4
52	0140010101012010	00024080000000	3
53	0140800101012010	04024080000000	4
54	014001010101a110	40024080000000	4
55	014080010101a110	44024080000000	5
56	0140010140012001	00124000000000	3
57	0140800140012001	04124000000000	4
58	014001014001a101	40124000000000	4
59	014080014001a101	44124000000000	5
60	0140010140012010	00124080000000	4
61	0140800140012010	04124080000000	5
62	014001014001a110	40124080000000	5
63	014080014001a110	44124080000000	6

TAB. A.1 – les valeurs de la clé K du DES et du registre CD au premier tour.

A.2 Resultats détaillés du DPAcontest

A.3 Suite des résultats du chapitre 9

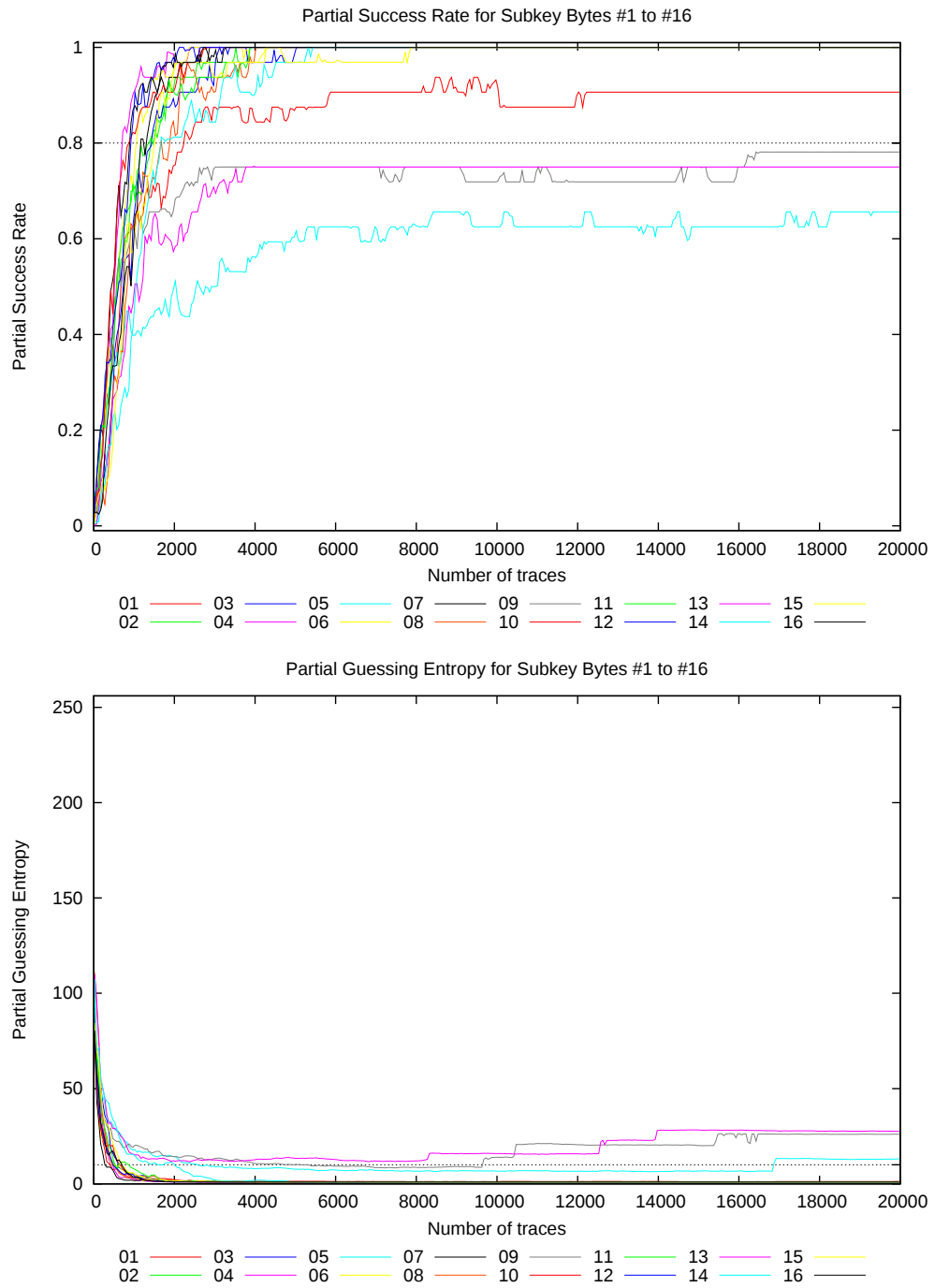


FIG. A.1 – Taux de succès partiels et entropies estimées partielles.

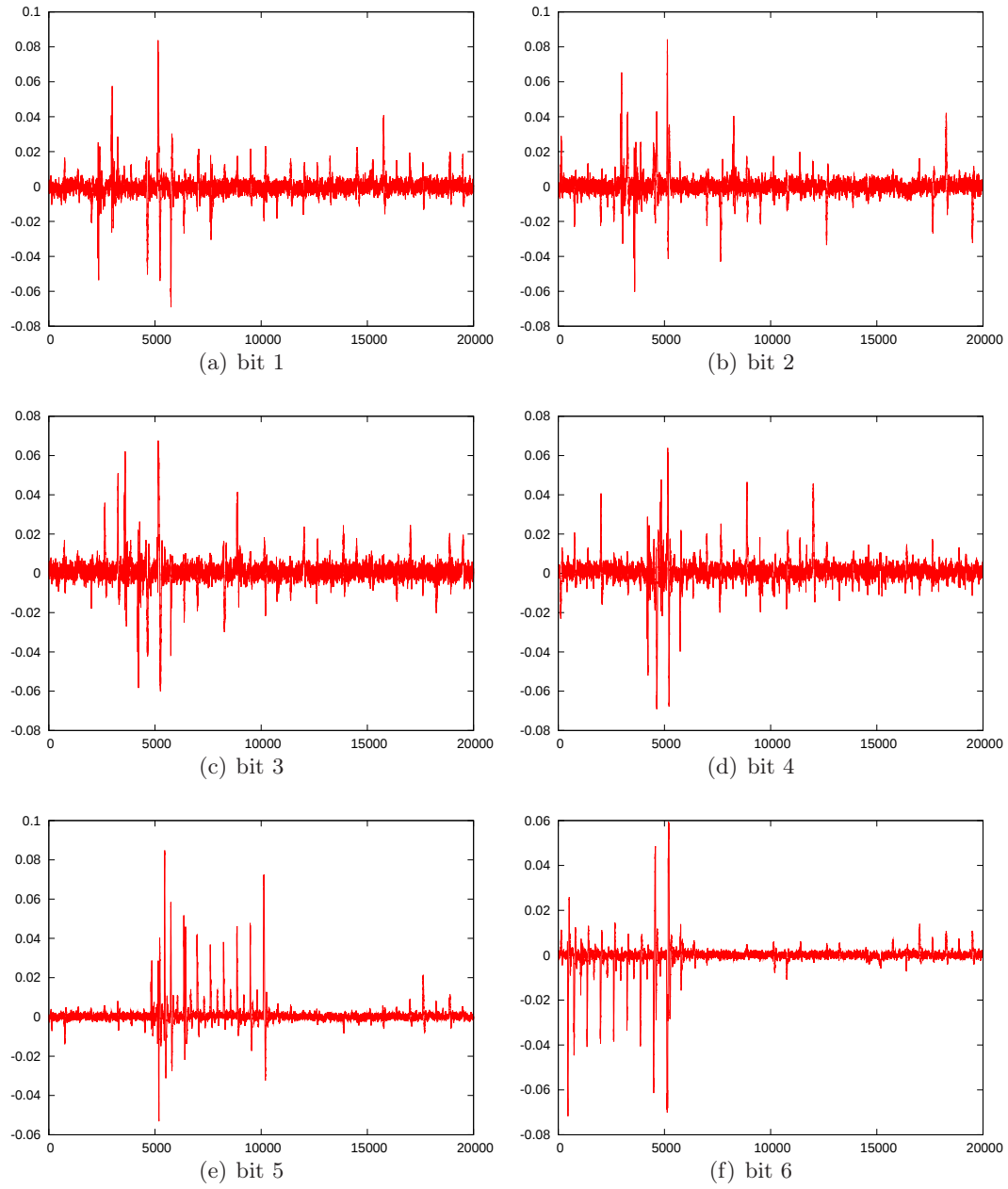


FIG. A.2 – Les vecteurs propres correspondant à chacun des six bits de la S-box 1

Traces	Entropie estimée partielle / Octet																Min	Max	Moy
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16			
10	88.4	85.7	103.0	93.7	102.0	98.9	84.2	126.3	92.8	88.4	87.6	90.2	121.6	112.7	108.8	105.6	84.2	126.3	99.4
20	79.6	85.1	90.2	91.7	98.4	84.8	86.7	118.3	107.8	83.1	73.7	79.7	110.9	116.0	89.3	74.6	73.7	118.3	91.9
30	84.1	90.0	80.0	87.2	99.4	68.8	70.1	105.2	88.8	77.4	76.8	72.7	101.1	100.5	73.2	72.3	68.8	105.2	84.2
40	75.8	96.4	76.5	80.8	93.5	61.2	57.1	93.5	82.1	71.0	70.8	64.2	99.2	90.4	62.6	64.4	57.1	99.2	77.5
50	69.8	87.9	71.8	87.0	86.0	56.4	52.0	94.5	78.8	65.1	70.3	65.3	112.4	78.8	74.2	62.5	52.0	112.4	75.8
100	51.0	62.1	57.1	54.9	54.4	44.0	39.0	84.0	55.5	42.2	59.8	43.5	85.3	76.1	67.3	55.3	39.0	85.3	58.2
200	25.1	42.3	47.9	37.1	45.6	23.1	18.1	43.8	38.0	26.0	31.0	28.3	49.8	52.1	46.7	29.3	18.1	52.1	36.5
300	15.0	24.1	32.3	22.0	38.6	21.6	8.8	24.3	33.0	18.8	26.6	20.2	39.0	42.4	32.4	19.0	8.8	42.4	26.1
400	10.7	12.2	24.1	11.4	32.5	14.9	8.1	14.3	28.0	20.8	19.8	14.6	31.8	39.3	26.9	15.4	8.1	39.3	20.3
500	7.8	10.5	14.2	8.4	31.3	9.8	7.0	9.3	22.8	17.0	11.2	7.6	28.8	33.9	20.3	17.8	7.0	33.9	16.1
1000	2.3	3.5	5.2	2.5	18.7	5.7	1.7	3.0	19.7	4.0	8.4	4.4	14.8	15.6	5.2	3.3	1.7	19.7	7.4
2000	2.2	1.3	1.3	1.0	14.3	1.1	1.2	1.8	14.7	1.2	1.2	1.0	11.9	10.3	1.6	1.0	1.0	14.7	4.2
3000	1.4	1.1	1.1	1.0	9.1	1.0	1.0	1.2	13.6	1.0	1.3	1.0	12.6	2.0	1.2	1.0	1.0	13.6	3.2
4000	1.3	1.0	1.0	1.0	8.1	1.0	1.0	1.0	10.7	1.0	1.0	1.0	12.8	1.8	1.1	1.0	1.0	12.8	2.9
5000	1.3	1.0	1.0	1.0	7.6	1.0	1.0	1.0	10.3	1.0	1.0	1.0	13.4	1.1	1.0	1.0	1.0	13.4	2.8
10000	1.2	1.0	1.0	1.0	6.7	1.0	1.0	1.0	14.0	1.0	1.0	1.0	15.9	1.0	1.0	1.0	1.0	15.9	3.1
15000	1.2	1.0	1.0	1.0	6.8	1.0	1.0	1.0	20.1	1.0	1.0	1.0	28.3	1.0	1.0	1.0	1.0	28.3	4.3
20000	1.2	1.0	1.0	1.0	13.1	1.0	1.0	1.0	26.1	1.0	1.0	1.0	27.6	1.0	1.0	1.0	1.0	27.6	5.0

TAB. A.2 – DPAcontest et résultats partiels de l'entropie estimée

Traces	Taux de succès partiel / Byte																Min	Max	Moy
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16			
10	0.06	0.06	0.03	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.06	0.03	0.00	0.03	0.03	0.00	0.00	0.06	0.02
20	0.03	0.03	0.03	0.06	0.03	0.03	0.03	0.03	0.06	0.00	0.06	0.03	0.00	0.00	0.00	0.03	0.00	0.06	0.03
30	0.03	0.03	0.09	0.06	0.00	0.06	0.00	0.03	0.00	0.00	0.06	0.03	0.00	0.06	0.00	0.03	0.00	0.09	0.03
40	0.00	0.03	0.09	0.06	0.00	0.06	0.06	0.03	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.03	0.00	0.09	0.03
50	0.06	0.03	0.06	0.06	0.00	0.06	0.06	0.09	0.00	0.00	0.06	0.06	0.00	0.00	0.03	0.09	0.00	0.09	0.04
100	0.06	0.06	0.06	0.06	0.03	0.09	0.12	0.03	0.06	0.09	0.06	0.12	0.00	0.00	0.06	0.00	0.00	0.12	0.06
200	0.12	0.19	0.06	0.22	0.19	0.12	0.19	0.09	0.16	0.09	0.19	0.22	0.09	0.03	0.12	0.00	0.00	0.22	0.13
300	0.25	0.34	0.12	0.28	0.22	0.22	0.28	0.06	0.34	0.28	0.22	0.34	0.12	0.09	0.12	0.16	0.06	0.34	0.22
400	0.25	0.38	0.19	0.38	0.31	0.38	0.44	0.19	0.38	0.44	0.31	0.31	0.19	0.19	0.09	0.31	0.09	0.44	0.29
500	0.31	0.44	0.34	0.44	0.34	0.41	0.47	0.28	0.34	0.44	0.38	0.44	0.28	0.25	0.22	0.34	0.22	0.47	0.36
1000	0.62	0.72	0.69	0.91	0.41	0.72	0.91	0.59	0.59	0.81	0.72	0.88	0.50	0.44	0.59	0.62	0.41	0.91	0.67
2000	0.75	0.91	0.88	0.97	0.50	0.94	0.94	0.81	0.69	0.91	0.91	1.00	0.56	0.81	0.94	0.97	0.50	1.00	0.84
3000	0.88	0.97	0.91	1.00	0.50	1.00	1.00	0.91	0.75	1.00	0.94	1.00	0.69	0.84	0.94	1.00	0.50	1.00	0.89
4000	0.84	1.00	0.97	1.00	0.56	0.97	1.00	1.00	0.75	1.00	1.00	1.00	0.75	0.88	0.97	1.00	0.56	1.00	0.92
5000	0.88	1.00	0.97	1.00	0.62	1.00	1.00	1.00	0.75	1.00	1.00	1.00	0.75	0.97	0.97	1.00	0.62	1.00	0.93
10000	0.91	1.00	1.00	1.00	0.62	1.00	1.00	1.00	0.72	1.00	1.00	1.00	0.75	1.00	1.00	1.00	0.62	1.00	0.94
15000	0.91	1.00	1.00	1.00	0.62	1.00	1.00	1.00	0.75	1.00	1.00	1.00	0.75	1.00	1.00	1.00	0.62	1.00	0.94
20000	0.91	1.00	1.00	1.00	0.66	1.00	1.00	1.00	0.78	1.00	1.00	1.00	0.75	1.00	1.00	1.00	0.66	1.00	0.94

TAB. A.3 – DPAcontest et résultats partiels du taux de succès

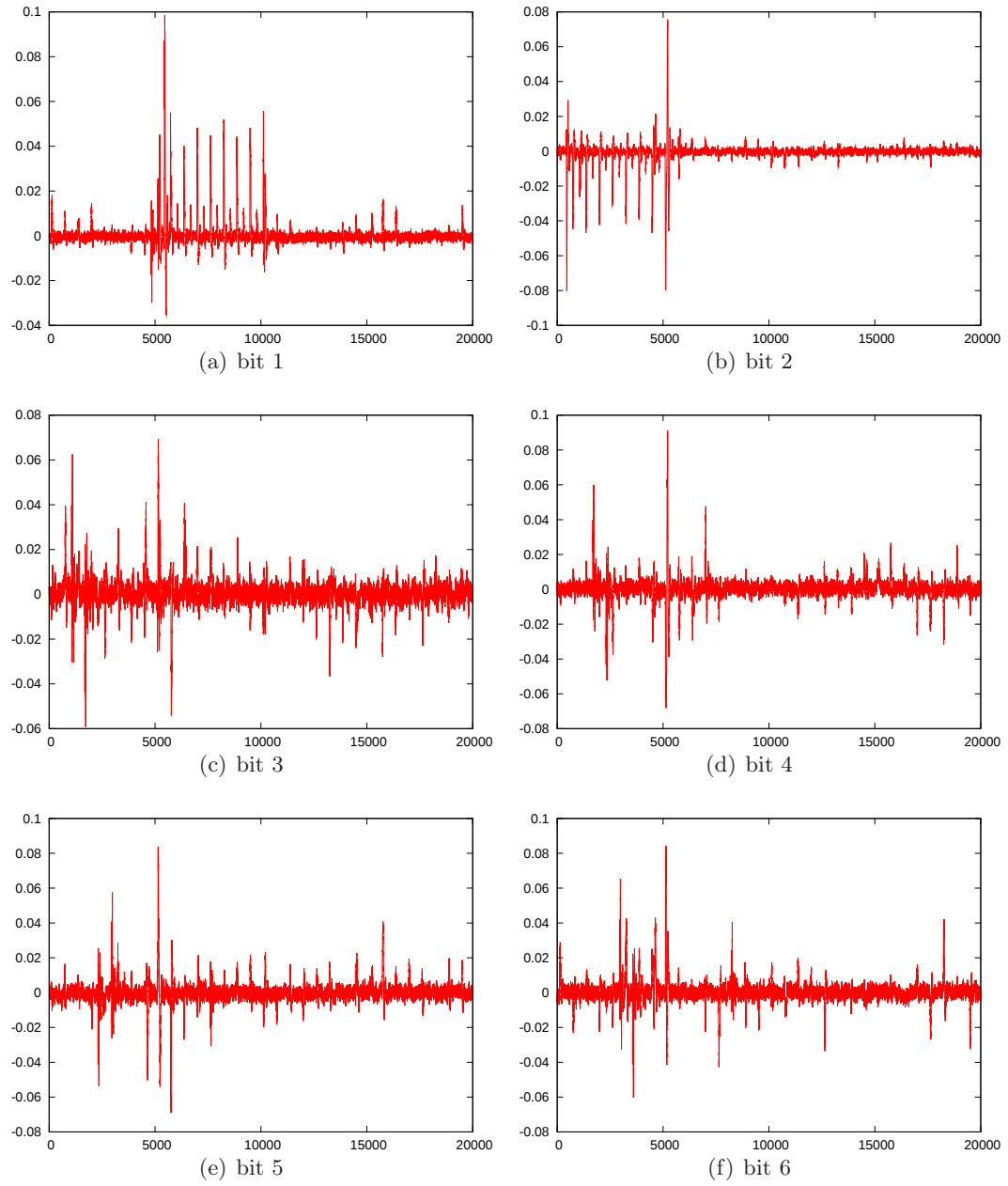


FIG. A.3 – Les vecteurs propres correspondant à chacun des six bits de la S-box 2

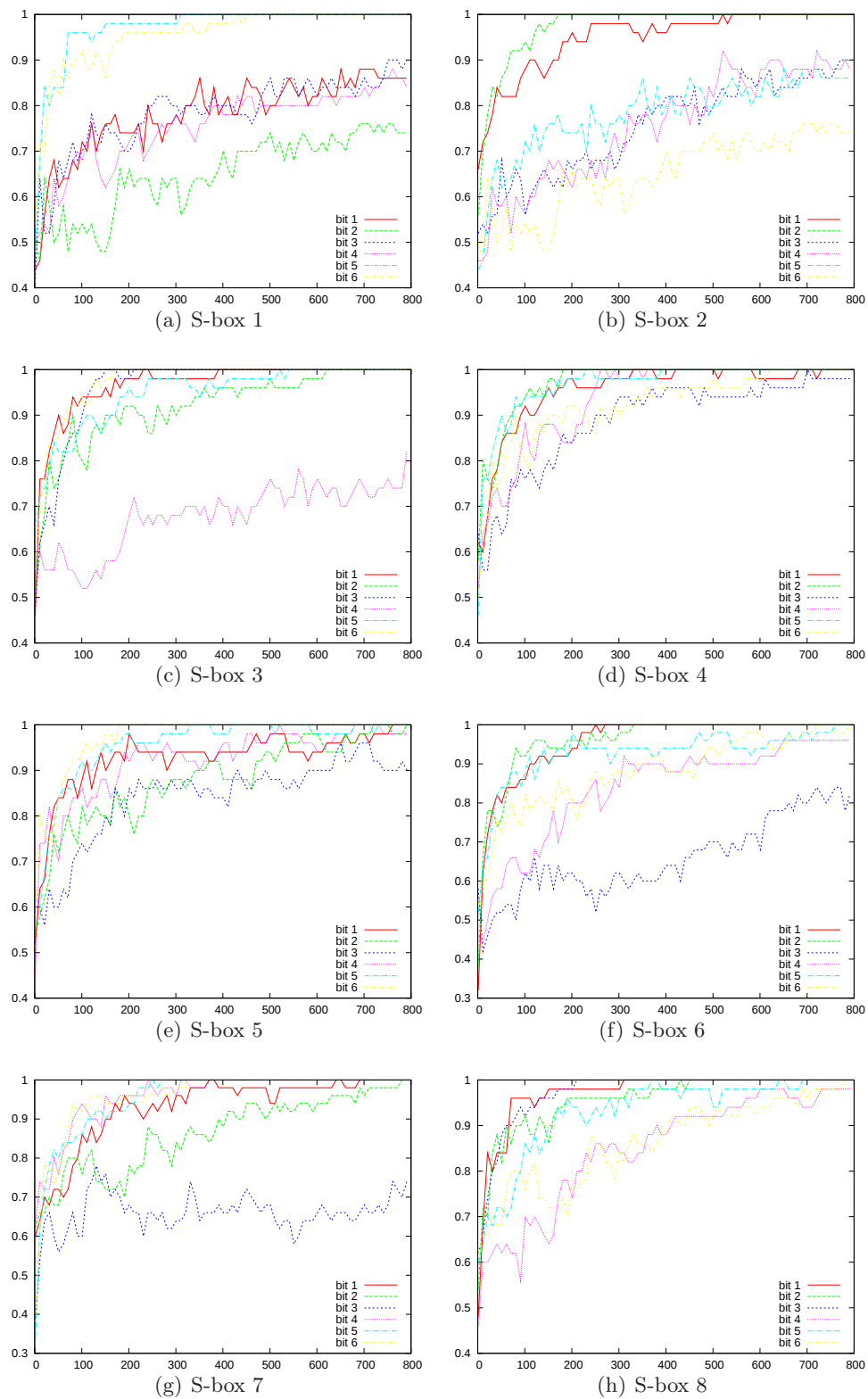


FIG. A.4 – Les taux de succès correspondant à chacun des six bits

Publications

A.4 Articles avec actes

1. M. Abdelaziz Elaabid, Sylvain Guilley, (2010), Practical Improvements of Profiled Side-Channel Attacks on a Hardware Crypto-Accelerator, AFRICACRYPT, Stellenbosch, Afrique du Sud, Volume 6055 , Lecture Notes in Computer Science, pages 243-260, Springer, 2010.
2. M. Abdelaziz Elaabid, Olivier Meynard, Sylvain Guilley, Jean-Luc Danger, (2010), Combined Side-Channel Attacks, WISA, Jeju Island, Korea, Volume 6513, Lecture Notes in Computer Science, pages 175-190, Springer, 2011.

A.5 Article avec acte soumis en attente

1. M. Abdelaziz Elaabid and Sylvain Guilley, Portability of Templates, Journal of Cryptographic Engineering, Springer.

A.6 Communications internationales

1. M. Abdelaziz Elaabid, Sylvain Guilley, Philippe Hoogvorst, (2007), Template Attacks with a Power Model, Cryptology ePrint Archive, Report, 2007/443 (2007), <http://eprint.iacr.org/>
 2. M. Abdelaziz Elaabid and Sylvain Guilley and Jean-Luc Danger, (2011), Exotic Leakage Models, CRYPTARCHI, Bochum, Germany.
 3. Youssef Souissi, M. Abdelaziz Elaabid, Nicolas Debande, Sylvain Guilley et Jean-Luc Danger, (2011), Novel Applications of Wavelet Transforms based Side-Channel Analysis, NIAT, Nara, Japan.
 4. Sylvain Guilley, Olivier Meynard, Maxime Nassar, Guillaume Duc , Philippe Hoogvorst, Houssem Maghrebi, M. Abdelaziz Elaabid, Shivam Bhasin, Youssef Souissi, Nicolas Debande, Laurent Sauvage, Jean-Luc Danger, (2011), Vade Mecum on Side-Channels Attacks and Countermeasures for the Designer and the Evaluator, DTIS, Athens, Greece.
-

5. Florent Flament, Housseem Maghrebi, M. Abdelaziz Elabid, Jean-Luc Danger, Sylvain Guilley and Laurent Sauvage, (2010), About Probability Density Function Estimation for Side Channel Analysis, COSADE, Darmstadt, Germany.

A.7 Communications nationales

1. M. Abdelaziz Elaabid, Claude Carlet, Codes correcteurs d'erreurs et attaques physiques par fautes sur les cartes à puces, (2010), Faute et Erreur, Université Paris 8, Paris, France.
 2. Shivam Bhasin, Taoufik Chouta, Guillaume Duc, Jean-Luc Danger, M. Abdelaziz Elaabid, Florent Flament, Philippe Hoogvorst, Tarik Graba, Sylvain Guilley, Housseem Maghrebi, Olivier Meynard, Maxime Nassar, Renaud Pacalet, Laurent Sauvage, Nidhal Selmane, and Youssef Souissi, (2010), Combined countermeasures against perturbation & observation attacks, PASTIS (PACA Security Trends In embedded Security), Gardanne (École des Mines de Saint-Étienne), France.
 3. Shivam Bhasin, Taoufik Chouta, Guillaume Duc, Jean-Luc Danger, Aziz Elaabid, Florent Flament, Philippe Hoogvorst, Tarik Graba, Sylvain Guilley, Housseem Maghrebi, Olivier Meynard, Maxime Nassar, Renaud Pacalet, Laurent Sauvage, Nidhal Selmane, and Youssef Souissi, (2010), DPA et Dérivées : Attaques et Contremesures, GDR SoC-SiP, Paris, France.
 4. M. Abdelaziz Elaabid, Les attaques templates, (2008), Les juniors de la recherche, Université Paris 8, Paris, France
 5. Sylvain Guilley, Jean-Luc Danger, Moulay Abdelaziz El Aabid, Renaud Pacalet, and Philippe Hoogvorst, Symbolic Simulation for Security, July 2007. PASR USEIT'07, CNES, Toulouse, France.
-

Bibliographie

- [1] Agilent Technologies. Agilent InfiniiVision 5000/6000/7000 Series Oscilloscopes – User’s Guide. <http://cp.literature.agilent.com/litweb/pdf/54695-97022.pdf>.
 - [2] Dakshi Agrawal, Josyula R. Rao, and Pankaj Rohatgi. Multi-channel Attacks. In *CHES*, volume 2779 of *LNCS*, pages 2–16. Springer, September 8-10 2003. Cologne, Germany.
 - [3] Frédéric Amiel, Karine Villegas, Benoît Feix, and Louis Marcel. Passive and Active Combined Attacks : Combining Fault Attacks and Side Channel Analysis. In *FDTC*, pages 92–102. IEEE Computer Society, 10 September 2007. Vienna, Austria.
 - [4] Cédric Archambeau, Éric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template Attacks in Principal Subspaces. In *CHES*, volume 4249 of *LNCS*, pages 1–14. Springer, October 10-13 2006. Yokohama, Japan.
 - [5] Sébastien Aumônier. Generalized Correlation Power Analysis. In *ECRYPT Workshop “Tools for Cryptanalysis”*, 24-25 September 2007. Krakow, Poland, PDF.
 - [6] A. Hakan Baba and Subhasish Mitra. Testing for Transistor Aging. In *VTS*, pages 215–220. IEEE Computer Society, 2009.
 - [7] BENNETT(C.H.). Logical reversibility of computation. *IBM J. Res. Develop.*, pages 525–532, Nov 1973.
 - [8] Régis Bevan and Erik Knudsen. Ways to Enhance Differential Power Analysis. In *ICISC*, volume 2587 of *Lecture Notes in Computer Science*, pages 327–342. Springer, November 28-29 2002. Seoul, Korea.
 - [9] Éric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In *CHES*, volume 3156 of *LNCS*, pages 16–29. Springer, August 11–13 2004. Cambridge, MA, USA.
 - [10] Claude Carlet. On Highly Nonlinear S-Boxes and Their Inability to Thwart DPA Attacks. In *INDOCRYPT*, volume 3797 of *LNCS*, pages 49–62. Springer, december 2005. Bangalore, India. (PDF on SpringerLink ; Complete version on IACR ePrint).
 - [11] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *CRYPTO*, volume 1666 of *LNCS*. Springer, August 15-19 1999. Santa Barbara, CA, USA.
 - [12] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template Attacks. In *CHES*, volume 2523 of *LNCS*, pages 13–28. Springer, August 2002. San Francisco Bay (Redwood City), USA.
-

-
- [13] Christophe Clavier, Jean-Sébastien Coron, and Nora Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In *CHES*, LNCS, pages 252–263, London, UK, August 2000. Springer-Verlag.
 - [14] Christophe Clavier, Benoît Feix, Georges Gagnerot, and Mylène Roussellet. Passive and Active Combined Attacks on AES. In *FDTC*, pages 10–18. IEEE Computer Society, 21 August 2010. Santa Barbara, CA, USA. DOI : 10.1109/FDTC.2010.17.
 - [15] “Circuits Multi-Projets” (alias CMP, <cmp@imag.fr>) Annual Report 2005. (Online PDF versions : web / local).
 - [16] Common Criteria consortium. Application of Attack Potential to Smartcards v2-5, April 2008. <http://www.commoncriteriaportal.org/files/supdocs/CCDB-2008-04-001.pdf>.
 - [17] Jean-Sébastien Coron and Ilya Kizhvatov. Analysis and Improvement of the Random Delay Countermeasure of CHES 2009. In *CHES*, volume 6225 of *LNCS*, pages 95–109. Springer, August 17-20 2010. Santa Barbara, CA, USA.
 - [18] Jean-Sébastien Coron, Paul C. Kocher, and David Naccache. Statistics and Secret Leakage. In *Financial Cryptography*, volume 1962 of *Lecture Notes in Computer Science*, pages 157–173. Springer, February 20-24 2000. Anguilla, British West Indies.
 - [19] Moulay Abdelaziz Elaabid and Sylvain Guilley. Practical Improvements of Profiled Side-Channel Attacks on a Hardware Crypto-Accelerator. In *AFRICACRYPT*, volume 6055 of *LNCS*, pages 243–260. Springer, May 03-06 2010. Stellenbosch, South Africa. DOI : 10.1007/978-3-642-12678-9_15.
 - [20] Federal Information Processing Standards (FIPS) Publication. *Announcing the Advanced Encryption Standard (AES)*. Number 197. November 2001.
 - [21] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2) :216–231, February 2005.
 - [22] Benedikt Gierlichs, Lejla Batina, Bart Preneel, and Ingrid Verbauwhede. Revisiting Higher-Order DPA Attacks : Multivariate Mutual Information Analysis. In *CT-RSA*, volume 5985 of *LNCS*, pages 221–234. Springer, March 1-5 2010. San Francisco, CA, USA.
 - [23] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In *CHES, 10th International Workshop*, volume 5154 of *LNCS*, pages 426–442. Springer, August 10-13 2008. Washington, D.C., USA.
 - [24] Benedikt Gierlichs, Elke De Mulder, Bart Preneel, and Ingrid Verbauwhede. Empirical comparison of side channel analysis distinguishers on DES in hardware. In IEEE, editor, *ECCTD. European Conference on Circuit Theory and Design*, pages 391–394, August 23-27 2009. Antalya, Turkey.
 - [25] Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. Stochastic Methods. In *CHES*, volume 4249 of *LNCS*, pages 15–29. Springer, October 10-13 2006. Yokohama, Japan.
 - [26] Louis Goubin and Jacques Patarin. DES and Differential Power Analysis. In *CHES*, LNCS, pages 158–172. Springer, Aug 1999. Worcester, MA, USA.
 - [27] Sylvain Guilley. *Contre-mesures Géométriques aux Attaques Exploitant les Canaux Cachés*. PhD thesis, École Nationale Supérieure des Télécommunications, January 2007. ENST 2007 E 003, <http://pastel.paristech.org/2562/01/phd.pdf>.
 - [28] Sylvain Guilley, Philippe Hoogvorst, and Renaud Pacalet. Differential Power Analysis Model and some Results. In Kluwer, editor, *Proceedings of WCC/CARDIS*, pages 127–142, Aug 2004. Toulouse, France. DOI : 10.1007/1-4020-8147-2_9.
-

-
- [29] Sylvain Guilley, Philippe Hoogvorst, and Renaud Pacalet. A Fast Pipelined Multi-Mode DES Architecture Operating in IP Representation. *Integration, The VLSI Journal*, 40(4) :479–489, July 2007. DOI : 10.1016/j.vlsi.2006.06.004.
 - [30] Sylvain Guilley, Philippe Hoogvorst, Renaud Pacalet, and Johannes Schmidt. Improving Side-Channel Attacks by Exploiting Substitution Boxes Properties. In Presse Universitaire de Rouen et du Havre, editor, *BFCA*, pages 1–25, 2007. May 02–04, Paris, France, <http://www.liafa.jussieu.fr/bfca/books/BFCA07.pdf>.
 - [31] Sylvain Guilley, Karim Khalfallah, Victor Lomne, and Jean-Luc Danger. Formal Framework for the Evaluation of Waveform Resynchronization Algorithms. In *WISTP*, volume 6633 of *LNCS*, pages 100–115. Springer, June 1-3 2011. Heraklion, Greece.
 - [32] Neil Hanley, Robert McEvoy, Michael Tunstall, Claire Whelan, Colin Murphy, and William P. Marnane. Correlation Power Analysis of Large Word Sizes. In *ISSC (Irish Signals and System Conference)*, pages 145–150. IET, 13-14 Sept 2007. Edinburgh, Scotland, UK.
 - [33] Harry L. Van Trees. *Detection Estimation and Modulation Theory*. 1968. Republished in paperback by Wiley in 2001, 720 pages, ISBN : 0471095176.
 - [34] Naofumi Homma, Sei Nagashima, Yuichi Imai, Takafumi Aoki, and Akashi Satoh. High-Resolution Side-Channel Attack Using Phase-Based Waveform Matching. In *CHES*, volume 4249 of *LNCS*, pages 187–200. Springer, October 10-13 2006. Yokohama, Japan.
 - [35] Naofumi Homma, Sei Nagashima, Takeshi Sugawara, Takafumi Aoki, and Akashi Satoh. A High-Resolution Phase-Based Waveform Matching and Its Application to Side-Channel Attacks. *IEICE Transactions*, 91-A(1) :193–202, 2008.
 - [36] Marc Joye, Pascal Paillier, and Berry Schoenmakers. On Second-Order Differential Power Analysis. In *CHES*, volume 3659 of *LNCS*, pages 293–308. Springer, August 29 – September 1st 2005. Edinburgh, UK.
 - [37] Jens-Peter Kaps and Rajesh Velegalati. DPA Resistant AES on FPGA Using Partial DDL. In *FCCM : 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, pages 273–280. IEEE Computer Society, May 02–May 04 2010. Charlotte, North Carolina, USA. DOI : 10.1109/FCCM.2010.49.
 - [38] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *Proceedings of CRYPTO’99*, volume 1666 of *LNCS*, pages 388–397. Springer-Verlag, 1999.
 - [39] Yuichi Komano, Hideo Shimizu, and Shinichi Kawamura. Built-in determined sub-key correlation power analysis. Cryptology ePrint Archive, Report 2009/161, 2009. <http://eprint.iacr.org/2009/161>.
 - [40] Thanh-Ha Le. *Analyses et Mesures Avancées du Rayonnement Électromagnétique d’un Circuit Intégré*. PhD thesis, “Institut National Polytechnique” (INP), September 5 2007. Grenoble, France.
 - [41] Thanh-Ha Le, Cécile Canovas, and Jessy Clédière. An overview of side channel analysis attacks. In *ASIACCS*, pages 33–43. ASIAN ACM Symposium on Information, Computer and Communications Security, 2008. DOI : 10.1145/1368310.1368319. Tōkyō, Japan.
 - [42] Thanh-Ha Le, Jessy Clédière, Cécile Canovas, Bruno Robisson, Christine Servièrre, and Jean-Louis Lacoume. A Proposition for Correlation Power Analysis Enhancement. In *CHES*, volume 4249 of *LNCS*, pages 174–186. Springer, 2006. Yokohama, Japan.
-

-
- [43] Thanh-Hà Le, Jessy Clédière, Christine Servièrè, and Jean-Louis Lacoume. How can Signal Processing Benefit Side Channel Attacks? In *Proceedings of IEEE Workshop on Signal Processing Applications for Public Security and Forensics (SAFE)*, pages 1–7, April 11-13 2007. Washington D.C., USA.
 - [44] Thanh-Hà Le, Quoc-Thinh Nguyen-Vuong, Cécile Canovas, and Jessy Clédière. Novel Approaches for Improving the Power Consumption Models in Correlation Analysis. Cryptology ePrint Archive, Report 2007/306, 2007.
 - [45] S. Lin and X. Huang. *Advanced Research on Computer Education, Simulation and Modeling : International Conference, CESM 2011, Wuhan, China, June 18-19, 2011. Proceedings*. Number ptie. 2 in Communications in Computer and Information Science Series. Springer, 2011.
 - [46] Louis Goubin and Jacques Patarin. DES and Differential Power Analysis - The "Duplication" Method, 1999.
 - [47] François Macé, François-Xavier Standaert, and Jean-Jacques Quisquater. Information theoretic evaluation of side-channel resistant logic styles. In *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 427–442. Springer, September 2007. Vienna, Austria.
 - [48] Stefan Mangard. Hardware Countermeasures against DPA – A Statistical Analysis of Their Effectiveness. In *CT-RSA*, volume 2964 of *LNCS*, pages 222–235. Springer, 2004. San Francisco, CA, USA.
 - [49] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks : Revealing the Secrets of Smart Cards*. Springer, December 2006. ISBN 0-387-30857-1, <http://www.dpabook.org/>.
 - [50] Jaekel (M.-T.) Matherat (P.). "Dissipation logique des implémentations d'automates - dissipation du calcul,. *Technique et Science Informatiques*, pages 1079–1104, 1996.
 - [51] Thomas S. Messerges. Using second-Order Power Analysis to Attack DPA resistant Software. In *CHES*, volume 1965 of *LNCS*, pages 71–77. Springer, August 17-18 2000. Worcester, MA, USA.
 - [52] Thomas S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In Christof Paar and Çetin Kaya Koç, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000*, volume 1965 of *LNCS*, pages 238–251. Springer, 2000.
 - [53] Thomas S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In *CHES*, volume 1965 of *LNCS*, pages 238–251. Springer-Verlag, August 17-18 2000. Worcester, MA, USA.
 - [54] Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *USENIX — Smartcard'99*, pages 151–162, May 10–11 1999. Chicago, Illinois, USA (Online PDF).
 - [55] Silvio Micali and Leonid Reyzin. Physically observable cryptography. In *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 278–296. Springer, February 19-21 2004. Cambridge, MA, USA.
 - [56] Sei Nagashima, Naofumi Homma, Yuichi Imai, Takafumi Aoki, and Akashi Satoh. DPA Using Phase-Based Waveform Matching against Random-Delay Countermeasure. In *ISCAS*, pages 1807–1810. IEEE Computer Society, May 27-20 2007. New Orleans, Louisiana, USA. DOI : 10.1109/ISCAS.2007.378024.
 - [57] Svetla Nikova, Christian Rechberger, and Vincent Rijmen. Threshold Implementations Against Side-Channel Attacks and Glitches. In *ICICS*, volume 4307 of *LNCS*, pages 529–545. Springer, December 4-7 2006. Raleigh, NC, USA.
-

-
- [58] Roman Novak. Side-Channel Attack on Substitution Blocks. In *ACNS*, volume 2846 of *LNCS*, pages 307–318. Springer, October 2003. Kunming, China.
 - [59] Roman Novak. Sign-Based Differential Power Analysis. In *WISA*, volume 2908 of *LNCS*, pages 203–216. Springer, 2003. Jeju Island, Korea.
 - [60] Hervé Pelletier and Xavier Charvet. Improving the DPA attack using Wavelet transform, September 26-29 2005. Honolulu, Hawaii, USA; NIST's Physical Security Testing Workshop. Website : <http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-3/physec/papers/physecpaper14.pdf>.
 - [61] Emmanuel Prouff. DPA Attacks and S-Boxes. In *FSE*, volume 3557 of *LNCS*, pages 424–441. Springer-Verlag, february 2005. Paris, France.
 - [62] Emmanuel Prouff and Matthieu Rivain. Theoretical and Practical Aspects of Mutual Information Based Side Channel Analysis. In Springer, editor, *ACNS*, volume 5536 of *LNCS*, pages 499–518, June 2-5 2009. Paris-Rocquencourt, France.
 - [63] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Transactions on Computers*, 58(6) :799–811, 2009.
 - [64] Denis Réal, Frédéric Valette, and M'hamed Drissi. Enhancing correlation electromagnetic attack using planar near-field cartography. In *DATE*, pages 628–633. IEEE, April 20-24 2009. Nice, France.
 - [65] Christian Rechberger and Elisabeth Oswald. Practical Template Attacks. In *WISA*, volume 3325 of *LNCS*, pages 443–457. Springer, August 23-25 2004. Jeju Island, Korea.
 - [66] Kyung Hyune Rhee and DaeHun Nyang, editors. *Information Security and Cryptology - ICISC 2010 - 13th International Conference, Seoul, Korea, December 1-3, 2010, Revised Selected Papers*, volume 6829 of *Lecture Notes in Computer Science*. Springer, 2011.
 - [67] Matthieu Rivain. On the Exact Success Rate of Side Channel Analysis in the Gaussian Model. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *SAC*, volume 5381 of *LNCS*, pages 165–183. Springer, 2008.
 - [68] Matthieu Rivain, Emmanuel Prouff, and Julien Doget. Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers. In *CHES*, volume 5747 of *LNCS*, pages 171–188. Springer, September 6-9 2009. Lausanne, Switzerland.
 - [69] Akashi Satoh. Side-channel Attack Standard Evaluation Board, SASEBO. Project of the AIST – RCIS (Research Center for Information Security), <http://www.rcis.aist.go.jp/special/SASEBO/>.
 - [70] Oliver Schimmel, Paul Duplys, Eberhard Boehl, Jan Hayek, and Wolfgang Rosenstiel. Correlation power analysis in frequency domain. In *COSADE*, pages 1–3, February 4-5 2010. Darmstadt, Germany. http://cosade2010.cased.de/files/proceedings/cosade2010_paper_1.pdf.
 - [71] Werner Schindler, Kerstin Lemke, and Christof Paar. A Stochastic Model for Differential Side Channel Cryptanalysis. In *CHES*, volume 3659 of *LNCS*, pages 30–46. Springer, Sept 2005. Edinburgh, Scotland, UK.
 - [72] Claude E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.
 - [73] François-Xavier Standaert. A Didactic Classification of Some Illustrative Leakage Functions. In *WISSEC, 1st Benelux Workshop on Information and System Security*, page 16, November 8-9 2006. Antwerpen, Belgium.
-

- [74] François-Xavier Standaert and Cédric Archambeau. Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer, August 10–13 2008. Washington, D.C., USA.
 - [75] François-Xavier Standaert, Benedikt Gierlichs, and Ingrid Verbauwhede. Partition vs. Comparison Side-Channel Distinguishers : An Empirical Evaluation of Statistical Tests for Univariate Side-Channel Attacks against Two Unprotected CMOS Devices. In *ICISC*, volume 5461 of *LNCS*, pages 253–267. Springer, December 3-5 2008. Seoul, Korea.
 - [76] François-Xavier Standaert, François Koeune, and Werner Schindler. How to Compare Profiled Side-Channel Attacks? In Springer, editor, *ACNS*, volume 5536 of *LNCS*, pages 485–498, June 2-5 2009. Paris-Rocquencourt, France.
 - [77] François-Xavier Standaert, Tal Malkin, and Moti Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *EUROCRYPT*, volume 5479 of *LNCS*, pages 443–461. Springer, April 26-30 2009. Cologne, Germany.
 - [78] François-Xavier Standaert, Tal G. Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. Cryptology ePrint Archive, Report 2006/139, 2006. <http://eprint.iacr.org/>.
 - [79] TELECOM ParisTech SEN research group. DPA Contest (1st edition), 2008–2009. <http://www.DPAcontest.org/>.
 - [80] Michael Tunstall and Olivier Benoît. Efficient Use of Random Delays in Embedded Software. In *WISTP*, volume 4462 of *Lecture Notes in Computer Science*, pages 27–38. Springer, May 9-11 2007. Heraklion, Crete, Greece.
 - [81] Rajesh Velegalati and Jens-Peter Kaps. DPA resistance for light-weight implementations of cryptographic algorithms on FPGAs. In *FPL*, pages 385–390. IEEE, August 31 - September 2 2009. Prague, Czech Republic. DOI : 10.1109/FPL.2009.5272260.
 - [82] Nicolas Veyrat-Charvillon and François-Xavier Standaert. Mutual Information Analysis : How, When and Why? In *CHES*, volume 5747 of *LNCS*, pages 429–443. Springer, September 6-9 2009. Lausanne, Switzerland.
-