

Optimization methods for side channel attacks

Youssef Souissi

▶ **To cite this version:**

Youssef Souissi. Optimization methods for side channel attacks. Embedded Systems. Télécom ParisTech, 2011. English. NNT : . pastel-00681665

HAL Id: pastel-00681665
https://pastel.hal.science/pastel-00681665

Submitted on 22 Mar 2012

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.

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2012-ENST-00681665

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Electronique et Communication »

présentée et soutenue publiquement par

Youssef SOUISSI

06 décembre 2011

Méthodes optimisant l'analyse des cryptoprocesseurs sur les canaux cachés

Directeur de thèse : **Jean-Luc DANGER**
Co-encadrement de la thèse : **Sylvain GUILLEY**

Jury
M. David NACCACHE, Professeur, ENS , Ecole Normale Supérieure
Mme Assia TRIA, Professeur, CEA-Leti, Commissariat à l'Energie Atomique
M. Guy GOGNIAT, Professeur, Lab-STICC, Université de Bretagne Sud
M. Pascal CHOUR, Expert en sécurité, ANSSI, Agence Nationale de la Sécurité des SI
M. Benoit FEIX, Expert en sécurité, Inside Secure
Mme Thanh-ha LE, Expert en sécurité, Morpho, SAFRAN

TELECOM ParisTech
école de l'Institut Télécom - membre de ParisTech

T H È S E

Abstract

The security of modern embedded systems has been the subject of intensive research in engineering areas. Recent threats called Side-Channel Analysis (SCA) have attracted much attention in embedded security areas. These analyses are serious concerns as they are able to retrieve the secret information from cryptographic implementations without tampering with the system, by exploiting unintentional physical leakage, such as the timing information, power consumption or radiated magnetic field. SCAs are passive attacks, in that the device under attack is not aware of its leaks being recorded. Therefore, the need of securing and evaluating the robustness of embedded systems against SCAs becomes obvious. Basically, four aspects of security evaluation analysis should be taken into consideration: the acquisition of Side-channel traces, the preprocessing of traces acquired, the detection and extraction of cryptographic patterns from the preprocessed traces, and finally the recovery of sensitive information, referred to as the secret key. This thesis investigates new techniques in the analysis of systems for Side-channel attacks. It considers how evaluation targets are characterized, how their behaviour may be simulated – in order to hone targets for empirical analysis and then how data can be collected and analysed. The overall goal is the establishment of a methodological basis for this work. The first part of this thesis focuses on physical cryptanalysis. Several solutions and generic Side-channel attacks are addressed. The second part of this thesis is devoted to the pre-processing of the Side-channel leaked information. We propose new techniques and efficient pre-processing algorithms to get rid off the issues related principally to the noise and de-synchronisation problems. In the last part of this thesis, we establish a methodological framework, which aims at best organizing the task of the evaluator. We also highlight common pitfalls made by evaluators and solutions to avoid them.

Remerciements

Je commence la présentation de ce travail par le remerciement de tous ceux qui ont aidé et facilité à sa réalisation . Je m'adresse à mes deux directeurs de thèse Monsieur Jean-Luc Danger et Monsieur Sylavin Guilley pour leur signaler combien je suis sensible à leurs qualités.

- Monsieur Jen-Luc Danger, sa disponibilité, ses qualités humaines et sa bonté resteront pour moi un exemple à suivre. Que ce travail soit pour lui le témoignage de mon respect. Il m'a dirigé efficacement tout au long de ce travail, ainsi que dans bien d'autres circonstances.
- Monsieur Sylvain Guilley, sa contribution, sa disponibilité, son sérieux et ses connaissances ont permis à mon travail d'avoir plus de valeur. Qu'il reçoive ce témoignage en souvenir de mon amitié et des années passées dans cette superbe école.

J'ai eu la joie et la fierté de bénéficier de la participation de Monsieur David Naccache. Cela a été un grand honneur qu'il ait accepté de juger mon travail, j'ai admiré sa gentillesse et sa chaleur. Que cette réalisation soit le témoin de ma profonde reconnaissance.

La présence de Madame Assia Tria m'a été précieuse, elle m'a impressionné par la précision de ses remarques et de son accueil chaleureux. Je lui exprime mes remerciements les plus sincères en témoignage de ma considération.

Mes remerciements vont aussi à tous les membres du jury à savoir Monsieur Guy Gogniat, Monsieur Benoît Feix, Monsieur Pascal Chour et Madame Thanh Ha Le, pour m'avoir fait l'honneur de participer à la soutenance de thèse et de juger mon travail.

Je tiens également à remercier Messieurs Florent Flament, Laurent Sauvage, Guillaume Duc, Philippe Nguyen, Philippe Hoogvorst, Yves Mathieu, Tarik Graba, Naofumi Homma, Philippe Materat, Hervé Chabanne et Hassan Triqui qui m'ont été d'un soutien continu. J'ai été très touché par leur

disponibilité et leurs conseils précieux, pour leur dire que je vous serais sincèrement reconnaissant.

Au cours de toutes ces années j'ai pu évoluer avec des amis exceptionnels, à savoir Sami Mekki, Walid Trabelsi, Maxime Nassar, Shivam Bhasin, Houssem Maghrebi, Mehrez Selmi, Nicolas Debande, Jeremie Brunel, Sompasong Savady, Nidhal Selmane, Sebastien Thomas, Taufik Chouta, Olivier Meynard, Aziz Elaabid, Zouha Cherif, Molka Ben Romdhane, Loic Thierry, Khalil Kchaou, Sumanta Chaudhuri, Farouk Khalil, Maya Badr, Ali Osmane, Mekki Amiri, Chadi Jabbour, Mohamed Ghrairi, Zahir laarabi, Amel Grira, Guillaume Barbu, et tous les membres de Secure-IC Sebastien Brais, Thibau Porteboeuf, Robert Nguyen ...

Cela était une chance d'avoir une telle bonne compagnie . Ils ont toujours été présents pour partager ensemble le meilleur et le pire.

Durant ces années passées dans notre chère école, j'ai bénéficié d'un corps administratif toujours présent pour m'aider et me soutenir très chaleureusement. Je les ai côtoyé quotidiennement avec une joie qui m'a permise d'avancer sérieusement. Ceci est l'occasion de leur présenter ma sincère et amicale considération. Je cite Messieurs Karim Ben Kalaia, Dominique Roux, Bruno Thedrez, Pascal Gelly (Ecole des Mines) et Frederic Pauget; et Mesdames Chantal Cadiat, Zouina Sahnoun, Nazha Essakkaki, Florence Besnard, Nouria Chaouy (Ecole des Mines), Fabienne Lassausaie et Danielle Childz .

Enfin, ces remerciements ne seraient complets sans mentionner les personnes que j'aime le plus au monde, mon père Jomaa, je l'admire pour son combat contre la maladie chaque jour, que Dieu le garde le plus longtemps possible; ma chère mère Faiza, je l'admire pour son soutien continu et son courage; mon adorable femme Sayda et ma bien aimable petite soeur Mouna. Mes remerciements vont également à mon beau frère Adnene et aux familles Souissi, Triki, Ben Chaabane et Zouaoui.

Contents

Abstract i

Contents iv

List of Figures 2

List of Tables 6

Résumé de la thèse en Français. 7

0.1 Chapitre 1: Introduction et Plan de la thèse . . . 7
0.2 Chapitre 2: Attaques par canaux cachés (SCA) . . . 12
0.2.1 Contribution 1: Optimalité de la CPA . . . 12
0.2.2 Contribution 2: Combinaison des distingueurs . . . 13
0.2.3 Contribution 3: Le Correceteur de rang (RC) . . . 14
0.2.4 Contribution 4: L'attaque multivarivariée FPCA . . . 16
0.2.5 Contribution 5: Attaque en ondelettes . . . 18
0.3 Chapitre 3: Pré-traitement des traces SCA . . . 21
0.3.1 Techniques de filtrage de traces . . . 21
0.3.1.1 Filtrage basé sur la théorie de Kalman . . . 21
0.3.1.2 Filtrage de Kalman combiné à l'algorithme Esperance-Maximisation EM . . . 22
0.3.1.3 Filtrage basé sur le principe de la cage de Faraday . . . 24
0.3.2 Resynchronisation de traces: algorithme RM . . . 25
0.4 Chapitre 4: Outils et méthodes d'évaluation . . . 26
0.5 Chapitre 5: Conclusion . . . 26

1 Introduction to Modern Cryptography 30

1.1 **Introduction** . . . 30

1.2 **Secret-Key Cryptography** . . . 31
1.2.1 Symmetric Ciphers . . . 31
1.2.1.1 Stream Ciphers . . . 31
1.2.1.2 Block-Ciphers . . . 31
1.2.2 Standard Secret-Key Systems . . . 32
1.2.2.1 Data Encryption Standard (DES) . . . 32
1.2.2.2 Advanced Encryption Standard (AES) . . . 33
1.3 **Public-Key Cryptography** . . . 34
1.4 **Embedded Cryptography & Vulnerabilities** . . . 34
1.5 **Side-Channel Analysis: General Background** . . . 36
1.5.1 SCA Classifications . . . 37
1.5.2 SCA Algorithms: Typical Description . . . 40
1.5.2.1 Basic Algorithm . . . 40
1.5.2.2 Template Attack Algorithm . . . 44
1.5.2.3 Stochastic Model Attack Algorithm . . . 45
1.5.3 SCA Countermeasures . . . 45
1.5.4 SCA Metrics . . . 46
1.5.4.1 Attack's Efficiency Metrics . . . 46
1.5.4.2 Leakage Quantification Metrics . . . 48

2 Side-Channel Attacks **49**
2.1 **Our Contributions** . . . 49
2.2 **On the Optimality of Correlation Power Analysis** . . . 51
2.2.1 Introduction . . . 51
2.2.2 Notations & Definitions . . . 51
2.2.3 The Optimality From the Historical View Point . . . 52
2.2.4 The Optimality From the Estimation Theory View Point . . . 53
2.2.4.1 The Approximation Problem . . . 53
2.2.4.2 Optimal Linear MMSE Estimation & Connection with ρ 55
2.2.5 Case Study . . . 59
2.2.6 Conclusion . . . 61
2.3 **Combined Side-Channel Distinguishers** . . . 63
2.3.1 Introduction . . . 63
2.3.2 Combination of Distinguishers . . . 63
2.3.2.1 Gini Correlation: A mixture of Pearson and Spearman Coefficients . . . 64

2.3.2.2 Practical Computation of Gini Correlation & Properties 65
2.3.2.3 Pearson-Spearman Combination: An Empirical Approach 67
2.3.2.4 Experimental Results & Discussion 68
2.3.3 Conclusion 69
2.4 **Secret Key Rank Correction** 71
2.4.1 Introduction: Background Knowledge 71
2.4.1.1 Rank-based SCAs 71
2.4.1.2 Notations 71
2.4.1.3 Key Rank Behaviours 72
2.4.2 Rank Corrector: Principle 73
2.4.2.1 Application Field 73
2.4.2.2 Basic Principle 74
2.4.2.3 RC Parameters & Evaluation 76
2.4.2.4 Algorithm Description 77
2.4.2.5 Case Study 78
2.4.2.6 Optimization 80
2.4.3 Experiments & Results 81
2.4.4 Conclusion 82
2.5 **First Principal Components Analysis for Secret Key Recovery** 83
2.5.1 Introduction 83
2.5.2 Principal Component Analysis: Background Knowledge 84
2.5.3 FPCA: Attack Process 84
2.5.3.1 Preliminary Preparation Phase 85
2.5.3.2 References computation 86
2.5.3.3 FPCA distinguisher 86
2.5.4 FPCA on DES Implementations 90
2.5.4.1 FPCA on Unprotected DES 90
2.5.4.2 FPCA on Masked DES 91
2.5.5 Conclusion 92
2.6 **Wavelets Transform based Side-Channel Attacks** 94
2.6.1 Introduction 94
2.6.2 An Understanding of the Multiresolution Analysis 94
2.6.2.1 Fourier Transform Overview 95
2.6.2.2 Short Fourier Transform (STFT) Overview 97
2.6.2.3 Wavelet Transform 98

2.6.2.4 Continuous Wavelet Transform (CWT) 98
2.6.2.5 Discrete Wavelet Transform (DWT) 99
2.6.3 Wavelets for Secret Key Recovery 101
2.6.3.1 Wavelets based CPA 101
2.6.3.2 Wavelets based Template attacks 103
2.6.3.3 Wavelets vs FFT based Template Attacks 103
2.6.4 Conclusion 104

3 Side-Channel Signal Processing 106

3.1 **Introduction & Contributions** 106
3.1.1 The Noise Problem 106
3.1.1.1 Our contributions 107
3.1.2 The De-synchronization Problem 107
3.1.2.1 Our contribution 108
3.2 **Side-Channel Filtering & Patterns Detection** 109
3.2.1 Kalman Noise Filtering 109
3.2.1.1 Kalman Filter Model 109
3.2.1.2 Experiments & Results 113
3.2.1.3 Conclusion 118
3.2.2 Kalman Combined Expectation Maximization Algorithm 119
3.2.2.1 An Overview of Kalman Smoother 119
3.2.2.2 Updating Kalman Parameters with EM Algorithm . . . 121
3.2.2.3 Experiments & Results 126
3.2.2.4 Conclusion 127
3.2.3 Wavelets: A Multiresolution Time-Frequency Analysis 129
3.2.3.1 Wavelets for Cryptographic Patterns Detection 129
3.2.3.2 Wavelets Combined Mutual Information for Side-Channel Traces Filtering 130
3.2.3.3 Conclusion 135
3.2.4 Electromagnetic Shielding 136
3.2.4.1 Electromagnetic Signals: General Background 136
3.2.4.2 Electromagnetic Shielding Overview 137
3.2.4.3 Experiments & Results 138
3.2.4.4 Conclusion 142
3.3 **Side-Channel Signals Re-synchronization** 144
3.3.1 Related Work 144

3.3.2 Effect of Traces Misalignment on SCA . . . 145
3.3.3 Re-synchronization by Statistical Moments . . . 146
3.3.4 Statistical Moments Based Jame's Method Principle . . . 146
3.3.4.1 Adequacy for Side-Channel Analysis . . . 148
3.3.4.2 Resynchronization by Moments (RM): Proposed Algorithm . . . 148
3.3.4.3 Link With POC . . . 150
3.3.5 Experiments, Results & Discussion . . . 150
3.3.5.1 Evaluation Metrics . . . 150
3.3.5.2 Experiments & Results . . . 151
3.3.5.3 Discussion . . . 155
3.3.6 Conclusion . . . 155

4 Side-Channel Security Evaluation & Methodologies 156
4.1 **Introduction & Contributions** . . . 156
4.2 **Certification Schemes & Standards: the Example of Common Criteria** . . . 157
4.3 **Towards a Common Framework for Security Evaluation** . . . 159
4.3.1 Characterization Phase . . . 159
4.3.1.1 SCA Constraints . . . 159
4.3.1.2 On the Choice of the Most Appropriate Analysis . . . 161
4.3.2 Acquisition Phase . . . 163
4.3.2.1 A Practical Example . . . 164
4.3.2.2 Combination of Measurements . . . 164
4.3.3 Pre-processing Phase . . . 168
4.3.4 Simulation Phase . . . 169
4.3.5 Analysis & Decision Phase . . . 171
4.3.5.1 Cautions on the Use of SCA Metrics . . . 171
4.3.5.2 Key-Time Success Rate Metric (KTSR) . . . 172
4.3.6 Methodological Scheme for the Evaluation . . . 173
4.4 **Conclusion** . . . 175

5 Conclusions & Perspectives 176

A Publications & Activities 180

B Appendix **185**
B.1 Adding noise decreases the quality of ρ . . . 185
B.2 Expectation-Maximization components calculation . . . 186
B.2.1 Component α . . . 186
B.2.2 Component β . . . 187
B.2.3 Component γ . . . 188
B.3 Binomial Formulas . . . 189
B.4 Procedure to Obtain the KTSR 2D plot (such as the one of Fig. 4.10) . 190
B.4.1 Identification of the correct and bad samples . . . 191
B.4.2 Compute an attack statistics matrix for k and t . . . 191
B.4.3 Draw the graphs . . . 191

C Glossary **193**
C.1 Acronyms . . . 194
C.2 Notations . . . 197

References **198**

List of Figures

1 Illustration de la métrique SR sur trois attaques différentes. 10
2 Illustration de la métrique GE sur deux attaques différentes. 10
3 Attaques classiques contre attaques combinées. 14
4 Evolution du rang d'une première fausse hypothèse de clé. 15
5 Evolution du rang d'une deuxème fausse hypothèse de clé. 15
6 Evolution du rang de la clé secrète. 16
7 Correction de l'evolution du rang de la clé secrète avec RC. 16
8 Efficacité d'une DPA avec et sans RC. 16
9 Illustration de l'efficacité de la FPCA. 18
10 Comparaison du processus de l'attaque en ondelettes avec un processus de filtrage classique. 20
11 Efficacité de l'attaque sur les coefficients ondelettes en terme de SR et GE. 21
12 Application du filtre de Kalman sur une trace bruitée de AES. 22
13 Points d'inflexion temporel: 1^{ere} limitation de l'analyse. 23
14 Illustration de l'efficacité du filtre KF combiné à l'algorithme EM sur une attaque CPA en terme de GE. 24
15 Identification de motifs d'un AES-128 en mode CBC avec le filtre KF combiné à l'algorithme EM. 25
16 Efficacité de la cage de Faraday sur un AES. 28
17 Resynchronisation de traces DES en utilisant l'algorithme RM. 29
18 Illustration d'une combinaison d'antennes pour l'acquisition des traces. . 29

1.1 First-order success rate for three different attacks. 47
1.2 Guessing entropy for two different attacks. 47

2.1 An example of a linear relationship between X and Y. 57
2.2 An illustration of Heteroscedasticity problem. 58
2.3 An illustration of outliers problem. 58

2.4 Examples of Binomial distributions. 60
2.5 Examples of Binomial distributions with additive Gaussian noise. 61
2.6 Leakage function of Sbox 0 (DPA contest v2). 66
2.7 Leakage function of Sbox 0 extended in (b) to higher values of α. 67
2.8 The mechanism of combination using an aggregate function Ψ. 68
2.9 CPA, Spearman vs Combination: (a) Success Rate and (b) Guessing Entropy. 69
2.10 Examples of rank behaviours for the secret key. 73
2.11 Examples of rank behaviours for false keys. 73
2.12 Rank of SK during a DPA, with and without RC. 75
2.13 Illustration of RC principle. 76
2.14 Illustration of an SCA using RC, at the first threshold. 80
2.15 First-order success rate for DPA with and without RC. 82
2.16 Guessing entropy for DPA with and without RC. 82
2.17 References dispersion for different number of traces 88
2.18 FPCA description. 89
2.19 Unprotected DES Guessing entropy metric. 92
2.20 Unprotected DES First-order success rate metric. 92
2.21 USM DES Guessing entropy metric. 92
2.22 USM DES First-order success rate metric. 92
2.23 Masked-ROM Guessing entropy metric. 93
2.24 Masked-ROM First-order success rate metric. 93
2.25 Fourier transform illustration. 95
2.26 Wavelet transform illustration. 95
2.27 An illustration of 3-levels DWT decomposition. 101
2.28 CPA First-order success rate. 102
2.29 CPA Guessing entropy. 102
2.30 Template attack First-order success rate. 104
2.31 Template attack Guessing entropy. 104
2.32 DWT vs FFT based Template attack: First-order success rate. 104
2.33 DWT vs FFT based Template attack: Guessing entropy. 104

3.1 Kalman process description. 110
3.2 Kalman filter algorithm. 112
3.3 An illustration of a noisy AES trace (basic trace). 114
3.4 AES trace filtered with the Kalman technique. 114

3.5 DES first round measurement without any filtering. 114
3.6 DES first round measurement with Kalman filtering. 114
3.7 An illustration of noise extraction from a DES encryption measurement. 115
3.8 Original trace of three AES-128 encryptions in CBC mode. 126
3.9 KF with 10 EM iterations with $Q = 0.1$, $R = 0.5$. 126
3.10 KF with 50 EM iterations with $Q = 0.1$, $R = 0.5$. 126
3.11 KF with 150 EM iterations with $Q = 0.1$, $R = 0.5$. 126
3.12 KF with 200 EM iterations with $Q = 0.1$, $R = 0.5$. 127
3.13 KF with 300 EM iterations with $Q = 0.1$, $R = 0.5$. 127
3.14 KF with 500 EM iterations with $Q = 0.1$, $R = 0.5$. 128
3.15 KF with 10 EM iterations with $Q = 0.1$, $R = 0.9$. 128
3.16 An illustration of the CWT on three AES encryption blocks. 131
3.17 Computation of the mutual information (MI) on the first-level DWT of unprotected DES traces. 134
3.18 Comparison of 'Donoho', 'Donoho combined MI' and 'Kalman combined EM' efficiency. 134
3.19 Em measurement setup. 138
3.20 Detected frequencies without shielding. 139
3.21 Detected frequencies with shielding. 139
3.22 Shielding technique setup. 140
3.23 AES Sboxes attack using the electromagnetic shielding. 141
3.24 Firs-order success rate metric. 142
3.25 Top view of a horizontal positioning Em antenna. 143
3.26 Front view of a horizontal positioning Em antenna. 143
3.27 Front view of a vertical positioning Em antenna. 143
3.28 An example of several Em antennas. 143
3.29 An example of an Anechoic chamber (this picture is taken from [13]). . 143
3.30 An illustration of misaligned traces 145
3.31 CPA Guessing entropy on misaligned traces. 145
3.32 An example to computing $\mu^{(1)}$. 149
3.33 An illustration of DES traces re-synchronization using RM. 153
3.34 RM (left) versus POC (right) when performed on noisy AES traces. . . 154
3.35 On the choice of the most appropriate re-synchronisation algorithm. . . 155

4.1 An example of decapsulated Virtex-5 FPGA (left). 160
4.2 Coverage of countermeasures for all physical attacks classes. 161

4.3 Leakage model and distinguisher connections. 163
4.4 Venn diagram representation 166
4.5 Calculation of PC for two cases when combination is (a) possible, (b) not possible. 167
4.6 Illustration of windows selection process. 169
4.7 DPA First-order success rate on different windows. 170
4.8 DPA guessing entropy on different windows. 170
4.9 Various correct times definitions, depending on a threshold. This degree of freedom can be assimilated to the rank of a key hypothesis amongst all the candidates. 173
4.10 Progression of an attack metric (here: the success rate) for two different distinguishers. 174
4.11 KTSR metric for DPA and CPA when performed 100 attacks on DES implementation with threshold of 90%. 174
4.12 Evaluation process scheme. 175

5.1 Involvement of wavelet analysis in SCA security aspects. 178

List of Tables

1.1 Various distinguishers suitable for SCA. 39
1.2 Classification of state-of-the-art attacks on cryptographic implementations. 39

3.1 Needed traces to break DES Sboxes. 117
3.2 Comparative results for the "Only time-shifting" case. 152
3.3 Number of traces needed to succeed a CPA attack on the re-synchronized campaigns. 153
3.4 Comparative results for the "Time stretching" case. 154

4.1 No. of traces to attack using C_1, C_2 and combination of both. 168

Résumé en Français

0.1 Chapitre 1: Introduction et Plan de la thèse

Ce travail de thèse concerne la sécurité des systèmes embarqués. Un système embarqué peut être défini comme un ordinateur intégré ou embarqué dans un environnement soumis à de fortes contraintes logicielles et matérielles. Parmi les contraintes classiques, nous pouvons citer la faible consommation, la capacité mémoire réduite, la rapidité et la robustesse. Récemment, nous avons vu s'ajouter la sécurité comme une nouvelle contrainte qui intéresse de plus en plus les ingénieurs et les architectes des systèmes embarqués. En réalité, l'essor des systèmes embarqués a favorisé l'expansion des nouvelles technologies, présentes dans des secteurs importants comme: le secteur militaire, l'éléctronique grand public, le paiement bancaire ou encore la télécommunication. Le plus souvent, les nouvelles technologies stockent et manipulent des informations dites sensibles ou secrètes qui doivent impérativement être protégées; or, récemment le monde des nouvelles technologies a recensé une multitude d'attaques importantes, en dépit de la robustesse des algorithmes cryptographiques (*e.g.* DES, AES, RSA, *etc*) du point de vue mathématique.

Généralement, les attaques physiques recensées peuvent être classées en deux catégories:

1. **les attaques actives** qui intéragissent avec le circuit attaqué, en manipulant par exemple sa consommation ou sa fréquence. Ces attaques peuvent aller jusqu'a l'endommagement du circuit et sont généralement connues sous le nom de "attaques en fautes" (Differential Fault Analysis (DFA)).
2. **les attaques passives** qui se basent sur une exploitation des propriétés physiques du système telsque: la consommation éléctrique ou

électromagnétique, l'information temporelle ou l'information acoustique. Les attaques passives sont connues aussi sous le nom de "attaques par canaux cachés" (Side Channel Attacks SCA). Nous notons que ce travail de thèse se focalise entièrement sur ce dernier type d'attaques (SCA).

Dans la littérature, l'algorithme classique d'une attaque par canaux cachés peut être décrit comme l'ensemble de deux parties complémentaires: Une *partie expérimentale* qui consiste à acquérir des données lors d'un cryptage ou décryptage. Nous appelons par "traces" ou "observation réelle" ces donnés qui contiennent le secret à extraire; et une *partie théorique* qui consiste à étudier l'algorithme cryptographique et construire ce que nous appelons un **modèle de fuite**. Ce modèle théorique sert à prédire l'activité d'une partie du circuit en faisant des hypothèses. Le but est de trouver la bonne hypothèse qui correspond à la clé secrète. En réalité, ces prédictions determinées à partir du modèle de fuite vont nous permettre de classer nos observations réelles dans des partitions. Ainsi, la bonne hypothèse de clé est celle qui correspond au meilleur partitionnement des traces. A ce stade, nous avons besoin d'un test statistique, que nous appelons **distingueur**, et qui va nous aiguiller vers ce meilleur partitionnement. Ce test peut être par exemple une différence de moyenne, une mesure de dispersion comme la variance intra ou inter-partition ou encore l'information mutuelle.

Généralement, la sécurité en tant que concept, elle engendre deux notions étroitement liées: "les attaques" et "les contre-mesures" (ou protections). En réalité, nous pouvons voir ce point de deux façons différentes: soit les attaques existent pour déjouer une contre-mesure, soit des contre-mesure existent pour limiter une attaque. Dans la littérature des SCA, les contre-mesures peuvent être classées selon différentes catégories. Nous citons deux catégories: celle des contremesures qui consistent à équilibrer la consommation et la rendre constante à chaque instant quelque soient les données manipulées par l'algorithme; et celle des contremesures qui consistent à randomiser les données manipulées afin de dissimuler la vraie fuite (le secret). Nous notons que ces deux contremsures agissent directement sur l'architecture de l'algorithme cryptographique. Par ailleurs, nous trouvons des contremesures qui agissent sur le domaine temporel et qui consistent par

exemple à désynchroniser les mesures ou ajouter explicitement du bruit afin de cacher le secret.

Précédemment nous avons dit que les deux notions "attaque" et "contre-mesure" sont liées. Cette liaison se manifeste à travers une phase importante de l'analyse SCA; il s'agit de *l'évaluation*. Il est évident que réfléchir à une nouvelle contremesure passe nécessairement par une phase d'évaluation de l'attaque. Dans ce contexte, cinq aspects d'analyse doivent être pris en compte:

(a) **la caractérisation du circuit**: cet aspect se base sur la documentation du propriétaire du produit.

(b) **la simulation des attaques**: cet aspect se base sur des mesures simulées. Ceci, va permettre de charactériser encore le circuit et déterminer l'attaque la plus appropriée à appliquer sur des mesures réelles.

(c) **l'acquisition des mesures réelles**: cet aspect d'analyse nécessite un minimum d'équipement qui se compose essentiellement d'un oscilloscope, des antennes et des sondes pour la collecte des données et un ordinateur pour le stockage et la manipulation de ces données.

(d) **le pré-traitement des traces acquises**: généralement, deux problèmes majeurs peuvent ce présenter lors de l'analyse des traces: le bruit, et la désynchronisation (non alignement) des traces.

(e) **l'évaluation de l'analyse**: il existe une relation mutuelle entre cet aspect et les aspects "pré-retraitement" et "attaque". En effet, si l'attaque ne marche pas, la qualité des traces doit être re-verifieé, ou bien une nouvelle attaque devra être envisager. De plus, dans le contexte SCA, quelques outils ont été proposés afin d'évaluer l'efficacité des attaques SCA. Généralement, un évaluateur dispose de trois outils ou métriques d'évaluation:

- **le critère de stabilité** qui selon lui, l'attaque est validée si la clé prédite (trouvée par le distingueur) reste stable pour un certain nombre de traces.
- **le taux de succès (SR: Success Rate)** qui représente la probabilité d'avoir la bonne hypothèse de clé en terme de nombre de traces. Dans cet exemple, trois attaques sont impliquées, nous disons que l'attaque 1 (en bleu) est la plus per-

formante en terme de rapidité. les deux autres attaques, 2 et 3, sont équivalentes. Par exemple, pour atteindre un taux de succès de 80% nous avons besoin seulement de 175 traces ou mesures réelle pour l'attaque 1. Tandis que, pour les autres attaques, nous avons besoin de 240 traces.

- **le filtre de rang (GE: Guessing Entropy)** qui représente le rang moyen de la bonne hypothèse de clé en terme de nombre de traces. Dans l'exemple, nous voyons que l'attaque 4 (en vert) atteint plus rapidement le premier rang que l'attaque 5.

Nous notons que les mtriques SR et GE se basent sur la connaissance de la bonne hypothèse de clé; ainsi en pratique elles nécessitent un grand nombre de traces pour pouvoir calculer une probabilité ou une moyenne avec une bonne précision.

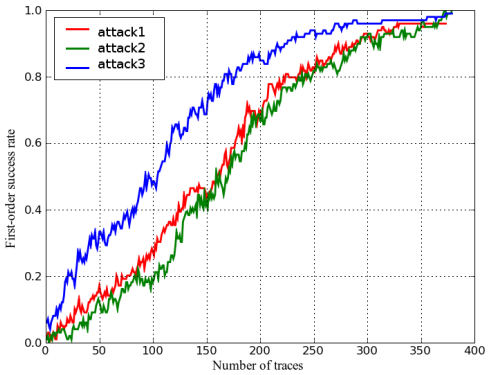


Figure 1: Illustration de la métrique SR sur trois attaques différentes.

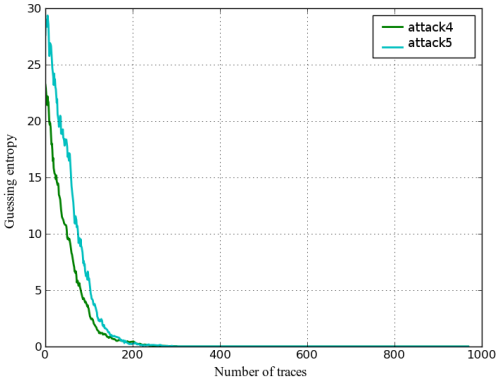


Figure 2: Illustration de la métrique GE sur deux attaques différentes.

Ce travail de thèse est organisé en quatre chapitres: dans le premier chapitre, nous commençons par une introduction générale sur la sécurité matérielle, en détaillant des mots clés comme la cryptographie ou les attaques physiques. De plus, nous introduisons les différentes contributions élaborées durant les trois années de thèse.

Dans le deuxième chapitre, nous proposons de nouvelles attaques SCA (algorithmes et distingueurs) qui sont à la fois génériques et plus efficaces que les attaques basiques. En effet, nous commençons par étudier l'efficacitéé de l'attaque SCA la plus répandue: il s'agit de *l'analyse de consommation*

(Corrélation Power Analysis CPA) qui est basée sur le coefficient de Pearson comme distingueur. Il est nécessaire de bien comprendre le fonctionnement de la CPA et sa capacité à retrouver la clé secrète, afin de pouvoir développer des analyses plus puissantes et plus génériques. A cet effet, la contribution qui suit sert à combiner deux distingueurs SCA. Nous montrons que, sous certaines conditions, de telles combinaisons aboutissent à un distingueur plus précis ce qui permet d'accélérer l'attaque; ainsi de réduire significativement le nombre de traces SCA nécessaires pour la récupération de la clé secrète. En réalité, nous montrons que certaines combinaisons sont directement liées au comportement des hypothèses de clé, en particulier celui de la clé secrète. Ceci est à l'origine d'une nouvelle contribution étudiée dans ce chapitre. Cette troisième contribution consiste à analyser l'évolution du rang de la clé secrète afin de developper un algorithme qui permet de corriger la décision du distingueur durant l'attaque. La quatrième contribution de ce chapitre repose sur un nouveau distingueur, que nous avons noté par FPCA. La pincipale caractéristique de ce distingueur c'est qu'il est multivarié, dans le sens où plusieurs instants de fuite sont considérés dans la trace SCA. D'un point de vue technique, la FPCA est basée sur l'Analyse en Composantes Principales (ACP). Finalement, nous clôturons ce chapitre par une nouvelle attaque basée sur l'analyse en ondelettes, qui a été utilisée initilement comme un outil de pré-traitement de traces dans le contexte des attaques SCA.

Dans le troisième chapitre, nous traitons un sujet crucial dans l'analyse SCA; il s'agit du *pré-traitement des traces SCA* qui intervient généralement comme un aspect d'analyse obligatoire avant l'application des attaques pour la récupération de la clé secrète. Généralement, lors d'une analyse SCA, deux problèmes majeures se présentent: le bruit de mesure et la désynchronisation des traces. Par conséquent, ce chapitre est articulé sur deux contributions principales: le filtrage du bruit et la resynchronisation des traces. Pour la première contribution dans ce deuxième chapitre, nous proposons de nouveaux outils et algorithmes (filtre de Kalman (KF), combinaison de KF avec l'algorithme Espérance-Maximisation (EM), cage de Faraday, analyse en ondelettes combinée à l'analyse de l'information mutuelle) afin de réduire significativement le bruit dans les trace acquises. La question qui se pose ici est comment effectuer un filtrage optimal sans perte de l'information nécessaire pour la récupération du secret. Pour la

deuxième contribution, nous proposons un nouveau algorithme, que nous avons noté par RM, pour la resynchronisation des traces SCA.

Dans le quatrième chapitre, nous developpons un ensemble d'outils et de méthodologies afin d'aider l'évaluateur dans son analyse SCA et lui permettre d'évaluer dans les meilleurs conditions et le plus génériquement possible un système embarqué sécurisé.

Finalement, dans le dernier chapitre nous concluons ce travail et nous ouvrons de nouvelles perspectives concernant l'analyse par canaux cachés.

0.2 Chapitre 2: Attaques par canaux cachés (SCA)

0.2.1 Contribution 1: Optimalité de la CPA

L'attaque CPA, ou analyse de consommation, est sans doute l'attaque SCA la plus populaire vu son efficacité d'une part et sa simplicité de calcul d'autre part. D'un point de vue technique, la CPA est basée sur un calcul du coefficient de Pearson comme distingueur. Dans la communauté cryptographique, plusieurs questions se posent autour de cette attaque; des qustions sur son efficaté par rapport aux autres attaques (*e.g.* la MIA qui est basée sur un calcul de l'information mutuelle ou le coefficient de Spearman qui est basé sur une corrélation des rangs), sur sa relation avec le bruit, et sur son application sur des implémentations protégées. Dans ce travail, nous avons vu que toutes ces questions sont liées et pourraient avoir des réponses objectives. En effet, il suffit d'étudier quand la CPA est optimale; optimale dans le sens où elle est capable de charactériser entièrement la dépendance statistique (forcément linéaire) entre les prédictions théoriques et les observations (traces). A cette fin, nous proposons d'étudier ce problème du point de vue *théorie de l'estimation*. En réalité, nous montrons qu'il existe un seul cas ou l'efficacité de la CPA est optimale: il s'agit du cas Gaussien qui suggère que la distribution jointe des deux variables aléatoires étudiées soit binormale. Dans notre cas, les deux variables correspondent respectivement aux prédictions et aux observations. Etant donné la complexité des calculs engendrée par le développement analytique de la distribution jointe, nous suggèrons de vérifier empiriquement des conditions simples qui satisfont la validité du cas Gaussien. Par ailleurs, d'un point de vue pratique, nous

vérifions la validité de l'étude théorique proposée sur une implémentation de DES non protégée. Au final, nous montrons que plus nous nous éloignons du cas Gaussien (ainsi nous parlons d'un cas non Gaussien), plus la CPA perd son efficacité.

0.2.2 Contribution 2: Combinaison des distingueurs

Dans le cas non Gaussien, d'autres dépendances autre que la linéarité doivent etre étudiées. Dans le contexte de l'analyse SCA, quelques distingueurs ont été proposés; nous pouvons citer la MIA qui est basée sur un calcul de l'information mutuelle qui permet de detecter en général toute forme de dépendance. En revanche, elle engendre des calculs complexes pour l'estimation de la densité de probabilité des données. Nous trouvons aussi un autre concurrent comme le coefficients de Spearman qui mesure une correlation de rangs. Ce coefficient est capable de détecter aussi une relation non linéaire. En réalité ce coeffcient n'est que l'application du coefficient de Pearson sur les rangs des variables. Toutefois, ce nouveau concurrent souffre d'autres problèmes liés par exemple au type et à la valeur prise par les variables. L'effet de tels problèmes devient visible lorsque nous nous approchons du cas Gaussien. En réalité, dans ce cas, il n'y a pas de règle fixe pour determiner qui est le meilleur entre la CPA ou l'attaque base sur le coefficient de Spearman. D'un point de vue pratique, ceci se manifeste à travers des cas o la CPA est plus rapide que l'attaque Spearman et inversement. Pour cette contribution, nous proposons de combiner de deux façons les deux attaques afin de développer une attaque plus puissante et plus générique. La première combinaison que nous proposons est théorique. Elle est basée sur le coefficient de correlation de Gini. La correlation de Gini, a été recemment proposé dans le monde des statistiques et est souvent décrite comme étant une mixture des coefficients Pearson et Spearman. En réalité, Gini dans sa formule implique un calcul de covariance entre les variables et leur rangs. La deuxième combinaison que nous proposons est purement empirique et basée essentiellement sur quelques observations lorsque les attaques CPA et Spearman sont exécutées simultanément sur certaines implémentations. Cette combinaison repose sur une fonction d'aggregation Ψ qui peut être simplement le Maximum ou la Somme et nécessite que les deux attaques soient executées en parallèle. En fait, cette fonction est

appliquée en temp réel sur les valeurs retournées par les distingueurs relatifs aux deux attaques, et pour chaque hypothèse de clé. Dans la Fig. 3, nous mettons en compétition cinq attaques: la CPA, l'attaque Spearman, l'attaque Gini (combinaion théorique), attaque $Comb_{max}$ (première combinaison empirique), et attaque $Comb_{som}$ (deuxième combinaison empirique). Il est clair que, pour les deux métriques SR et GE, les attaques combinées sont nettement meilleures que les attaques classiques (CPA et Spearman).

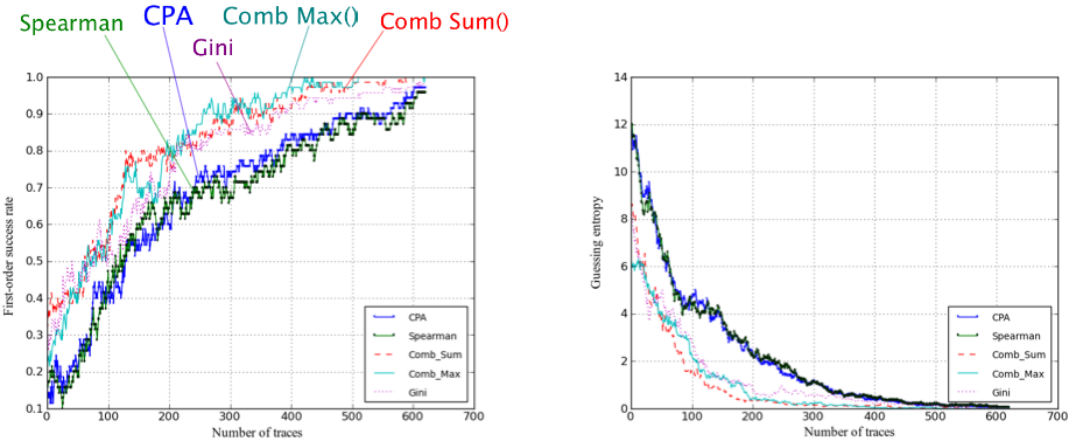


Figure 3: Attaques classiques contre attaques combinées.

0.2.3 Contribution 3: Le Correceteur de rang (RC)

Le choix optimal de la function d'aggregation Ψ ou en general la combinaison des distingueurs repose essetiellement sur le comportement des hypothèses de clé, en particulier la clé secrète, durant l'attaque. Dans cette contribution, nous mettons l'accent sur un tel comportement et de plus proposer un algorithme générique qui permet d'accélérer les attaques SCA. Cet algorithme, que nous avons appelé *le correcteur de rang* (rank corrector RC) consiste à améliorer la décision prise par un distingueur SCA. En revanche, le correcteur de rang, est un algorithme paramétrique qui nécessite deux paramètres qui peuvent être determinés d'une manière précise grâce à une phase de profilage à l'aide d'un circuit clone, qui est souvent disponible dans le contexte de l'évaluation. Algorithmiquement, grâce aux distingueurs, la plupart des attaques SCA mettent à jour le classement de chauque hypothèse de clé pour chaque trace analysée. Si nous nous interessons aux

comportement des fausses hypothèses de clé, deux cas peuvent se présenter:

- Le premier cas est illustré par la (Fig. 4). Nous pouvons voir que l'évolution du rang de la fausse hypothèse de clé est aléatoire dès le début; et n'atteint jamais le premier rang. Une telle clé est facilement cartée par le distingueur SCA.
- Le deuxième cas, illustré par la (Fig. 5), est plus important, vu que la fausse clé peut avoir un comportement non aléatoire qui peut être décrit par trois phases: un décroissement du rang, ensuite une fluctuation autour du meilleur rang pour quelques itérations; enfin, la troisième phase se distingue par un comprtement aléatoire.

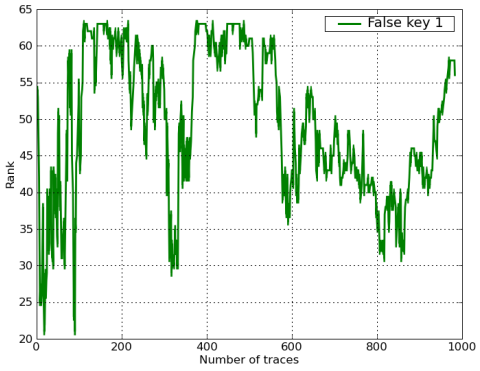


Figure 4: Evolution du rang d'une première fausse hypothèse de clé.

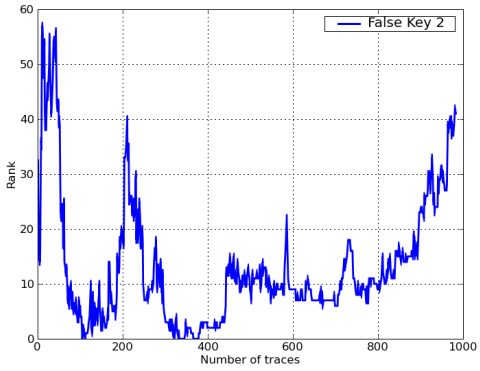


Figure 5: Evolution du rang d'une deuxème fausse hypothèse de clé.

En ravanche, en ce qui concerne la clé secrète, l'évolution du rang est différente (Fig. 6). Elle se distingue généralement par un décroissement, une fluctuation puis une stabilité au niveau du meilleur rang. L'algorithme que nous proposons, le correcteur de rang RC, agit précisément au niveau de la zone de fluctuation du rang. En gros, le but de RC est de corriger d'une manière générique la stabilité de la clé secrète en se basant sur le rang de la clé prédite à chaque itération et l'historique des rangs des hypothèses de clé. Dans cet exemple (Fig. 6 7), en agissant sur cette zone de fluctuation, la stabilité de départ est recalée de 360 à 240 traces, soit un gain de 120 traces.

Comme résultat, nous montrons que l'attaque différentielle multibits DPA avec RC est plus rapide qu'une DPA sans RC. Ceci se manifeste clairement

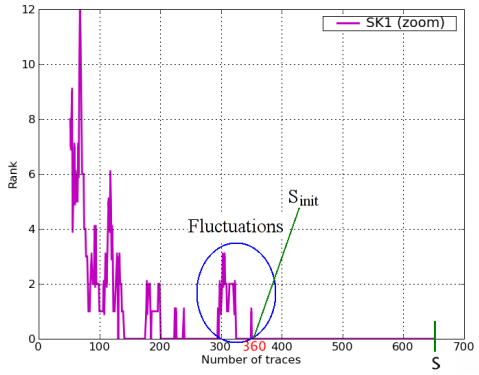


Figure 6: Evolution du rang de la clé secrète.

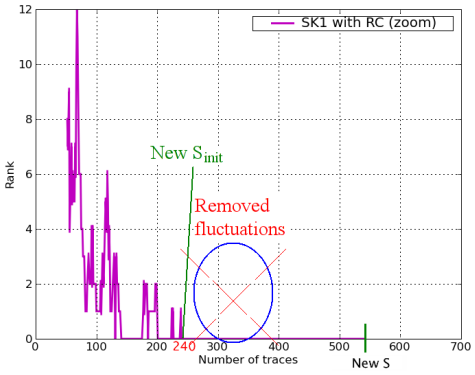


Figure 7: Correction de l'evolution du rang de la clé secrète avec RC.

à travers le SR (Fig. 8). Par exemple, pour atteindre un SR de 80% nous avons besoin de seulement 80 traces pour une DPA avec RC, alors que 120 traces sont nécessaires pour atteindre le même SR pour une DPA sans RC.

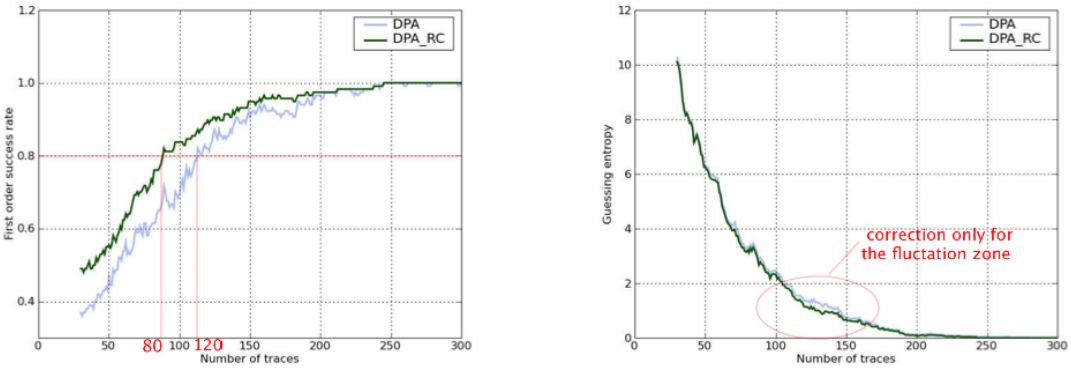


Figure 8: Efficacité d'une DPA avec et sans RC.

0.2.4 Contribution 4: L'attaque multivarivariée FPCA

Ici, nous proposons une nouvelle attaque **multivariée** (FPCA); multivariée dans le sens où plusieurs points de fuite temporels, dans la trace analysée, sont considérés contrairement aux attaques classiques comme la CPA. La FPCA est principalement basée sur un outil performant d'analyse multivariée, il s'agit de l'ACP (Analyse en Composantes Princpales), qui fut initialement utilisée pour le pré-traitement des traces. Ici, l'ACP est utilisée

dans le coeur de l'attaque. Afin de mettre en avant les points forts du distingueur de la FPCA, nous commençons par une comparaison avec une attaque différentielle simple (un seul bit à prédire pour retrouver une partie de clé secrète). Nous considérons un cas d'étude classique: l'attaque d'un DES non protégé. Il s'agit de la prédiction de l'activité d'un seul bit à la sortie d'une sbox, donc deux partitions à construire pour chaque hypothèse de clé. Dans une attaque différentielle monobit (DoM), le distingueur est monovarié car il agit verticalement sur un seul point temporel. Ce distingueur calcule une différence de moyennes comme critère de dispersion entre les deux partitions. Ce calcul est effectué en tout point de la trace; ainsi, un critère de selection (*e.g.* le maximum absolu) est nécessaire pour selectionner la valeur qui correspond au meilleur instant. En d'autres termes trouver l'instant de la fuite. A ce niveau, un biais pourrait exister, car en réalité l'information secrète n'est pas centralisée sur un seul point temporel, elle est répartie sur quelques points génénralement adjacents. En revanche, contrairement à une attaque SCA basique, comme par exemple l'attaque différentielle monobit (DoM) ou multibits (DPA), le distingueur de la FPCA agit non seulement verticalement mais aussi horizentalement puisqu'il prend en compte l'ensemble des points temporels sur lesquels l'information secrète est répartie. Donc, au lieu de calculer une métrique par point temporel, la FPCA calcule une métrique vectorielle qui tient en compte plusieurs points temporels et que nous appelons *référence*. En réalité, notre critère de dispersion est assuré par l'ACP. Plus précisément, il s'agit de la *variance inter-partitions*. Ainsi, nous n'avons plus à nous soucier du critère de selection. Dans la Fig. 2.5, nous illustrons la dispersion des références relatives à la clé secrète et une fausse hypothèse de clé pour 10000 et 81000 traces de DES. Nous précisons que le modèle de fuite utilisé est la distance de Hamming à la sortie des sboxes. Nous avons cinq valeurs de HD que nous avons répartit sur trois partitions. Nous voyons clairement qu'en augmentant le nombre de traces, la dispersion des références relative à la clé secrète augmente aussi. En revanche, cette dispersion diminue pour la fausse hypothèse de clé puisque nous avons fait un mauvais partitionnement de traces.

Afin de mettre en valeur l'efficacité de la FPCA, Nous l'avons comparée avec trois autres attaques (DoM, DPA et CPA) sur une implémentation de DES non protégée. Selon les deux métriques, SR et GE (Fig. 2.5), il est claire que la FPCA est plus efficace globalement que les autres attaques.

Nous notons que d'autres résultats concernant des architectures protégées ont été fournis dans le manuscrit de thèse.

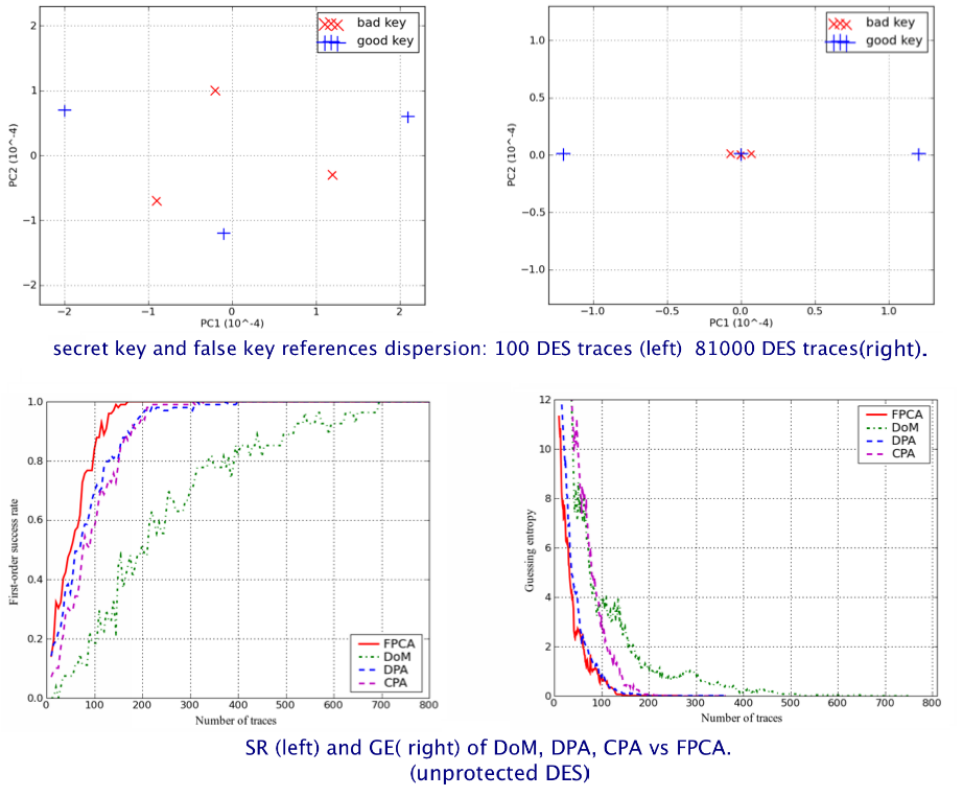


Figure 9: Illustration de l'efficacité de la FPCA.

0.2.5 Contribution 5: Attaque en ondelettes

Cette contribution repose sur une technique générique basée sur l'analyse en ondelettes et qui permet d'accélérer significativement les attaques SCA. Jusqu'à présent, nous avons évoqué que les attaques qui exploitent une dépendance entre des prédictions determinées à partir d'un modèle de fuite et des mesures dans le domaine temporel. En réalité, il est important de noter que l'analyse SCA peut aussi bien s'effectuer dans le domaine temporel que dans le domaine fréquentiel en se basant essentiellement sur les principes de la transformée de Fourier. En revanche, dans la littérature des attaques SCA, il est souvent montré que les analyses fréquentielles sont moins efficaces que les analyses temporelles. Du point de vue évaluteur, les

deux analyses sont utiles puisqu'elles permettent de représenter la même information de façon différente, ce qui permet de révéler plus de détails sur l'implémentation à évaluer. Ici, nous proposons une nouvelle méthode afin de tirer profit des avantages des deux domaines (temporel et fréquentiel). Cette méthode repose principalement sur l'analyse en ondelettes. Contrairement à une analyse de Fourier basique, l'analyse en ondelttes fait appel à de nouvelles formes d'ondes autre que les sinusoides, sur lesquelles elle applique des homothéties pour bien caractériser le signal dans les deux domaines. L'analyse en ondelettes est proposée en deux variantes: analyse continue (CWT) et analyse discrète (DWT).

Pour cette contribution, nous nous intéressons seulement à l'analyse discrète. En réalité, la DWT permet de décomposer le signal en bandes de fréquences. Ainsi, plutôt que d'utiliser différents filtres passe-bande, nous utilisons une combinaison de filtres passe-haut et passe-bas appelés "bancs de filtre" avec des opérations de sous et sur-échanilonnage, que nous appliquons d'une manière itérative. Le signal est alors représenté différemment d'une décomposition à l'autre. En fait, à chaque décomposition, nous obtenons ce que nous appelons "les coefficients approximations" qui représentent les basses fréquences et "les coefficients détails" qui représentent les hautes fréquences.

Généralement, en traitement de signal nous négligeons les coefficients liés au bruit, puis nous faisons une reconstruction du signal vers le domaine temporel. Ici, il est important de noter que c'est au niveau de la reconstruction qu'il pourrait y avoir une perte de l'information. Notre contribution est différente dans le sens où nous arrêtons l'analyse au niveau des coefficients obtenus (Fig. 10). En effet, du point de vue SCA, nous pouvons décrire l'information globale qui contient l'information secrète comme un ensemble de sources superposées. Ainsi, nous avons besoin d'un moyen pour séparer ces sources. D'ou l'idée d'appliquer l'attaque directement sur les coefficients des ondelettes en prenant en compte toutes les décompositions. Dans ce cas, un distingueur SCA va chercher des dépendances entre coefficients ondelettes et prédictions, au lieu de dépendances entre mesures temporelles et prédictions.

D'un point de vue pratique, nous avons conduit l'analyse sur des traces moyennées d'un DES non protégé. Le moyennage des traces ici a pour but de mettre en valeur l'efficacité des ondelettes en tant qu'outil qui agit

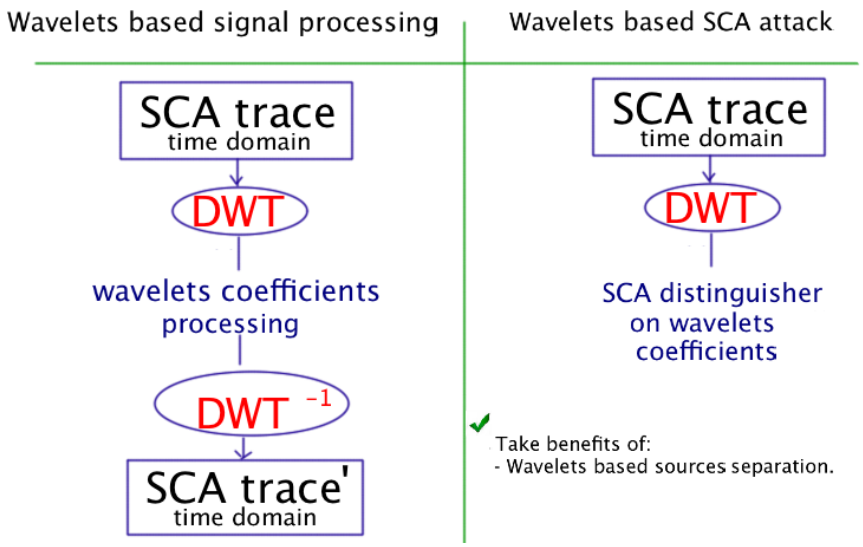


Figure 10: Comparaison du processus de l'attaque en ondelettes avec un processus de filtrage classique.

principalement comme séparateur de sources et non comme un filtre de bruit. Plus précisément, trois analyses ont été impliquées: une CPA sur les traces brutes (notée par *CPA orig* dans la Fig.11), une CPA sur le premier niveau de décomposition en ondelettes (*i.e.* les approximations et les détails notée par *CPA AppDet* dans la Fig.11) et une CPA uniquement sur les coefficients approximations (notée par *CPA app* dans la Fig.11). Selon la Fig.11, l'analyse des traces impliquant les coefficients d'ondelettes est nettement plus efficace qu'une attaque classique sur le domaine temporel. Par exemple, pour atteindre un SR de 90%, nous avons besoin de 150 traces pour la *CPA orig*, et seulement 75 traces pour l'attaque *CPA AppDet* (*i.e.* soit un gain de 50%). Par ailleurs, nous avons remarqué que pour les faibles niveaux de décomposition, comme dans cet exemple (premier niveau utilisé), la fuite engendrée par les implémentations non protégées se trouve essentiellement au niveau des coefficients approximations, contrairement aux coefficients détails qui sont liés au bruit de mesure. Néanmoins, au cours de nos expériences, nous avons remarqué que *CPA AppDet* est souvent plus efficace que l'attaque *CPA App* surtout lorsqu'il s'agit d'une implémentation protégée. Par conséquent, l'évaluateur est tenu de procéder à une analyse *CPA AppDet*, en particulier lorsque les contre-mesures déployées ne sont

pas connues.

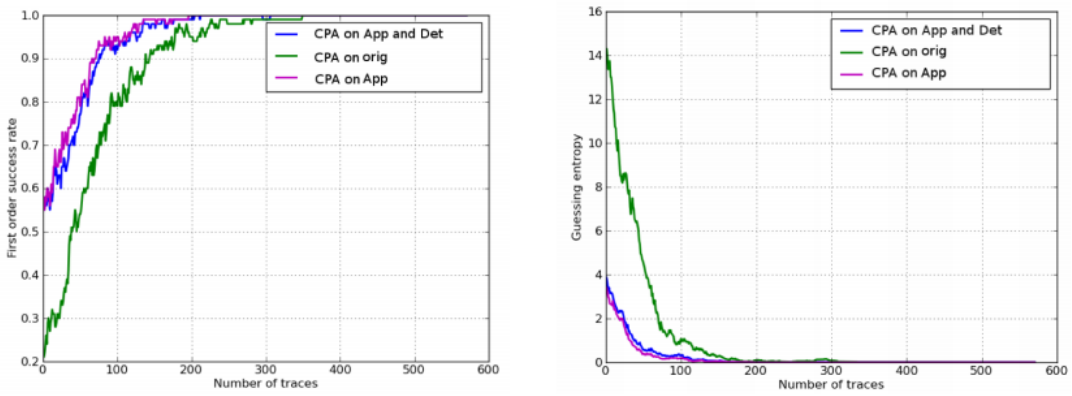


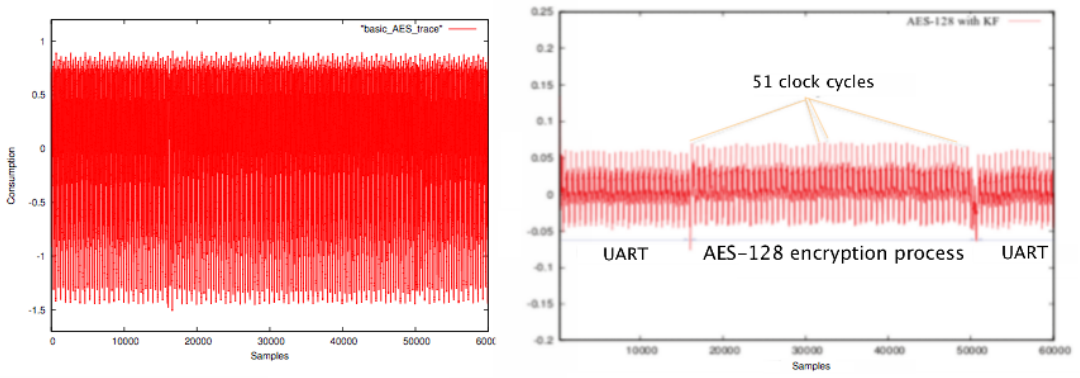
Figure 11: Efficacité de l'attaque sur les coefficients ondelettes en terme de SR et GE.

0.3 Chapitre 3: Pré-traitement des traces SCA

0.3.1 Techniques de filtrage de traces

0.3.1.1 Filtrage basé sur la théorie de Kalman

Le filtre de Kalman qui est largement utilisé dans les sciences de l'ingénieur vu sa simplicité et son efficicacité, sert principalement à estimer une information utile cachée par le bruit. Formellement, il est basé sur un modèle théorique, qui décrit l'evolution de l'information utile dans le temps; et une observation qui décrit comment l'information est réellement observée. Le but du filtre est de calculer en temps réel une estimation de l'information à l'état courant. Dans cette contribution, nous avons commencé par étudier le cas le plus simple qui est le modèle linéaire discret. D'un point de vue algorithmique, le filtre est paramètrique et consiste en deux phases: une phase de prédiction de l'état courant en se basant sur l'estimation précédente; et une phase de correction de la pédiction en se basant sur la mesure courante. Sur des traces réelles acquises à partir d'implémentations cryptographique (AES et DES) sur FPGA, nous montrons qu'un tel filtre permet de distinguer facilement les opérations de l'algorithme cryptographique (nombre de tours, début et fin)(Fig. 12).



An illustration of a noisy AES trace (basic trace).

AES trace filtered with the Kalman technique.

Figure 12: Application du filtre de Kalman sur une trace bruitée de AES.

0.3.1.2 Filtrage de Kalman combiné à l'algorithme Esperance-Maximisation EM

Généralement, le modèle linéaire de Kalman est suffisant pour estimer la forme globale du signal en contrôlant la variance du modèle, un paramètre du filtre qui traduit en réalité notre confiance dans le modèle. Il suffit alors de donner une grande valeur à cette variance, ainsi le filtre va accorder plus de confiance à la mesure qu'au modèle. Une valeur non précise de cette variance pourrait créer une erreur au niveau des points de changement d'évolution temporels (ou points d'inflexions)(Fig. 13). Un tel problème provient du fait que le filtre agit en temp réel. Il prend en considération seulement l'instant présent et les instant passés. Or dans le contexte des attaques par canaux cachés, une telle imprécision pourrait conduire à une perte de l'information liée au secret et par la suite compromettre l'efficacité de l'attaque. Ici, la solution que nous proposons est de faire appel à une technique avancée de la théorie de Kalman, il s'agit de *Kalman smoother* qui prend en compte toute l'information temporelle dans la mesure (passé, présent, futur) pour estimer l'état présent.

Le deuxième incovenient du modèle linéaire réside au niveau de l'estimation des paramètres du filtre. En effet, pour une detection simple des motifs d'une trace, il suffit de chercher manuellement les paramètres en fixant quelques paramètres et variant d'autres. Or lorsqu'il s'agit d'une attaque

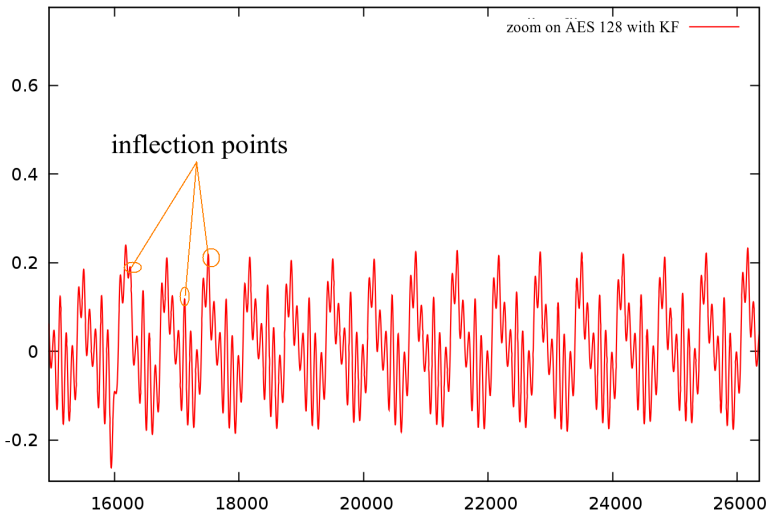


Figure 13: Points d'inflexion temporel: 1^{ere} limitation de l'analyse.

impliquant plusieurs traces (ce qui est le cas généralement); nous ne pouvons pas se permettre de chercher manuellement les paramètres qui correspondent à chaque trace. D'ou la solution que nous proposons est d'utliser un estimateur automatique de ces paramètres: il s'agit de l'algorithm EM. Finalement, pour palier les problèmes cités, nous proposons de combiner d'une manière itérative les algorithmes EM et Kalman smoother.

D'un point de vue pratique, nous montrons l'efficacité d'une telle combinaison sur des traces réelles d'une implémentation AES. En effet, d'après la Fig. 14 qui illustre l'efficacité de la méthode à travers la métrique GE, il est évident que la CPA classique, quand elle est appliquée sur 500 traces bruitées d'un AES $128-bit$ est beaucoup moins efficace, qu'une CPA appliquée sur les même traces mais filtrées avec la technique proposée (*i.e.* KF combiné à l'algorithme EM).

Dans une autre expérience (Fig. 15), nous montrons comment le nouveau filtre est capable de révéler clairement l'activité de l'algorithme implémenté (trois cryptages d'AES-128 bit en mode CBC) au bout de quelques itérations. Ici, Q et R sont les paramètres initiaux du filtre; il s'agit respectivement de la variance du modèle et la variance du bruit.

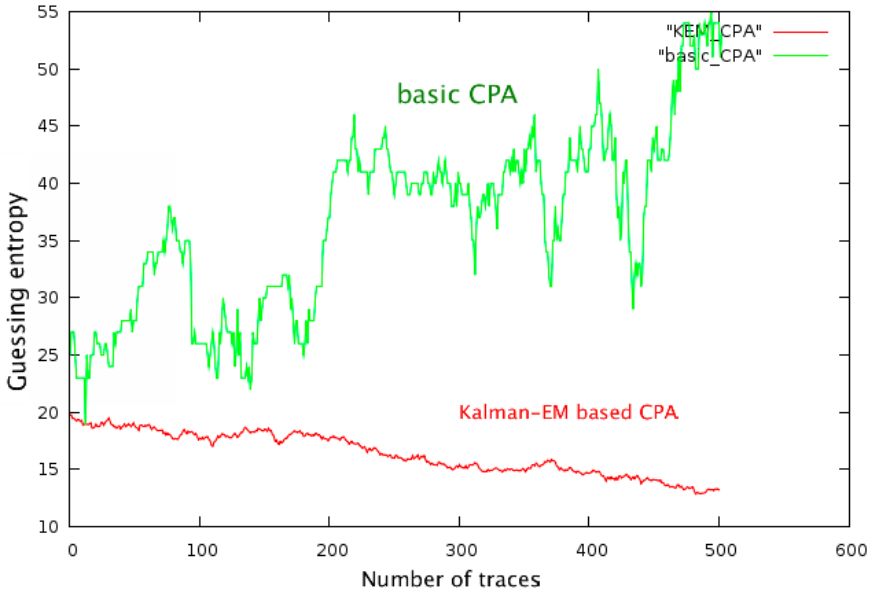


Figure 14: Illustration de l'efficacité du filtre KF combiné à l'algorithme EM sur une attaque CPA en terme de GE.

0.3.1.3 Filtrage basé sur le principe de la cage de Faraday

Dans cette section, nous présentons une technique matérielle (non algorithmique) basée sur les principes des lois de Faraday. Cette technique permet principalment de réduire l'impact du bruit de mesure sur les attaques SCA.

Nous avons vu que l'acquisition des traces constitue un aspect important lors d'une analyse SCA. Dans la littérature, il existe généralement deux techniques d'acquisition de données selon le matériel utilisé (antennes ou sondes) : analyse de courant ou analyse électromagnétique. Nous nous intéressons principalement aux attaques électromagnétiques qui sont souvent perturbées par le bruit de mesure.

A cet effet, nous avons concu quelques prototypes de la cage de Faraday afin d'isoler l'antenne et le circuit, dans lequel se trouve le secret, du milieu extérieur (perturbations des ondes électromagnétiques). Nous montrons qu'un tel système permet de réduire efficacement l'impact des perturbations et rendre l'attaque beaucoup plus rapide.

En effet, du point de vue pratique, nous avons mis en valeur l'efficacité du système sur des traces AES moyennées sur plusieurs niveaux (16x, 256x

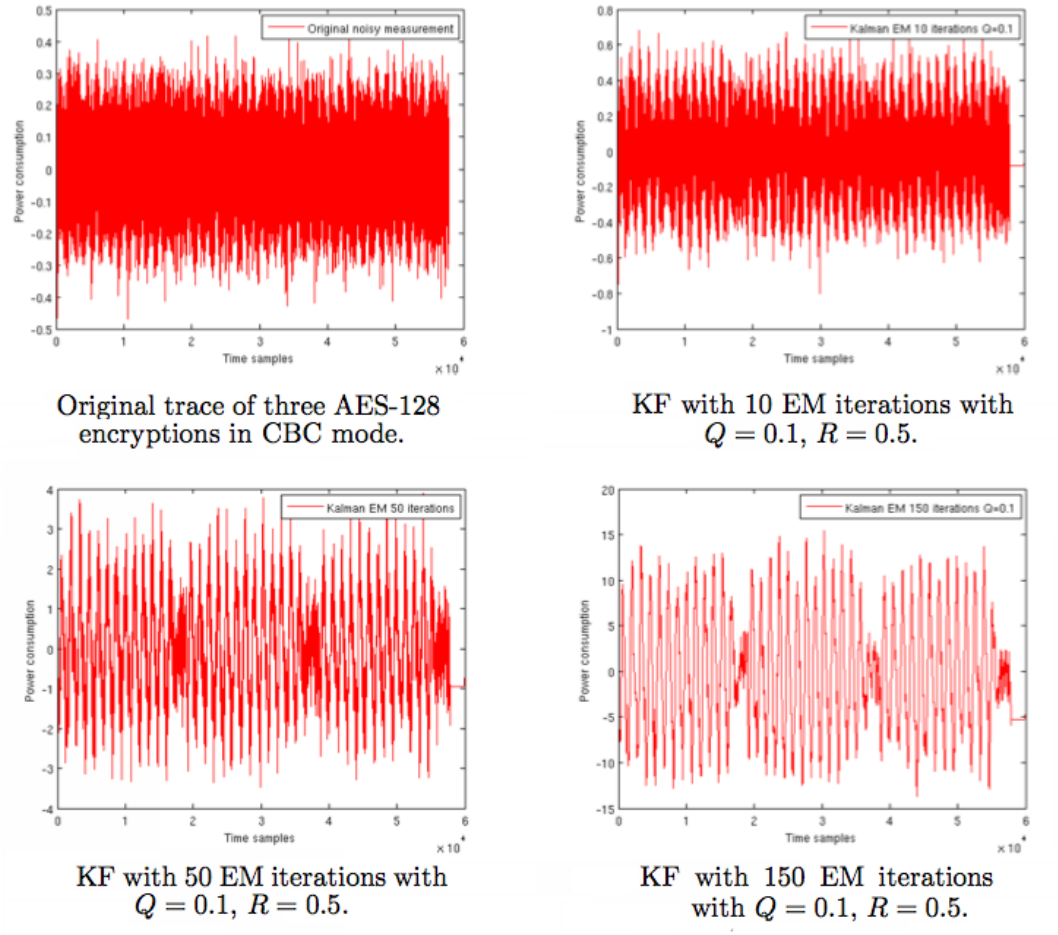


Figure 15: Identification de motifs d'un AES-128 en mode CBC avec le filtre KF combiné à l'algorithme EM.

et 1024x) (Fig. 16). Nous montrons par exemple que lorsque les traces ne sont pas moyennées (1x), alors avec la cage de Faraday nous arrivons comme même à casser certaines sboxes {0,2,8,10,14}. Contrairement à une analyse électromagnétique simple (sans cage), qui ne parvient pas à casser de sboxes. Par ailleurs, il est évident de noter que plus le moyennage est grand, plus nous arrivons à casser des sboxes.

0.3.2 Resynchronisation de traces: algorithme RM

Dans ce travail, nous avons étudié principalment la forme la plus répandue des attaques SCA: il s'agit des attaques qui agissent verticalement sur un

ensemble de traces. Ainsi, toute désynchronisation des traces, va necessairement affaiblir l'attaque. Pour cette nouvelle contribution, nous proposons un nouveau algorithme (RM) qui permet de corriger une telle désynchronisation et améliorer la qualité des attaques SCA.

D'un point de vue technique, nous considèrons que chaque trace peut être représentée par une base temporelle incluse dans un support temporel plus grand. Dans cette base, deux facteurs sont à considérer: le facteur translation et le facteur dilatation. Si nous considèrons deux traces désynchronisées alors il existe une fonction linéaire qui lie les deux traces. Le but de l'algorithme que nous proposons est de trouver tous les paramètres de cette fonction pour un ensemble de traces.

0.4 Chapitre 4: Outils et méthodes d'évaluation

Dans ce chapitre, nous regroupons les méthodes et les algorithmes détaillés dans les chapitres précédants dans un flot d'analyse (une boite à outils) structuré qui pourrait être utilisé à des fins d'évalution de la sécurité des systèmes embarqués contre les attaques par canaux cachés. Plus précisément, nous clarifions le lien etre les différentes phases d'analyse: caractérisation du circuit, simulation de l'attaque, acquisition et pré-traitement des traces; et évaluation. De plus, deux nouvelles techniques sont proposées pour enrichir les phases acquisition et pré-traitement. La première technique qui concerne la phase acquisition s'agit d'une combinaison de différentes méthodes de mesure (Fig. 18). Nous montrons qu'une telle combinaison permet non seulement d'avoir plus de détails sur la nature de l'algorithme implémenté, mais aussi de rendre les attaques SCA plus rapides. La deuxième technique concerne la phase évaluation; il s'agit d'une métrique graphique qui permet d'évaluer l'efficacité des attaques SCA.

0.5 Chapitre 5: Conclusion

Ces dernières années, la sécurité des systèmes embarqués a fait l'objet de recherches intensives. Comme l'énergie, le coût et la performance; la sécurité est un aspect important qui doit être considéré tout au long du processus

de conception d'un système embarqué. Des menaces récentes appelées "attaques par canaux cachés" (Side-Channel Analysis (SCA)) ont attiré beaucoup d'attention dans le milieu de la sécurité embarquée. Ces attaques exploitent des propriétés physiques, tels que la consommation d'énergie ou le champ magnétique rayonné afin de retrouver le secret. De plus, elles sont passives dans le sens où l'analyse se contente d'une obsevation extérieure du système sans l'endommager. Dans ce contexte, il est évident que la sécurisation des systèmes embarqués contre les attaques SCA constitue un aspect vital dans le flot de conception. Par conséquent, la nécessité d'assurer et d'évaluer la robustesse des systèmes embarqués contre ces attaques devient clair.

Cette thèse propose principalement des techniques et méthodes **génériques** dans l'analyse par canaux cachés. Ces techniques qui touchent à différents aspects de l'analyse SCA (acquisition, pré-traitement, attaque et évaluation) peuvent être utilisées dans un cadre d'évaluation plus officiel telque les Critères Communs (CC) ou le FIPS-140 afin d'améliorer la visibilité de l'évaluateur. Par ailleurs, le propriétaire d'un produit pourrait aussi se baser sur ces techniques dans le but d'évaluer la sécurité de son produit face aux attaques par canaux cachés avant de solliciter un certificat.

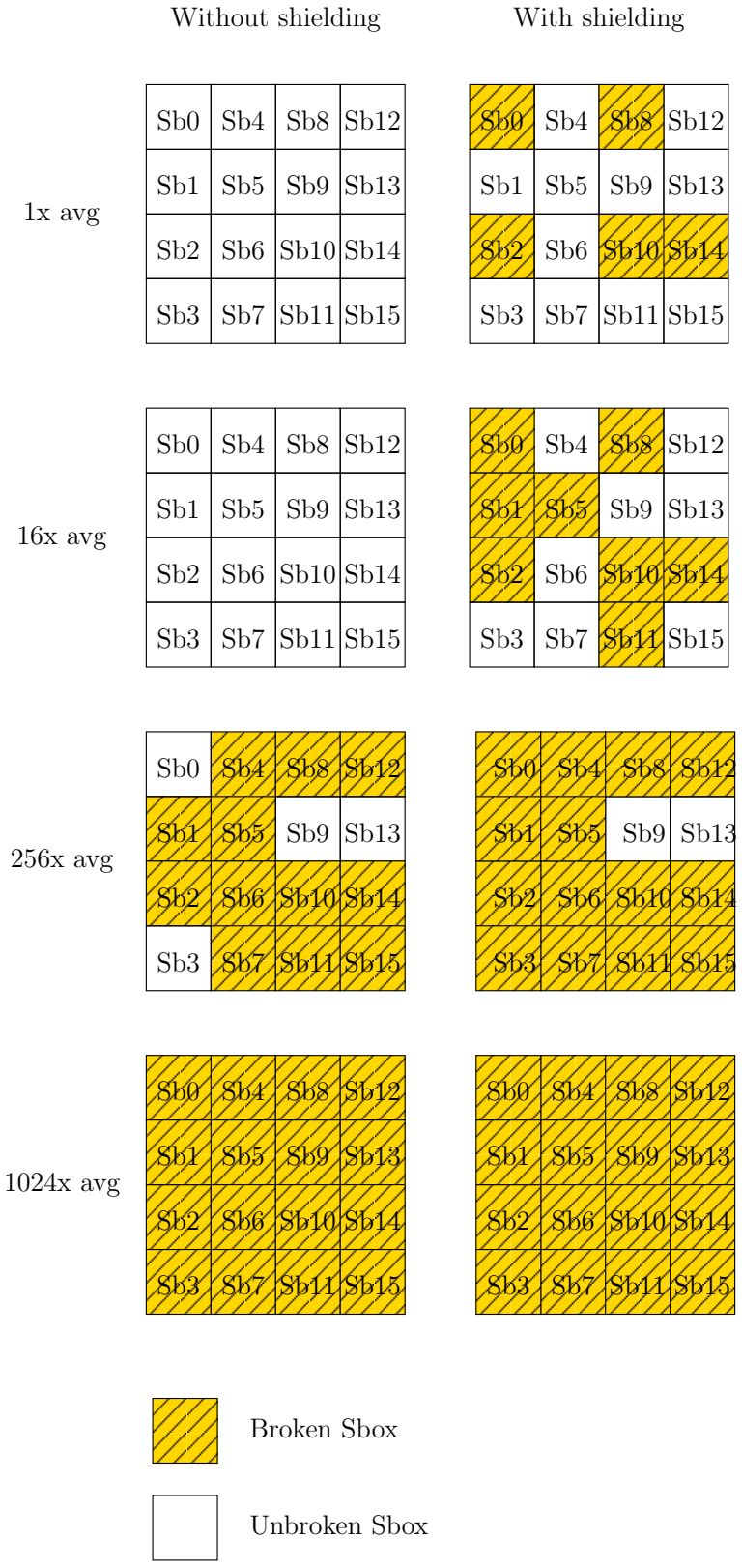


Figure 16: Efficacité de la cage de Faraday sur un AES.

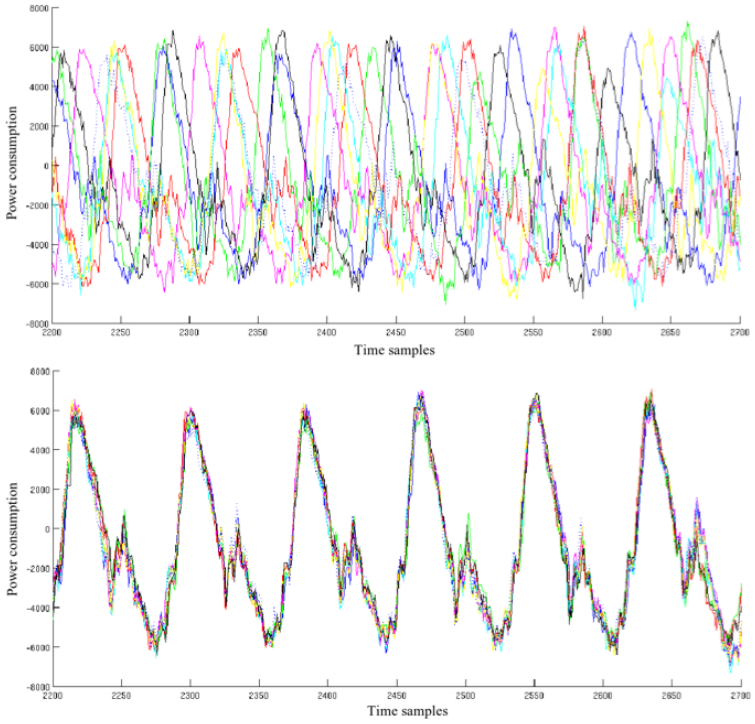


Figure 17: Resynchronisation de traces DES en utilisant l'algorithme RM.

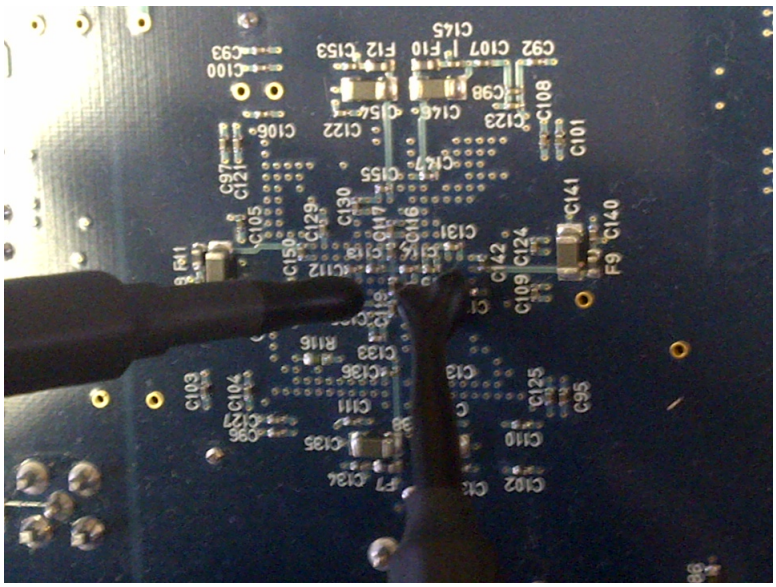


Figure 18: Illustration d'une combinaison d'antennes pour l'acquisition des traces.

Chapter 1

Introduction to Modern Cryptography

1.1 Introduction

Cryptography can be defined as the science of protecting sensitive information. Cryptography is intimately connected to a second term, namely *Cryptanalysis*, which aims at breaking cryptographic means and reading the secret information. Often, the term *Cryptology* is used to involve both of these aspects. Basically, Cryptography aims at providing four principal security aspects: **privacy** or **confidentiality**, **authenticity**, **integrity**, and **non-repudiation**.

- **Confidentiality** It authorises the access for only allowed parties (or users).
- **Authenticity** It allows different parties entering into a communication to identify each other (source/destination).
- **Integrity** It guarantees that the message is properly transmitted from the source to the destination.
- **Non repudiation** It allows controlling message acknowledgment.

Depending on the system to secure and the nature of the secret information, usually one or all of the mentioned security aspects, that Cryptography can provide, are required.

1.2 Secret-Key Cryptography

1.2.1 Symmetric Ciphers

Secret-key cryptography is where a unique key is shared between the source and the destination. For this reason, Secret-key Cryptography is also called *Symmetric encryption*. Obviously, the main issue with Symmetric encryption relies on the distribution of the secret key. Basically, Secret-key cryptography schemes are classified into *stream ciphers* and *block ciphers*.

1.2.1.1 Stream Ciphers

A stream cipher aims at transforming the same plaintext bits sequence into a different ciphertext bits sequence for each encryption.

In the open literature, there is two common stream ciphers used in practice:

- **Self-synchronizing stream ciphers** It computes each bit in the key-stream as a function of the antecedent bits in the key-stream. Nonetheless, any error occurring in the sending of the message will corrupt all the ciphertext.
- **Synchronous stream ciphers** It produces the key-stream in a manner independent of the sent message. Nonetheless, the same key-stream generation function is used both for the source and the destination. Such stream ciphers are not carried about transmission errors.

From the designer point of view, stream ciphers are suitable for hardware implementations; and one can expect an increasing use of these ciphers in future.

1.2.1.2 Block-Ciphers

Basically, Block-cipher algorithms are used for encryption and decryption purposes. They aim at dividing a message into blocks of bits, which are then processed by several mathematical functions (linear and non linear). Actually, the blocks of bits processed by Block-cipher algorithms can have different sizes (*i.e.* 64-*bits*, 128-bits, *etc.*). Theoretically, it has been shown that Block-ciphers are very robust and hard to be broken. Block-ciphers are mainly related to four common modes of operations [106]:

- **Electronic Code Book (ECB) mode** *"The same block of ciphertext is always generated for a given block of plaintext and a given key"* [157]. ECB mode is often used for small sizes of input block, such as encrypting and protecting secret keys.

- **Cipher Block Chaining (CBC) mode** This mode makes the being processed ciphertext block dependant on plaintext blocks previously processed. In fact, the plaintext block currently processed is XORed[1] with the previous ciphertext block before being encrypted. Besides, an initialization vector must be used for the initial block, in order to guarantee the uniqueness of the message.

- **Cipher Feedback (CFB) mode** Data are encrypted in new blocks smaller than the initial block size. CFB mode operates in the same way as the self-synchronizing stream cipher.

- **Output Feedback (OFB) mode** It operates in a manner to guarantee the uniqueness of generated ciphertext blocks. This mode is structurally similar to synchronous stream ciphers.

Generally, each mode aims at managing the way a block-cipher will operate. In practice, there is no rule to determine which is the best mode of operation to use. In fact, the choice of one mode depends essentially on the needed requirements and security fonctionalities. Up to this point, we refer the reader to [106] for more details about modes of operation and more in depth discussion on the choice of the most appropriate mode.

1.2.2 Standard Secret-Key Systems

1.2.2.1 Data Encryption Standard (DES)

The Data Encryption Standard (DES) [104] is the most commonly used symmetric-key block-cipher. In fact, it is largely used by financial institutions and government applications. The DES structure is basically related to two associated designing concepts: the *Feistel function f*[2] and the *Feistel scheme*. First, Feistel function basically aims at building a robust encryption function by combining different operations, which individually are not capable of providing the wanted level of security. Actually, with

[1] XOR stands for exclusive OR.
[2] In the open litterature, the Feistle function is also called *the inner function f*.

the Feistel function, the level of protection is efficiently improved. Combined operations include linear and non linear transformations. Second, the Feistel scheme aims at making both encryption and decryption symmetrical structures. This has a pure designing flavour since it greatly simplifies hardware implementation, as both encryption and decryption processes are mixed together, which relaxes considerably the space constraint. In addition to the sixteen rounds, the DES structure is composed of an initial permutation IP and a final permutation FP, which is the inverse of IP. DES processes plaintext blocks of 64 bits and produces 64-*bit* ciphertext blocks. The size of the secret key SK is 64 bits. Only 56 bits are used by the algorithm. More precisely, there are 8 bits (bits 8, 16,..., 64) that are used to verify parity. Then, these bits are ignored. Now, from the input key, sixteen 48-*bit* sub-keys k_i are computed, one for each round. Before the main rounds, the 64-*bit* plaintext is divided into 32-*bit* halves L0 and R0 and processed alternately thanks to the Feistel scheme. Within each round, eight 6-to-4 *bit* substitution operations, namely *S-boxes* S_i, are used. Note that, each round is functionally equivalent, taking 32-*bit* inputs L_{i-1} and R_{i-1} from the previous round and producing 32-*bit* outputs L_i and R_i for i in [1..16], as follows:

$$L_i = R_{i-1}, \tag{1.1}$$

$$R_i = L_{i-1} \oplus P(S(E(R_{i-1}) \oplus k_i)) = L_{i-1} \oplus f(R_{i-1}, k_i) \ , \tag{1.2}$$

where the $\oplus$ symbol is the exclusive-OR (XOR) operation, E is a fixed expansion permutation mapping R_{i-1} from 32 to 48 bits, P is another fixed permutation on 32 bits, and f the Feistel function. The whole process (16 round operations) is bounded by the initial permutation IP, which precedes the first round, and its inverse FP, which directly follows the last round.

1.2.2.2 Advanced Encryption Standard (AES)

In Secret-key cryptography, the Advanced Encryption Standard (AES) [105], also known as Rijndael, is a famous block-cipher too, which is designed by Joan Daemen and Vincent Rijmen. AES is a U.S. encryption standard that is developed essentially to be an alternative to the Data Encryption Standard DES. AES processes data using blocks of 128 bits length, and a variable secret key length (128, 192 or 256 bits). Hence, as specified by the standard, three different block-ciphers can be used: AES-128, AES-192, AES-256. From the structural point of view, AES operates on a 4x4 matrix of bytes, usually referred to as the state. Each round of AES is composed of four stages:

- **Sub-byte** The Sub-byte modifies each byte in the state using an 8-*bit* substitution box, often called Sbox. From the mathematical view point, Sub-byte function is a non linear operation.
- **Shift-Rows** The Shift-Rows rotates the bytes in each row of the state.
- **Mix-Columns** The Mix-Columns is a linear transformation that operates on the column of the state. Note that this transformation is omitted for the last round.
- **AddRoundKey** The AddRoundKey mixes the state with a sub-key. The sub-key is basically generated from the initial key (or the input key) using what we call key generator module.

1.3 Public-Key Cryptography

Public-key cryptography is also called *asymmetric cryptography* as it does not use the same key for the two processes: encryption and decryption. Consequently, such cryptography is not concerned about the key distribution issue as it is the case for symmetric encryption. Indeed, in Public-key cryptography, two separated and dependent keys are involved: *the public key* and *the private key*. More precisely, the public key is used for encryption; and alternatively the private key is used for decryption. For technical details about Public-key Cryptography, we refer the reader to [37; 125].

1.4 Embedded Cryptography & Vulnerabilities

An embedded device is an electronic system that is mainly designed for fast, robust and specific purposes. When designing an embedded system, both hardware and software competencies are usually met together. Therefore, embedded systems designers must have a thorough knowledge of both the advantages and limitations of the system's architecture. Consequently, designing for an embedded device is different from designing a personal computer (PC).

Nowadays, embedded systems are omnipresent in our daily life (telecommunication systems, consumer electronics, *etc.*). But more importantly, the majority of recent technologies involving embedded systems, such as mobiles phones and smart cards, require a certain level of security to work properly. The physical security has always

been an open question and usually treated as an integral part of embedded system design. Indeed, any violation of embedded systems security could lead to the loss of sensitive and personal information. This would be more critical if we were simply dealing with military and defense market that have always been ruled by high reliable embedded systems, often called Systems on Chips (SoC) such as ASICs and FPGAs.

Attacks on embedded systems involving security functionalities are growing at rapid pace; and can be basically categorised into **passive** and **active** attacks, as it is well described by Stefan Mangard et al. in their book [86].

- **Passive attacks** which exploit the physical properties leaked from the system during a cryptographic process, in order to reveal the sensitive information. Physical properties can be for instance power/electromagnetic consumption and timing execution.

- **Active attacks** which are based on the manipulation of the secure embedded system (inputs, access, *etc.*). Such manipulation (or tampering with the system) aims at making the behaviour of the system abnormal, which can be exploited to retrieve the sensitive information.

In the literature [86], a second classification of attacks deals with the several interfaces (logical / physical) of embedded systems that are exploited by the attack. Depending on the way of accessing these interfaces, attacks can be classified into **invasive**, **semi-invasive** and **non-invasive** attacks.

- **Invasive attacks** They are very strong attacks that target secure embedded systems. For such attacks, the sensitive information is retrieved at all costs. Basically, an invasive attack usually includes the decapsulation of the system and the probing (altering) of signals. Besides, such attacks require, in general, sophisticated (expensive) materials.

- **Semi-invasive attacks** They also involve the decapsulation of the system. Nonetheless, the system surface is not altered at the opposite of invasive attacks.

- **Non-invasive attacks** They do not tamper with the system, in the sense that they target only the accessible parts of the system. Such attacks pose a serious threat to embedded systems as they are low cost and easy to mount (relatively to an experienced attacker). In the literature, we can distinguish **active non-invasive** attacks often called **Fault Attacks** (but without de-packaging

the system), and **passive non-invasive** attacks often called **Side-Channel Attacks**, that principally exploit unintentional physical leakages (timing information, power consumption, *etc.*).

1.5 Side-Channel Analysis: General Background

Side-Channel Attacks (SCA) can be defined as any attack exploiting unintentional physical information leaked from a cryptographic device, without tampering with the system. Actually, during a cryptographic process, the device is likely to leak sensitive information, that can be timing information, power consumption, electromagnetic radiations, sound leaks, *etc.* SCAs are passive attacks, in that the device under attack is not aware of its leaks being recorded.

The idea of analysing the physical properties (electromagnetic radiations) of embedded systems dates back many years ago. For instance, we mention the US project, so-called TEMPEST, that provided solutions to counteract the electromagnetic emanations based attacks. These analyses were also studied and revisited by W.V Eck in [42]. In the mid-1990s, Public-key implementations were the target of a non-invasive attack based on the analysis of the time calculations while running the cryptographic instructions. Such analysis, named *Timing attack*, was proposed by Paul Kocher in [71]. In the late of 1990s, Paul Kocher et .al proposed in [72] two variants of power consumption based Side-channel attacks: Simple Power Analysis (SPA) and Differential Power Analysis (DPA). Authors in [72], showed the efficiency of such analyses to break a DES implementation. Simple Power Analysis (SPA) that is "*a direct analysis of patterns of instruction execution, obtained through monitoring variations in electrical power consumption of a cryptographic algorithm*", as defined by P.Kocher. In other words, SPA aims at revealing the secret by analysing the cryptographic patterns within one Side-channel measurement. As for the Differential Power Analysis (DPA), it involves statistical computations and several Side-channel measurements, contrarily to SPA. After its first publication, DPA has been markedly improved from the statistical view point ([29], [35]). An alternative to DPA was suggested by Brier [24] called Correlation Power Analysis (CPA) that is principally based on linear correlation analyses. CPA offers more efficient analysis by eliminating the "ghost peak" problem in DPA. Recently, new powerful variant of Side-channel attacks, called *Mutual Information Analysis* (MIA) [49], have been proposed. These attacks that are based on an information theoretic approach aim at exploiting both linear and non linear correla-

tions. This fact makes these attacks more generic and more efficient than first-order attacks like DPA and CPA, as they can even be applied on protected implementations. One other kind of attacks called *Template analysis* are often considered to be the most powerful SCA, if certain conditions are satisfied. Indeed, it is shown that such attacks can easily break cryptographic implementations and countermeasures which security is dependent on the assumption that an adversary cannot obtain more than one or a limited number of Side-channel traces. However, these attacks require that an adversary has access to a clone device on which he can perform his trials and tests. Actually, he is first led to profile the clone device by building what we call templates. Second, those templates are used to recover the *secret key* from a real cryptographic co-processor.

Generally, for all SCAs, the leaked information can be statically defined by a continuous random variable for which the probability law P_{law} is unknown or uncertain. The main challenge of SCA is to make a sound estimation of P_{law} without loss of information. Basically, random variables are measured and analysed in term of their statistical and probabilistic features. Obviously, taking into account the high variety of existing attacks, there are many ways to play statistics in the Side-channel field. For instance, new calculations, based on the second order statistics (the variance) seem to be a good solution to quantify the secret information on some protected implementations. As a matter of fact, those calculations have already been exploited to mount an efficient attack called *Variance Power Analysis (VPA)* [78; 144]. Nonetheless, playing statistics is a task often guided by certain conditions to get accurate results. For instance, attacks based on the mutual information theory like MIA, require a reliable estimation of the probability density function of P_{law}. Theoretically, an accurate probabilistic statistic, such as the entropy measure, describes better one random variable than high-order statistics. Unfortunately, the optimal accuracy is hardly achieved specially when the probability law is unknown or uncertain. Actually, in the open litterature of statistics, it is shown that the probability density of an unknown law is nearly impossible to properly estimate, especially when the available data (*i.e.* Side-channel traces in the SCA context) to be studied are limited.

1.5.1 SCA Classifications

Side-channels can basically be classified in two categories: those where the duration of the cryptographic process is the leakage source, and those where a physical quantity depending on time is leaked. In the first case, for every invocation of the cryptographic primitive, a scalar is measured, whereas in the second case, many samples are collected,

to obtain a trace. However, in both cases (*i.e.* the observations are scalar or vectorial), the SCA unfold according to a classical cryptanalytic scenario, that is mainly composed of:

1. A **leakage model** to manage the partitioning of the acquired Side-channel observations, which depends on the scenario (known/chosen plaintext/ciphertext), algorithm (to explore the internal rounds by guessing manageable parts of the secret) and the implementation (software or hardware, pipelined or unrolled, protected or not, *etc*). The most commonly used power models for characterizing the power consumption are the Hamming Distance (HD) and the Hamming Weight (HW) [24].

2. A **distinguisher** to select the most relevant partitioning, amongst all the hypotheses on the secret. The distinguisher is basically a statistical tool, that aims at putting forward any bias. Basically, the first statistical formalization of DPA as a distinguisher was proposed by J.S Coron, D.Naccache and P.Kocher in [35]. Since such formalization, extensive research has been proposed to develop more powerful distinguishers. Generally, distinguishers can be for instance a difference of means [72], a covariance [59], a correlation (linear [24] or rank-based),mutual information [50] or variance [144].

   It is noteworthy to mention that the notion of distinguishers for differential attacks for instance, is initially derived from the *Timing attack* [71]. Actually, the Timing attack acts horizontally on the measurement by measuring the variability between distant executions in time (time variability). Indeed, the Timing attack aims at minimizing such variability to recover the value of one bit of the secret key. Similarly, differential attacks like DPA, analyse the variability but vertically, considering several measurements. But more importantly, the attack is still functional even if the time variability is constant. In fact, it analyses the relation between the vertical variability and a leakage model. A notable difference, between the two concepts, is that Timing attack requires a clone device unlike the general case for differential attacks.

A recent study shows that all monovariate distinguishers are equivalent asymptotically [87] (*i.e.* they are sound), and that they only differ by statistical artifacts that are data-dependent when the environmental noise tends to zero. Typically, the options for choosing a distinguisher are listed in Tab. 1.1.

In the SCA litterature, another classification of Side-channel attacks can be found. Indeed, SCAs can be classified according to the hypotheses their realization require

Table 1.1: Various distinguishers suitable for SCA.

Distinguisher	**Decision**	**Comments**
Difference of means (DoM)	Max.	Models are called "selection functions" [72]; refinements are provided in [93].
Covariance	Max.	Introduced initially as the multi-bit generalization of the DoM [20].
Correlation	Max.	Variants are Pearson [24] (often noted "ρ"), Spearman [19] or Kendall ("τ") correlation coefficients.
Likelihood	Max.	Used when probability density functions (*pdf*) can be estimated, and leads to Bayesian attacks [30].
Mutual information	Max.	Rely on off- or on-line *pdf* estimations [50; 84]. Models are also called "partitioning functions".
Least squares	Min.	Introduced in stochastic attacks [133]. Winning distinguisher for the 1st DPA contest v1 (by Ch. Clavier).
Variance	Min./Max.	Many references are available [62; 79; 82; 144].

Table 1.2: Classification of state-of-the-art attacks on cryptographic implementations.

Model characterization		**Attack's granularity**	
Offline	**Online**	**Monovariate**	**Multivariate**
No	**No**	DoM [72], DPA [20], CPA [24], DCA [74]	SCAN [38]
No	**Yes**	MIA [50], Stochastic attacks [133]	
Yes	**No**		Stochastic attack [145], Template attack [30]

and to their exploitation of the leakage. More precisely, two criterions can distinguish them:

- whether or not the attack requires a characterization (of parameters, as in the case of stochastic attacks, or of *pdf*, as in MIA or Template attacks, *etc.*) and
- whether or not the attack exploits the leakage from one or many time samples.

The table 1.2 presents the classification of most commonly used attacks with respect to those two criterions. It appears clearly that attacks without offline characterization are, in majority, exploiting the leakage from a single sample, whereas attacks with an offline characterization are all multivariate. Note that we include in the class of monovariate attacks those that:

- average several samples, as in [32], or
- change the trace representation, as the time $\leftrightarrow$ frequency transformation, described for instance in [90].

Indeed, those attacks actually perform a *multivariate pre-processing* and continue with a *monovariate exploitation phase*. The interpretation of this fact is the following:

- Attacks without offline characterization do not know the noise. Now it is possible to sort the samples of a trace by their signal-to-noise ratio (SNR). It is clear that the subset of samples that maximizes the SNR is the singleton containing only the sample where the SNR is maximal. It is thus favorable to improve the attack.
- At the opposite, attacks with offline profiling do know the noise, and can thus constructively take of advantage of the leakage of multiple samples.

1.5.2 SCA Algorithms: Typical Description

1.5.2.1 Basic Algorithm

In practice, it is difficult to model the signal leaked by a hardware implementation. The reason is that hardware implementations manipulate a large amount of data in parallel, but that we target only few bits of this data when performing the power analysis. Suppose that D power consumption traces are recorded while a cryptographic device is performing an encryption or a decryption operation. The evaluator chooses an intermediate result of the cryptographic algorithm. The intermediate value represents the binary value(s) of one or several bits at one or different leakages instants. Formally,

it can be seen as a deterministic function that takes two parameters in input. The first parameter, denoted by d, is known and can be either the plain text or the cipher text. The second parameter, denoted by sk, is secret, hence unknown. Indeed, sk is a small part of the cryptographic key and can take K possible values referred to as key hypotheses, denoted by k. In what follows, for sake of convenience, we denote the intermediate value by $v_{d,sk}$. Some physical leakage[1], $l_{d,sk}$, is generated when $v_{d,sk}$ is computed. In the literature of SCA, the leakage $l_{d,sk}$ is assumed to be composed of two terms: a deterministic term, $\phi(v_{d,sk})$, and an independent noise term ϵ_d. With these notations, the actual leakage $l_{d,sk}$ is written as follows:

$$l_{d,sk} = \phi(v_{d,sk}) + \epsilon_d \,. \tag{1.3}$$

Practically, a leakage model, which is based on a logical function h_{func}, enables the evaluator to compute a hypothetical intermediate value $h_{d,k} = h_{func}(v_{d,k})$ for every possible k key hypothesis and d. This way, the physical leakages are implicitly classified into several partitions, according to the hypothetical intermediate values computed for each key hypothesis. Eventually, the evaluator uses a statistical test, referred to as *distinguisher* Δ_k, to compare $h_{d,k}$ with $l_{d,sk}$. Formally, the evaluator builds a score vector $\Delta_{vect} = (\Delta_k)_{k=1}^{K}$. The key candidate k that is the most likely to be the right key hypothesis (*i.e.* the secret key sk) is the one which corresponds to the absolute maximum score, $\arg\max_k |\Delta k|$. Clearly, such key is related to the most appropriate partitioning of physical leakages.

Hereinafter, we give the analytical expressions of the usual Side-channel analyses that are principally based on the explained algorithm:

- **monobit-DPA** often called *Difference of Means (DoM)*, which basically involves two partitions as a single bit consumption activity is considered. The computation of DoM distinguisher is simple and can be expressed as:

$$DoM : \Delta_k = \mu_1 - \mu_2 \ , \tag{1.4}$$

  where μ_1 and μ_2 are the averaged traces of first partition and second partition, respectively. For false key hypotheses the partitioning is more or less random and the differential trace is flat. The secret key can be consequently identified as the one that yields the highest peak in the differential trace.

- **T-test** is an improved difference of means (DoM) test that takes the variance

[1] In SCA literature, physical leakages are often referred to as observations or measurements.

and the number of measurements of each partition into account as described for instance in [51]. The *T-test* distinguisher can be written as:

$$Ttest : \Delta_k = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{N_1} + \frac{\sigma_2^2}{N_2}}} , \tag{1.5}$$

where σ^2 and N are respectively the variance and the size of one partition.

- **multibit-DPA** or DPA for short is a generalization of DoM, as it considers the consumption activity of several bits in the implementation. DPA can be seen as a weighted version of the DoM. Indeed, it can be computed by attributing centered weights to considered partitions and calculating the sum over (centered) partitions. Otherwise, DPA can be reduced to a simple mathematical calculation thanks to *the Covariance (Cov)* operator. Hence, taken into account the notations provided previously, DPA can be written as:

$$DPA : \Delta_k = Cov(l_{d,sk}, h_{d,k}) . \tag{1.6}$$

- **Correlation Power Analysis (CPA)** is the normalisation version of DPA. The normalisation aims essentially at reducing the noise affecting the acquired traces. In fact, CPA is the actual *Pearson Correlation Coefficient*; and generally has the following form:

$$CPA : \Delta_k = \frac{Cov(l_{d,sk}, h_{d,k})}{\sigma_l . \sigma_h} . \tag{1.7}$$

where σ_l and σ_h are the standard deviations of obtained physical leakages and computed hypotheses, respectively.

- **Variance Power Analysis** aims at minimizing or maximizing a certain criterion based on the analysis of variance. From the statistical point of view, two options are provided: computing either the inter-class variance or the intra-class variance. For sake of clarity, we replace "class" by "partitions". More precisely, when several partitions are being involved, the secret key corresponds naturally to the lowest intra-partitions variance value partitioning, assuming that the consumption model used is correct. Therefore, it corresponds not only to the lowest intra-partitions variance, but to the highest inter-partitions variance, that can be used as a metric for quantifying the secret key's discriminatory power. Besides, we note that recent attacks based on weighted variance analysis (VPA) [83] have been developed to break protected cryptographic implementations.

- **Mutual Information Analysis (MIA)** is an information theoretic approach that has been presented to cryptographic community to measure the amount of information (linear or non linear) in the Side-channel leakages; and therefore extract the value of the secret key with more flexibility. Basically, the *mutual information*, measured in bits, is computed between the global observation O (*i.e.* the set of traces acquired) and the leakage L. L corresponds to leakages partitions involving the couple $(l_{d,sk}, h_{d,k})$ defined previously.

$$MI(O;L) = \sum_{o}\sum_{l} p(o,l)\log\frac{p(o,l)}{p(o)p(l)} \ , \tag{1.8}$$

$$MI(O;L) = H(O) - H(O|L) \ , \tag{1.9}$$

where l and o are realizations of L and O respectively, $H(O)$ is an estimation of the entropy of O, $p(o,l)$ is the joint probability density function of O and L, $p(o)$ is the marginal probability density function of O and $H(O|L)$ is the conditional entropy of O knowing L. $MI(O;L)$ can be regarded as a positive (*i.e.* $MI(O;L) \geq O$) and symmetric (*i.e.* $MI(O;L) = MI(L;O)$) measure of the strength of a 2-way interaction between two variables: the observation O and the leakage L. But more importantly, the higher the value of the mutual information, the higher the dependency between O and L. Statistically speaking, $MI(O;L) = 0$ if and only if O and L are independent random variables. In practice, it is hard to get an accurate estimation for the probability density function. Many methods have been proposed to estimate entropy like histograms, kernel density functions, Gaussian parametric estimators *etc.* [113]. In practice, the Gaussian parametric estimation, where the joint distribution of (O,L) is assumed to be Gaussian, can serve usually as a first approximation for the distributions' shape. In this case, entropy can be calculated as a function of standard deviation σ_o of O as:

$$H(O) = -\sum_{i} p(o_i)\log(p(o_i)) = \log(\sigma_o\sqrt{(2\pi e)}) \ .$$

Moreover, under the Gaussian assumption, it is easy to verify that mutual information is intimately connected to the Pearson coefficient ρ [120] and can be expressed as [119] follows:

$$MI(O;L) = -\frac{1}{2}\log(1-\rho^2_{0,L}) \ . \tag{1.10}$$

1.5.2.2 Template Attack Algorithm

Template attacks were introduced by Suresh Chari *et al.* in [30]. The salient feature of Template attacks is that it characterizes the noise in measurements, unlike other approaches. The main idea is to capture an amount n of traces $C_{M,k}(t)$ (typically $n > 1000$) on the programmable device for each subkey k and to describe the behaviour of the noise depending on k. Each set of $C_{M,k}(t)$ is averaged, to obtain a new set $\mathbf{A} = \{\mathbf{A}_k,\ \forall k \in K\}$. In order to reduce the profiling time, a set of point of interest has to be selected.

Let $\mathbf{T} = \{t_i,\ 1 \leq i \leq p\}$ be a set of p points of interest. For a given key k, we can now compute a noise vector for each traces $C_{M,k}(t)$ as follows:

$$\mathbf{N}_k(M) = [C_{M,k}(t_1) - \mathbf{A}_k(t_1), \ldots, C_{M,k}(t_p) - \mathbf{A}_k(t_p)] \ . \tag{1.11}$$

Let $\mathbf{N}_{k,t}$ be the vector of all elements of $\mathbf{N}_k$ at the instant t. Now, we can compute the covariance matrix which has its elements defined as:

$$\Theta_k[t_i, t_j] = Cov(\mathbf{N}_{k,t_i}, \mathbf{N}_{k,t_j}) \ . \tag{1.12}$$

The couple $(\mathbf{A}_k, \Theta_k)$ is the template for the key k. Profiling phase is finished when a template is computed for each key $k \in K$.

The key extracting phase uses the maximum likelihood principle. For each key k and for each measured traces, we compute a noise vector $\mathbf{n}$ on the points of interest (using $\mathbf{A}_k$). Thereafter we compute $f_k(\mathbf{n})$, where f_k is the multivariate Gaussian distribution, as follows:

$$f_k : \mathbb{R}^p \to \mathbb{R} \qquad f_k(\mathbf{n}) = \frac{1}{\sqrt{(2\pi)^p . |\Theta_k|}} e^{-\frac{1}{2}\mathbf{n}^{\mathrm{T}}\Theta_k^{-1}\mathbf{n}} \ , \tag{1.13}$$

where $|\Theta_k|$ is the determinant of Θ_k. $f_k(\mathbf{n})$ will give the highest value if k is the good guess. It gives the probability of each key candidate. Once the probability of each key candidate is known, we can compute the entropy and eventually mutual information.

1.5.2.3 Stochastic Model Attack Algorithm

Stochastic Models [133] are also a type of profiled attacks slightly. The profiling phase needs only one test key i.e. the power consumption is modeled, at a time t as follows:

$$W_t(x,k) = h_t(x,k) + \mathcal{B}_t \ , \tag{1.14}$$

where x is the plain text and k the key. The first summand h_t is the deterministic part of the power consumption (which depends on x and k) and $\mathcal{B}_t$ a random noise with zero expectation ($\forall t,\ \mathbb{E}(\mathcal{B}_t) = 0$). The first profiling step consists in approximating h_t, followed by estimation of $\mathcal{B}_t$ using h_t. h_t is assumed to have the *EIS* property (Equal Image under different Subkeys), which implies that only one test key is needed for the profiling phase. Let $\tilde{h}_t$ be the best estimation of h_t computed as:

$$h_t(x,k) = \beta_0 + \sum_{i=1}^{u} \beta_{it} g_i(x,k) \ , \tag{1.15}$$

where the g_i are chosen base functions, which depend on x and k, and β_{it} are coefficients, which estimates the system. It is the choice of base functions which define the degree of stochastic models. A linear model takes just a function of individual bits where as a higher degree model considers multiple bits for each coefficient. We assume that β_0 is always equal to 1. The second step of the profiling phase consists of characterization of the noise. First, some relevant instants have to be selected (*e.g.* by using the *T-test* [127] or *Euclidean norm* [127] of the coefficients β_{it}). The noise is characterized by constructing the probability density function of the multivariate normal distribution, using a covariance matrix (computed with a noise random variable associated on each point of interest). When the first device is profiled, attack can be performed using the maximum likelihood principle.

1.5.3 SCA Countermeasures

In the open litterature of SCA, many techniques, usually known as *countermeasures*, have been proposed to securing devices against Side-channel attacks. Principally, there exist two basic countermeasures that are very often deployed by designers to counteract SCAs: **information masking** [86, Chp. 9] and **information hiding** [86, Chp. 7]. First, information masking essentially aims at randomizing the Side-channel; and therefore masking the intermediate values that occur during the cryptographic process. Many masking schemes have been proposed for DES and AES [73; 88; 111]. Gen-

erally, they differ in term of hardware design complexity. However, it has been proved that masking countermeasure is still sensitive to first-order SCA as long as glitches problem remains not completely resolved [88]. Moreover, it has been shown that basic masked implementations, and even a full-fledged masked DES implementation using a ROM, are not resistant against new variants of SCA like High-order differential attacks (Ho-DPA) [94] or VPA [82]. Second, information hiding is a countermeasure that principally aims at balancing the Side-channel. The most commonly used technique for information hiding is called *Dual rail Precharge Logic (DPL)*. In fact, DPL aims at making the activity of the cryptographic process constant independently from the manipulated data. In the literature, existing DPL designs vary in term of performance and complexity. In [102], authors introduce different DPL styles (BCDL, WDDL, IWDDL, *etc.*) and make comparison between them. Generally, it is hard to perfectly implement such countermeasure in practice.

1.5.4 SCA Metrics

1.5.4.1 Attack's Efficiency Metrics

In the security field, it is assumed that the level of robustness of secure devices can be measured and deduced through attacks while the secret key is known. This is true as such analyses are worthy in that they pinpoint the vulnerabilities that the secure product is designed to resist. In the SCA literature, the first used evaluation metric, called *stability criteria* [143], consists in determining the number of Side-channel measurements acquired to guess the secret information: one key is supposed to be correctly guessed if a stability criteria is achieved (*i.e* the Side-channel analysis has to continuously return the correct key when accumulating the traces). For instance, let C be a secure device under test, A and B are two SCAs and S the fixed stability. Suppose that B needs more traces than A to achieve the stability S. Therefore, the trivial deduction is that the device C is more vulnerable to A than to B. This kind of deduction would give more details about the statistical nature of the vulnerability. Such metric is useful especially when the evaluator is not free to acquire as much Side-channel traces as he wants. Indeed, in such case, he has to perform the analysis once and for all on the totality of traces that he may acquire and wait until the secret key stabilises. Recently, two independent evaluation metrics [146] have been proposed by F.X. Standaert to assess the performance of different analyses: the *First-Order Success Rate* and the *Guessing Entropy*. Both metrics measure the extent to which an adversary is efficient in turning the Side-channel leakage into a key recovery. On

the one hand, the First-order success rate expresses the probability that, given a pool of traces, the attack's best guess is the correct key. On the other hand, the Guessing entropy measures the position of the correct key in a list of key hypotheses ranked by a distinguisher. Such metrics are very useful when the number of Side-channel traces that can be acquired is unlimited. The two metrics are independent [146], but both quantify the efficiency of one attack in term of rapidity and stability when retrieving the correct key (or secret key). Indeed, they just analyse differently the behaviour of the secret key with regards to false key hypotheses; which might be helpful to the security evaluation perspective. In practice, they are represented by a 2D-diagram which axis is the number of traces required for a successful attack. Fig. 1.1 and Fig. 1.2 depict, respectively, a simulation of the First-order success rate and the Guessing entropy metrics for different Side-channel attacks. Hereinafter, the attacks names are not revealed, as we just want to give a general description for both metrics. According to Fig. 1.1, three attacks are involved *attack1*, *attack2* and *attack3*. Obviously, *attack3* is the most powerful (in term of rapidity). For instance, it needs around 175 Side-channel traces to reach a success rate of 80%. Whereas, *attack1* and *attack2*, which are slightly different, need around 250 traces for the same success rate. Fig. 1.2 shows the Guessing entropy metric for two new attacks (*attack4* and *attack5*). We say that *Attack4* globally outperforms *attack5*. For example, for 100 traces needed, the average rank of the secret key is 5 for *attack4*; whereas it is 10 for *attack5*. Generally (but not always [51]), when comparing the efficiency of two attacks, it is often shown that results given by the Success rate and the Guessing entropy metrics are not contradictory.

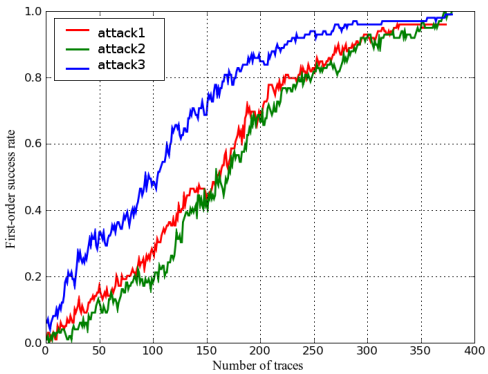


Figure 1.1: First-order success rate for three different attacks.

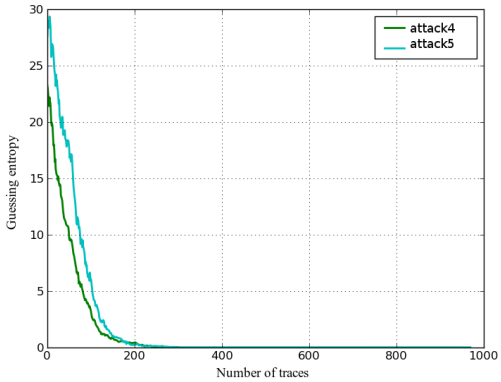


Figure 1.2: Guessing entropy for two different attacks.

1.5.4.2 Leakage Quantification Metrics

Information theoretic approach

This approach is used to measure the amount of the useful information, which is leaked from the device under test. Technically speaking, this metric is mainly based on the computation of the *conditional entropy*, briefly invoked in 1.5.2.1. Using the same notation in 1.5.2.1, *the conditional entropy* can be written as:

$$H(O|L) = \mathbb{E}_o \cdot \mathbb{E}_{l|o} - \log p(O = o|L = l) \ ,$$

where $\mathbb{E}$ is the expectation operator. Basically, the higher the value of H, the more robust the implementation.

Hypothesis testing approach

This approach, which is initially proposed by S. Mangard in [85], involves Signal-to-Noise Ratio (SNR) notion and Fisher transformations [127]. Basically, it aims at estimating the correlation coefficients that occur in first-order SCA, in particular CPA, without actually performing the attack in practice. In other words, computations made with the correct key are sufficient to apply this approach. But more importantly, these estimations of the correlation coefficients, denoted by ρ_{estim}, are used in a mathematical formula to predict the number of Side-channel traces needed to perform a successful CPA (*i.e* extracting the value of the secret key among all key hypotheses). Formally, this formula is expressed as follows:

$$N = 3 + 8 \left(\frac{Z_{1-\alpha}}{\ln\left(\frac{1+\rho_{estim}}{1-\rho_{estim}}\right)} \right)^2 \ , \tag{1.16}$$

where N is the predicted number of traces, α is a the confidence interval which is used to indicate the accuracy of an estimate and Z_α is a quantile of a normal distribution for the 2-sided confidence interval with error *1-*α. For more details about this formula, we refer the reader to [86].

Chapter 2

Side-Channel Attacks

2.1 Our Contributions

In this Chapter, we propose new Side-channel distinguishers and attack algorithms that attempt to provide both **genericity** and **efficiency**. From the evaluation point of view, genericity is a crucial criterion that allows the evaluator to reduce the cost of the analysis before the multitude of existing attacks in the litterature.

First, in Sec. 2.2, we answer the question of what are the necessary conditions under which CPA is optimal with regards to attacks that exploit the same leakage model. For this purpose, we offer an in-depth theoretical study which aims at determining the conditions under which the Pearson Correlation Coefficient is maximized: we recall the fundamental principles of this coefficient and provide the mathematical background necessary to allow us to make concrete statements later on. This is a required contribution towards putting the CPA on a sound theoretical basis. From a theoretical point of view, the mathematical approach that we present is different relatively to previous works [25; 85]. Indeed, it can be seen as a complementary study for these studies. Summarizing, in this Section we provide answers about the common problem of quantifying the efficiency of the Correlation Power Analysis. Besides, we note that this study is necessary to understand the basics of the next Section. 2.3.

Second, in Sec. 2.3, we put forward new methodologies, based on the combination of most commonly known SCA distinguishers, in order to accelerate the key recovery, in a **generic** manner. Precisely, we provide a theoretical method and an empirical approach to combine Pearson and Spearman correlation coefficients. We show that such combination leads to a more powerful attack. The empirical approach that we provide is essentially dependent on the practical behaviour of distinguishers. For this purpose, in the next Section 2.4, we propose to give a special attention to such behaviour, by

analysing the rank evolution of key hypotheses.

Third, in Sec. 2.4, we present the *Rank Corrector (RC)*, an empirical approach aiming at enhancing most Side Channel Attacks. We show that during an SCA on symmetric encryptions, the rank of the secret key displays a specific behaviour with regards to other hypotheses. Hence the Rank Corrector algorithm is devised, in order to improve existing SCAs by exploiting such behaviours. With a profiling phase on a clone device, we precisely evaluate the set of parameters that ensure the adaptability of RC to a large range of cryptographic systems, and the possibility to discriminate the secret key from other hypotheses in an efficient manner. The main principle of RC is to detect and discard the false keys hypotheses when analysing the ranking evolution. This results in improving the rank of the secret key, thus accelerating the attack. The efficiency of our algorithm is assessed by performing a DPA with and without the Rank Corrector. We show a significant gain compared to basic attack.

Fourth, in Sec. 2.5, we introduce *First Principal Components Analysis (FPCA)*, which consists in evaluating the relevance of a partitioning using the projection on the first principal directions as a distinguisher. Indeed, FPCA is a novel application of the Principal Component Analysis (PCA). In SCA like Template attacks, PCA has been previously used as a pre-processing tool. The originality of FPCA is to use PCA no more as a pre-processing tool but as a genuine distinguisher. We show that FPCA is more performant than first-order SCA (such as DoM, DPA or CPA) when performed on an unprotected DES architecture. Moreover, we outline that FPCA is still efficient on masked DES implementation, and show how it outperforms VPA, which is a known successful attack on such countermeasures.

Eventually, in Sec. 2.6, we propose to use *Wavelet transforms*, that are initially used to pre-process Side-channel traces, in the very core of the attack. We show that SCAs when performed with such multi-resolution analysis are more efficient, in term of security metrics, than considering only the time or the frequency resolution. Actually, through our experiments, we show that the gain in number of traces needed to recover the secret key is about 50%, relatively to an ordinary attack. For this purpose, two attacks are considered: CPA and Principal Components Analysis (PCA) based Template attacks. Besides, we note that such analysis is **generic** as it can be seen as a plug-in used to improve existing Side-channel attacks. Second, the overall goal of this section is about valuing, in a methodological manner, the use of Wavelet transforms to mount an efficient SCA.

2.2 On the Optimality of Correlation Power Analysis

2.2.1 Introduction

Recently, E.Prouff *et al.* have shown in [40] that Side-channel distinguishers are not only asymptotically equivalent but also can be rewritten one in function of the other, only by modifying the power consumption model. In particular, they have established an equivalence between most univariate Side-channel distinguishers and CPA performed with different leakage models. Besides, based on the same correlation concept (*i.e.* Pearson coefficient), it is shown that it is possible to break protected implementations (masking countermeasure) by considering the leakage at different time samples. Such attacks, called *Higher-Order Power Correlations*, were suggested and investigated by T.Messerges in [94]. Additionally, the CPA has inspired new disciplines that are essentially based on the analysis of power consumption leaked from embedded systems. For instance, CPA has been recently proposed as innovative technique for watermarks detection mainly used for Intellectual Properties (IP) protection [47]. Obviously, CPA continues to outsmart its competitors, and one can fear the excesses of this powerful tool in the context of Side-channel analysis. In this part of Chapter, we answer the question of what are the optimal conditions under which CPA is optimal with regards to attacks that exploit the same leakage model. For this purpose, we provide a comprehensive study about Pearson correlation coefficient: we recall the fundamental principles of this coefficient and provide the mathematical background necessary to allow us to make concrete statements later on. In this study, the theoretical approach that we put forward can be seen as a new paradigm (and not an analytical demonstration) based on *Estimation theory*; to study the optimality of the CPA. For more in-depth study about Estimation theory, we refer the reader to [28; 68; 92]. The overall goal of this study is to put the Correlation Power Analysis on a sound theoretical basis; and therefore brighten the task of the evaluator when using CPA in his analysis.

2.2.2 Notations & Definitions

Let X be a random variable with probability density function (*pdf*) $P_X(x)$. X is said to be Gaussian random variable (or normally distributed variable) with mean μ_X and standard deviation σ_X if it is drawn from a normal distribution, denoted by $\mathcal{N}(\mu_x, \sigma_x^2)$.

The *pdf* of a Gaussian variable X is given by:

$$P_X(x) = \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu_x}{\sigma_x}\right)^2} . \tag{2.1}$$

If two variables X and Y are **Gaussians** and **independent**, then they are said to be *jointly Gaussian* variables.

Definition: Jointly Gaussian
A collection of n random variables $X_1, X_2, \ldots, X_n$ are jointly Gaussian if $\sum_{i=1}^{n}(a_i X_i)$ is a Gaussian random variable $\forall$ real a_i, with $i \in [1..n]$.

But we note that, jointly Gaussian distributed variables need not to be independent. If X and Y are jointly Gaussian then the pair of variables[1] (X, Y) must be drawn from *the bivariate normal distribution* defined by the following probabilty density function:

$$P_{X,Y}(x, y) = \frac{1}{\sigma_x\sigma_y 2\pi\sqrt{1-\rho_{X,Y}^2}} e^{\frac{-({z_x}^2+{z_y}^2-2\rho_{X,Y}z_x z_y)}{2(1-\rho_{X,Y}^2)}} , \tag{2.2}$$

where $z_x = \frac{x-\mu_x}{\sigma_x}$ and $\rho_{X,Y}$ is the Pearson correlation coefficient between X and Y.

2.2.3 The Optimality From the Historical View Point

In 1885, Francis Galton [45], the man responsible for the correlation coefficient ρ defined the term *regression* and determined the mathematical formula of the bivariate normal distribution [108]. This fill out the studies carried out by Gauss and Bravais half century before [121]. Authors, in [121], note that Galton defined ρ as a metric to measure the strength of the association (correlation), and suggested that this metric is lesser than 1. Few years later, Pearson developed the mathematical formula, called Pearson's product moment correlation coefficient. The Pearson's product moment correlation coefficient between two random variables X and Y with expected values μ_x, μ_y and standard deviation σ_x, σ_y, respectively, was defined as:

$$\rho_{X,Y} = \frac{\mathbb{E}[(X-\mu_x).(Y-\mu_y)]}{\sigma_x.\sigma_y} , \tag{2.3}$$

[1] For the misuse of language, the pair (X, Y) is also called *Gaussian couple*.

where $\mathbb{E}$ is the Expected value operator. ρ is a dimensionless index, which is invariant to linear transformations of either variables. Moreover, it satisfies the following inequality:

$$|\rho_{X,Y}| \leq 1 \ . \tag{2.4}$$

When developing the mathematical formula of Pearson correlation, an important assumption has been made as for the joint distribution of variables. Indeed, this assumption states that the joint distribution of analyzed variables is bivariate normal [108]. Moreover, as initially stated in Galton and Pearson papers, ρ provides the best estimate of the linear association between two variables only when the joint distribution of the variables is Gaussian. Besides, most techniques related hypothesis testing or constructing confidence intervals of correlation coefficients require the assumption that the joint distribution of (X, Y) is bivariate normal. According to the mathematician Caroll (1962), the most common mistake that could occur in the bivariate study is to start by calculating the correlation then make (false) statements about the relation[1] between the variables without taken seriously the Gaussian assumption.

2.2.4 The Optimality From the Estimation Theory View Point

2.2.4.1 The Approximation Problem

Suppose we want to best approximate Y with another variable X based on their joint distribution. The approximation problem is to seek for a function $\phi(.)$ of X that best fits Y among all possible forms of $\phi(.)$. We write $\widehat{Y} = \phi(X)$ and we call $\widehat{Y}$ an estimator of Y. In our study, the variable X is deterministic since it is theoretically predicted from a known cryptographic process. Whereas, the variable Y is a real measure acquired by an oscilloscope. For sake of clarity, in what follows the variable X is called *the prediction* and Y *the measurement* (or the observation). Let $\epsilon = Y - \widehat{Y}$ denotes the error in estimating Y, and let $pos(\epsilon) = pos(Y, \widehat{Y})$ denotes a non negative function of ϵ. $pos(\epsilon)$ can be for instance the absolute difference or the square difference between Y and $\widehat{Y}$ (*i.e.* $|Y - \widehat{Y}|$ or $(Y - \widehat{Y})^2$ respectively). The average cost, *i.e*, $\mathbb{E}[pos(Y, \widehat{Y})]$, is referred to as the *Bayes risk* $\Re_B$. Obviously, the approximation problem comes down to a minimization problem. In fact, minimizing the *Bayes risk* with respect to $\widehat{Y}$ for a given cost function is a proper solution of the problem. The most popular $\Re_B$ is the *Mean Square Error (MSE)*, since it is parameter free, straightforward to implement and memory-less. The MSE measures the average of the squares of the errors. In this case,

[1] or dependency.

it is clear that $pos(Y, \widehat{Y}) = (Y - \widehat{Y})^2$. In what follows, we will focus on the important role played by the MSE in the approximation problem. There are several ways in which the role of the MSE can be introduced. A particular way for especial convenience is to work with the L^2 space that is defined as the space of square summable variables[1]. If Z is a random variable belonging to this space, then the corresponding norm is given by:

$$\|Z\|_2 = \sqrt{\mathbb{E}[Z^2]} \ , \tag{2.5}$$

so that the distance between two elements Z_1 and Z_2 of L^2 space is

$$\|Z_1 - Z_2\|_2 = \sqrt{\mathbb{E}[(Z_1 - Z_2)^2]} \ . \tag{2.6}$$

Z_1 and Z_2 are said to be *orthogonal* ($Z_1 \perp Z_2$) if and only if $\mathbb{E}[(Z_1 Z_2)] = 0$. Moreover, the norm $\|.\|_2$ is often called the L_2 norm, and the corresponding notion of convergence is of course convergence in mean square

$$\|Z_n - Z\|_2 \longrightarrow 0 \Longleftrightarrow Z_n \longrightarrow_{m.s} Z \ . \tag{2.7}$$

Orthogonality property and mean square convergence will allow us in the following to introduce the notion of optimal estimation in the sense of L_2 norm. With these notations, the optimal estimator of Y given X, in the sense of the L_2 norm, is the function $\widehat{Y} = \phi(X)$ for which $\|Y - \widehat{Y}\|_2^2$ is a minimum [68]. But more importantly, it is proved that the conditional expectation $\widehat{Y} = \mathbb{E}[Y|X]$ is the estimator that gives such a minimum. Besides, using the error notation, ϵ, the MSE is written in the following form:

$$MSE(\widehat{Y}) = \mathbb{E}[\epsilon^2] = \|Y - \widehat{Y}\|_2^2 \ . \tag{2.8}$$

Besides, in [11], it is shown that MSE can be expressed as follows:

$$MSE(\widehat{Y}) = Var(\widehat{Y}) + bias(\widehat{Y})^2 \ , \tag{2.9}$$

where $Var(\widehat{Y})$ is the variance of $\widehat{Y}$ and $bias(\widehat{Y}) = \mathbb{E}(\widehat{Y}) - Y$. Note that for an unbiased estimator ($i.e\ bias = 0$), the MSE is just the variance of the estimator. In the litterature of estimation theory [43], two naturally desirable properties of estimators are for them to have minimal MSE and to be unbiased. Common criterions for estimation are Maximum

[1] The L^2 space is often referred to as a weighted Euclidean norm.

Likelihood (MLE), Minimum Mean Squared Error (MMSE) and Maximum A Posteriori Probability (MAP)([68]). From the theoretical point of view, MLE approach is more efficient than the rest of criterions. But more importantly, estimation theory says that no asymptotically unbiased estimator has lower MSE than the MLE (see *Cramer-Rao Lower Bound theory*) [18; 22; 141]. However, in practice, statisticians prefer using MMSE estimator, specifically in the linear case, which is in fact the approach that minimizes the MSE in the sense of the L_2 norm, because of its simplicity relatively to the other criterions. Additionally, later on, we will show that, under few assumptions, MMSE estimator produces the lowest MSE among all estimators, in particular unbiased ones, and can be derived as a maximum likelihood estimator.

2.2.4.2 Optimal Linear MMSE Estimation & Connection with ρ

As stated before, the conditional expectation is the optimal estimator in the sense of the L_2 norm, which is indeed the MMSE estimator. Hence, the MSE can be rewritten as follows:

$$MSE(\widehat{Y}) = \mathbb{E}[\epsilon^2] = \|Y - \mathbb{E}[Y|X]\|_2^2 \ . \tag{2.10}$$

A useful property of the MMSE estimator is that the estimation error $Y - \mathbb{E}[Y|X]$ is orthogonal to every function of the variable X. This property is known as the *Orthogonality Principle*. But more importantly, this principle provides a necessary and sufficient condition for the optimal estimation in the L^2 space. More formally, $\phi(X)$ is the MMSE estimator $\hat{Y}_{MMSE}$ if and only if the error $Y - \phi(X)$ is orthogonal to every function $\gamma(X)$ that is

$$\mathbb{E}\left[(Y - \phi(X))\gamma(X)\right] = 0 \ . \tag{2.11}$$

Now, the problem is that MMSE is very general; and therefore, the conditional expectation can be complicated to compute. Nonetheless, the analysis is very simple when the *linear assumption* is made (*i.e.* Linear MMSE, often termed by LMMSE). For this purpose, statisticians usually make such assumption as a first approximation. However, when the true data does not fit the linear case, we say that LMMSE is sub-optimal to the optimal estimate of MMSE. In the context of Side-channel analysis, the linear case has a pure theoretical flavour for us especially when considering linear leakage models, related basically to unprotected implementations. But it is noteworthy that even for unprotected implementations it is possible to have recourse to what we call *linear*

transformations [135]; and therefore to fall into the linear case. In "Introduction to optimal estimation" book ([68] Chapter 3), using the orthogonality principle (Eqn. 2.11), authors show that when the true data fit exactly the linear case (*i.e.* optimal LMMSE) the associated MSE of $\hat{Y}_{LMMSE}$ is expressed with Pearson coefficient ρ, as follows:

$$MSE_{LMMSE} = {\sigma_Y}^2(1 - \rho^2_{X,Y}) \ . \tag{2.12}$$

In the linear case, $\hat{Y}_{LMMSE}$ is the optimal estimate in the sense of MMSE estimation. But more importantly and always from the MMSE estimation point of view, it is clear that ρ is the optimal metric to measuring the linear association between involved variables. Actually, the **maximization** of ρ^2 implies the **minimization** of MSE_{LMMSE}.

Limitations of Optimal MMSE Estimation

Up to this point, we have shown that in the linear case Pearson correlation coefficient is an optimal indicator of linearity in the sense of MMSE estimation. A simple graphical illustration of the linear case is depicted in Fig. 2.1. In this figure, clearly the values taken by the measurement Y are linearly increasing with the values taken by the prediction X. However, the MMSE does not make any assumptions about the joint distribution; which contradicts the optimality of Pearson coefficient from the historical point of view. One may ask, is there any connection between the estimation theory and what is assumed by Pearson and Galton ? The Pearson Correlation does he still the best linear indicator even if the joint distribution is not Gaussian ? Indeed, the fact that the MMSE is distribution free[1]is often seen as a weak point in the estimation litterature, specifically when performing a linear estimation (LMMSE). Generally, when no assumptions are made about the joint distribution, there exist two important cases in which the optimality of LMMSE, relatively to all estimators, is not guaranteed. In other words, in these cases LMMSE does not give the lowest MSE among the other estimators such as the Maximum Likelihood (MLE).

Case 1: Heteroscedasticity This basically occurs when the error of estimation ϵ depends on the prediction X. The LMMSE only states that the error of estimation ϵ is uncorrelated with the prediction X. In the linear case, this statement follows since

$$Cov(X, \epsilon) \ = Cov[X, (\mathbb{E}[Y|X] - Y)] \ ,$$

[1]In statistics, a statistical criterion that does not make any assumption about the joint distribution is said to be distribution free.

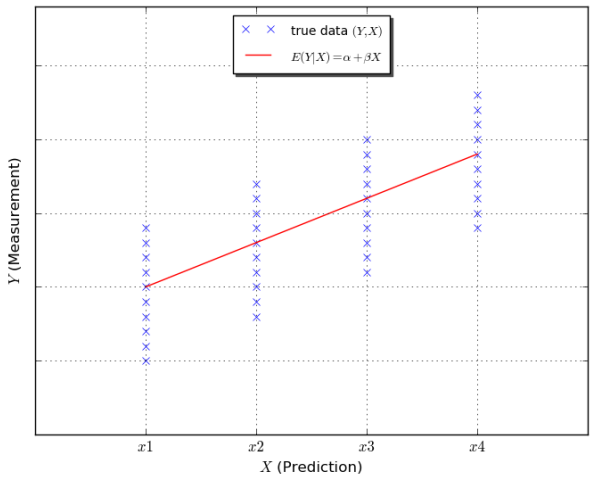


Figure 2.1: An example of a linear relationship between X and Y.

$$
\begin{aligned}
&= Cov[X, (\alpha + \beta X) - Y] \ , \\
&= -Cov(X, Y) + Cov(X, \alpha) + \beta Cov(X, X) \ , \\
&= -Cov(X, Y) + 0 + \beta Var(X) \ , \\
&= -Cov(X, Y) + \frac{Cov(X,Y)}{Var(X)} Var(X) = 0 \ .
\end{aligned}
$$

However, $Cov(X, \epsilon) = 0$ does not imply the independence of X and ϵ. In other words, even if the linear estimation is optimal in the sense L_2 norm ($i.e$ $\mathbb{E}[Y|X] = \alpha + \beta X$), it could exist a relation between X and ϵ which compromises the efficiency of the LMMSE in estimating the parameters α and β of the linear model. In this case, the linear model is said to display a *heteroscedasiticity*. A frequent situation of Heteroscedasticity is that the error is linearly increasing with the values taken by the prediction X. For such situation that is illustrated in Fig. 2.2, it is easy to verify that the MLE estimator is more efficient than the LMMSE as it produces the lowest MSE([127] page 398).

Case 2: Imperfect data The data, which is composed by the prediction X and the measurement Y, is often disturbed by the presence of what we call outliers. An *outlier* can vaguely be defined as an observation which shows a different behaviour with regards to observations composing the data. The reason might be due to the type of variables (continuous, discrete) and the shape of the marginal distributions, $P_X(x)$ and $P_Y(y)$. Actually, it is often reported that a frequent cause of outliers is a mixture of two distributions. Moreover, by contrast to the de-

terministic nature of the prediction X, the measurement Y is usually dependent on the acquisition environment which is a source of undesirable effects like the noise. Therefore, the measurement error might be a second reason for such a phenomena. As it is shown in Fig. 2.3, the outliers are clearly those points that lie far from the line describing the true relationship (called least square line).

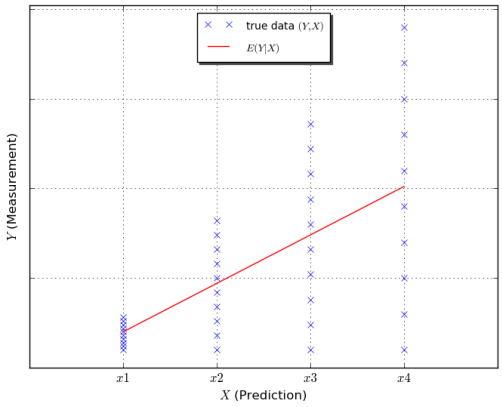


Figure 2.2: An illustration of Heteroscedasticity problem.

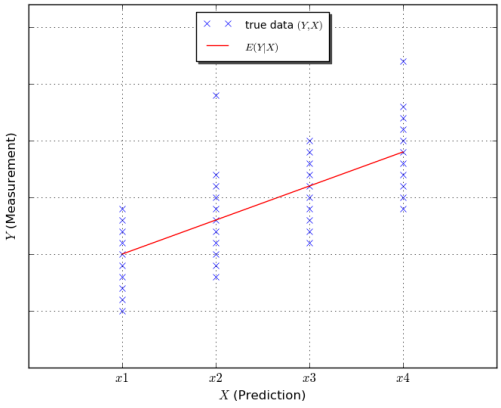


Figure 2.3: An illustration of outliers problem.

Overall Optimality of MMSE

According to **case 1** and **case 2**, the MMSE is not enough to totally characterise the dependence between X and Y, even if the true relationship between them is **linear**. More importantly, the Pearson Correlation Coefficient could not be considered as the best linear metric to measuring the true relationship. In statistic, there exist several candidates, such as Spearman, Kendall or intra-class coefficient correlations, that are designed to be less sensitive (more robust) to outliers or Heteroscedasticity and therefore they would be better than Pearson Coefficient. However, the estimation theory proved that there exists one and only one condition when satisfied then the MMSE is equivalent to the MLE; and therefore considered to be the optimal estimator among all estimators, in particular unbiased ones, as it gives the lowest MSE. Thus, the Pearson coefficient ρ is the best metric for measuring a linear association. This condition requires that the joint distribution $P_{X,Y}(x, y)$ should be bivariate normal (Gaussian) [68; 112]. In fact, under the condition of Gaussianity, the true relationship is **linear**, not heteroscedastic (*i.e* homoscedastic) and not disturbed by some undesirable effects like the outliers.

Note that in this case the error of estimation follows a normal distribution. Hence, we can state that the Pearson Coefficient Correlation CPA is the best linear metric only when the true relationship in the MMSE sense satisfies the Gaussian condition (or Gaussianity), which is in agreement with the historical point of view. If Gaussianity is not validated the MMSE is less efficient than MLE; and therefore the optimality of CPA is compromised.

Sufficient and practical Conditions for the Optimality

A common error about the validity of the Gaussian assumption is to check only that X and Y are drawn from normal distributions. This is not sufficient. Indeed, if X and Y are each individually Gaussian then this does not imply that they are jointly Gaussian. Generally, a joint distribution is said to be bivariate normal if all following conditions are satisfied ([15] page 54):

- **Linearity** The true relationship between X and Y is linear.
- **Normal conditional distribution** The conditional distribution of Y given X=x is normal.
- **Homoscedasticity** The conditional distribution of Y given X=x has a constant variance (*i.e.* the variance of the error) for each x.
- **Normal marginal distribution** The marginal distribution of X is normal (Gaussian).

Moreover, under these conditions, the error ϵ must be constant and drawn from zero mean normal distribution. In other words, ϵ is a random variable strictly independent from X and that a linear function ϕ characterizes the dependence between X and Y, entirely. In practice , these conditions are not supposed to be strictly verified but to hold to a certain degree. Actually, in real situations, it is mostly hard to get a perfect binormal joint distribution. In such situations, the higher the departure from the Gaussian assumption is, the lower the efficiency of Pearson correlation coefficient ρ will be.

2.2.5 Case Study

In this study, we are interested in the basic attack of DES, that targets the first rounds of the algorithm. Let p_l be a plain message, SK be the 48-*bit* round key used for the first DES round and f_r be the round function computed during each DES round. Precisely,

we focus on the activity of the right-hand side 32-*bit* R register at the first DES round and assume that the register's power consumption is proportional to the amount of transitions occurring in it. The transition *activity* at the R register output can be expressed by the Hamming Distance of the two consecutive values when switching to the first round:

$$activity = HD\left(p_l, f_r\left(p_l \oplus sk\right)\right) = HW\left[p_l \oplus f_r\left(p_l \oplus sk\right)\right] \ , \tag{2.13}$$

where HW and HD are the Hamming weight and Hamming distance, respectively. Here, sk is a 6-*bit* value, that can be recovered by predicting the activity of the four output bits of the corresponding Sbox. Recall that the DES implementation is composed of eight different Sboxes, there are thus eight secret keys to retrieve $SK_{DES} = [sk_i]_{i=0}^{7}$[1]. Therefore, in this case, the HD model can take five possible values, $HD = \{0, 1, 2, 3, 4\}$, and 2^6 key hypotheses are required to break one Sbox. Now, let us analyse the marginal distributions of the measurement Y and the prediction X. Firstly, the variable X, that is represented by the values taken by HD, is a discrete type variable following a symmetric distribution [54] $\beta_{(n_b=4, p=\frac{1}{2})}$ where n_b represents the predicted bits in the targeted register R, and p is the success probability. In statistics, if n_b is large, say $n_b > 20$, and $p = \frac{1}{2}$, then the binomial distribution is approximately equal to the normal distribution [122; 123; 127]. In Fig. 2.4, we simulate three different binomial distributions $\beta_{(4,\frac{1}{2})}$, $\beta_{(4,\frac{2}{3})}$ and $\beta_{(35,\frac{1}{2})}$, denoted by β_1, β_2 and β_3, respectively.

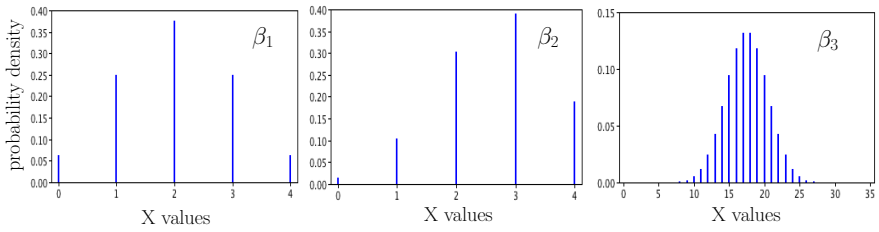


Figure 2.4: Examples of Binomial distributions.

Only β_3 can be approximated to a normal distribution. Besides, β_1 does not fits our case since $n_b << 20$. Hence X can not be strictly approximated to a normal distribution. Secondly, it can be shown that the shape of the marginal distribution of the measurement variable Y is dependent on the variation of the independent noise term, ϵ. In practice, a noise version of the marginal distribution of Y can be simulated by adding a Gaussian noise to the deterministic term. Assuming that ϵ is Gaussian, an empirical

[1]For misuse of language, we often use the notion of "broken Sbox" to say that the secret key corresponding to the attacked Sbox is found.

assumption is just an approximation of the real noise term distribution. Indeed, there exists different types of noise but their distributions can be usually approximated to a normal distribution. The assumption that the noise term is Gaussian does not imply that the measurement Y is Gaussian too. In fact, this basically depends on the amount of the noise variance σ^2_{noise}. This can be intuitively illustrated through the example of Fig. 2.5.

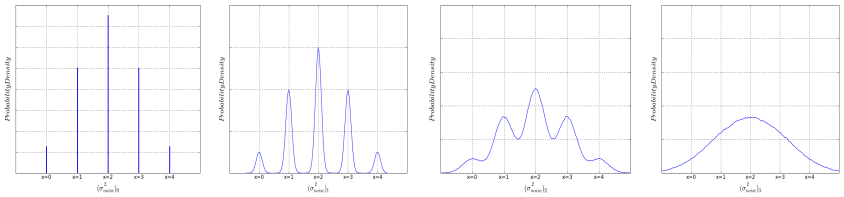


Figure 2.5: Examples of Binomial distributions with additive Gaussian noise.

The left image of the figure shows the binomial distribution that corresponds to the HD leakage model (i.e the marginal distribution of X) with zero noise variance, $\sigma^2_{noise} = 0$. The rest of the images show the same leakage model with some increasing additive noise ($0 < (\sigma^2_{noise})_1 < (\sigma^2_{noise})_2 < (\sigma^2_{noise})_3$). Obviously, the shape of the leakage distribution is converging towards the normal distribution, when the noise variance is getting higher. Moreover, a low σ^2_{noise} leads to a Gaussian mixtures, which does not fulfill the bivariate Gaussian assumption. Thus, in this case, ρ might lose its efficiency when measuring the true relationship between X and Y. However, adding noise could affect considerably the quality of ρ. Hence, a large number of traces is needed to succeed the attack (see Appendix B.1). Eventually, the analysis of the marginal distributions of both, the prediction X and the measurement Y, allows to conclude that their joint distribution is not perfectly Gaussian. In fact, there is some deviation from the bivariate normal assumption; and therefore the Pearson correlation coefficient ρ might not be efficient.

2.2.6 Conclusion

In this part of Chapter, we have studied the efficiency of Correlation Power Analysis (CPA) from the estimation theory point of view. This study is useful in that it allows the evaluator to assess the performance of CPA; and therefore to decide on the choice of an appropriate Side-channel distinguisher for his analysis. Actually, if the Gaussian case conditions are satisfied, then CPA must be the best analysis to quantify the sensitive information (real leakage). Besides, if these conditions do not hold to a certain degree,

then CPA might not be efficient and therefore is not the best analysis anymore. In this case, the evaluator is required to investigate more powerful techniques. For this purpose, in the next Section 2.3, we propose a new Side-channel distinguisher based on the robust[1] Gini correlation, and show its efficiency especially when the Gaussian case is not satisfied.

[1] A statistical criterion that does not make any assumption about the joint distribution is said to be robust or distribution free.

2.3 Combined Side-Channel Distinguishers

2.3.1 Introduction

In the open litterature of Side-channel attacks, very few papers have tried to devise combined methods to improve existing attacks. In [132], a combination of timing and power attack is used to attack RSA. Authors in [3], propose two different kinds of combined attacks. In first case, they use the same set of traces but partition them using two different leakage models: a 4-*bit* model and a mono-bit model. A third model built combing these two models is used for the attack, which turns out to be more efficient. In the second method, some relevant time samples are localised and combined. Authors take a product of the correlation coefficient at relevant time samples. Both these methods result in a faster convergence towards Success rate of 100%. Another related work is [142] where authors tend to concatenate and combine electromagnetic (Em) and power traces. The problem of this work was that Principle Component Analysis (PCA) was applied to the traces without normalization. If the variance of two campaign is not the same, PCA will favor the one with higher variance.

In this part of Chapter, we take advantage of the previous study (optimality of CPA 2.2) to put forward new methodologies, based on the combination of most commonly known SCA distinguishers, in order to accelerate the key recovery, in a **generic** manner.

2.3.2 Combination of Distinguishers

Up to this point, we have seen that the choice of a good distinguisher is essential for a successful SCA. We propose to combine different distinguishers to accelerate the attack. The methodology can be used to combine can take different types and number of SCA distinguishers depending on the application. We demonstrate our methodology by combining Pearson and Spearman correlation coefficients. Nevertheless, more than two distinguishers can also be combined. We combine Pearson and Spearman correlation because our experiments are based on traces acquired from DES implementation running on FPGA and these two distinguishers are commonly used for attacking FPGA. Combination can be done by two methods. Some complex correlation coefficient exist which tend to combine advantages of other distinguisher. This can be seen as a theo-

retical combination. On the other hand, some practical methods can be applied to use results of different distinguishers which increases the signal to noise ratio. Under some conditions, we show that the combination of these coefficients is possible and leads to a more powerful SCA.

Previously, when we dealt with the optimality of Pearson correlation, we have seen that under the Gaussian assumption, ρ is theoretically, precisely from Estimation theory perspective, the best tool to totally characterize the linear association between two random variables X and Y [36; 147]. However, in real situations, it is mostly hard to get a perfect binormal joint distribution. In such situations, the higher the deviation from the Gaussian assumption is, the lower the efficiency of ρ is. In this insight, other correlation coefficients have been developed to be more robust than the Pearson correlation, or more sensitive to nonlinear relationships. Among these correlations the Spearman's *(r)* and Kendall's tau (r_τ), Total correlation, Biserial, Tetrachoric are also used [99]. Spearman correlation measures both the linear and the non-linear relationship between the two variables, as it does not require that the observations be drawn from a Gaussian distribution. It is a *non-parametric* coefficient which was first applied in side channel context in [19]. In the literature of correlation analysis, it is known that provided the deviation from Gaussianity is not excessive, there is no rule to determine whether ρ will outperform its competitors. In this insight, statisticians have recently started to investigate actual combinations between existing correlation coefficients, which bridge the gap between Pearson coefficient and its competitors.

2.3.2.1 Gini Correlation: A mixture of Pearson and Spearman Coefficients

Basically, modern statisticians concur on two facts: On one hand, Pearson correlation, ρ, might perform poorly when the data is attenuated by non-linear transformations, in contrast to Spearman correlation, r. On the other hand, r is not as efficient as ρ under the Gaussianity. However, this robust alternative to ρ might loose its efficiency especially when the data involves different types of variables (*e.g.* discrete/continuous) [127]. Moreover, when the number of different values taken by either variables is small, then this might create another problem, called *problem of ties* [127] (*i.e.* there is a tie efficiently ranking the data which affects considerably the quality of r [53; 127]), which affects considerably the quality of r. In such cases, the loss of efficiency might not be compensated by the robustness in practice. Recently, statisticians have started to investigate actual combinations between Pearson and Spearman correlations. For this purpose, statisticians have recently came with an interesting combination between

Pearson and Spearman coefficients, namely Gini correlation (ξ), which has been proposed in [124]. Spearman correlation r , which is just the Pearson correlation ρ applied on already ranked data, can be defined using the notion of cumulative distributions as:

$$r_{(X,Y)} = \rho_{(F_X,F_Y)} = \frac{1}{\sigma_{F_X}\sigma_{F_Y}} Cov(F_X(X), F_Y(Y)) \tag{2.14}$$

where F_X and F_Y are the cumulative distribution of X and Y, respectively. The Gini correlation coefficient is given by:

$$\xi_{X,Y} = \frac{Cov(X, F_Y(Y))}{Cov(X, F_X(X))} \tag{2.15}$$

Note that in general ξ is not symmetric, that is $\xi_{X,Y} \neq \xi_{Y,X}$. In practice, the choice between the two forms, depends on the type of variables X and Y. In statistics, it has been reported that if, for instance, X is discrete and Y is continuous, then $\xi_{Y,X}$ would be a good choice.

2.3.2.2 Practical Computation of Gini Correlation & Properties

Consider n couples (X_i, Y_i) with $i \in$ [1..n] of independent variables drawn from a bivariate distribution. If these couples of variables are ordered (sorted from low values to high values) with respect to the X_i, new couples of variables $(X_{(i)}, Y_{[i]})$ can be generated, where $X_{(1)} <\ldots< X_{(n)}$ and $Y_{[1]},\ldots, Y_{[n]}$ the related concomitants [101], which depend on the ordering of the X_i. As proposed in [124], $\xi_{Y,X}$ is computed as:

$$\xi_{Y,X} = \frac{\sum_{i=1}^{n}(2i - 1 - n)Y_{[i]}}{\sum_{i=1}^{n}(2i - 1 - n)Y_{(i)}} \ . \tag{2.16}$$

Note that, $\xi_{X,Y}$ is computed in the same way as $\xi_{Y,X}$, by just reversing the roles of X and Y.

Authors in [155] showed several properties of ξ. The most interesting ones are:

- $|\ \xi\ | \leq 1$.
- $\xi_{X,Y} = \xi_{Y,X} = \pm 1$ if X is a monotone increasing (decreasing) function of Y.
- If X and Y are independent , then $\xi_{X,Y} = \xi_{Y,X} = 0$.
- $\xi_{Y,X}$ is not sensitive to monotonic transformation of X, as for Spearman's correlation coefficient, r.

- $\xi_{Y,X}$ is not sensitive to linear monotonic transformation of both X and Y, as for Pearson's correlation coefficient, ρ.

We compared the three correlation functions using simulated traces. The leakage function of a FPGA can be modelled as $\mathcal{L}(x) = \text{HW}(x)+\alpha\cdot\delta(x)$, where $\delta(\cdot)$ is the Kronecker symbol. Here $\mathcal{L}$ is the leakage function and HW is the Hamming weight function. Kronecker symbol $\delta(x)$ is 1 when $x = \pm 1$ else 0. We verified this leakage model on AES traces of DPA contest v2 [149]traces as shown in Fig. 2.6. As we attack the output of an AES s-box, in the linear model we take the 8 bits at the output of the Sbox as base vectors giving base function of length 9 (8 bits and 1 constant). Thus, HW function can take 9 possible values (HW=0,1,2,3,4,5,6,7,8). Fig. 2.7 (a) shows comparison of three correlation coefficient as a function of α. A proper approximation for the Gaussian case is seen at when α is 0 and all three correlation coefficients are equivalent empirically. When α is negative, Pearson correlation is not efficient. Spearman and Gini still perform very well as they are not sensitive to monotonic transformation. For positive α, Pearson is not efficient as well and Spearman also becomes less efficient as the function $\mathcal{L}$ is no more monotonic. Since the transformation is not drastic we see that Spearman correlation tries to stabilise itself (Fig. 2.7 (b)). However, Gini does show some improvement over Pearson and Spearman. As stated earlier that Gini is a combination of Pearson and Spearman, we can say that combination can help in non-ideal cases. Next, we propose some empirical approach to combine Pearson and Spearman.

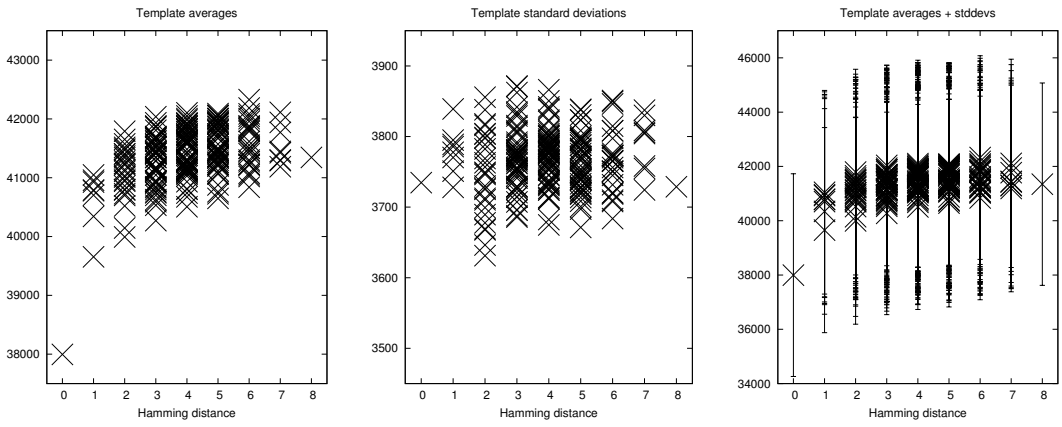
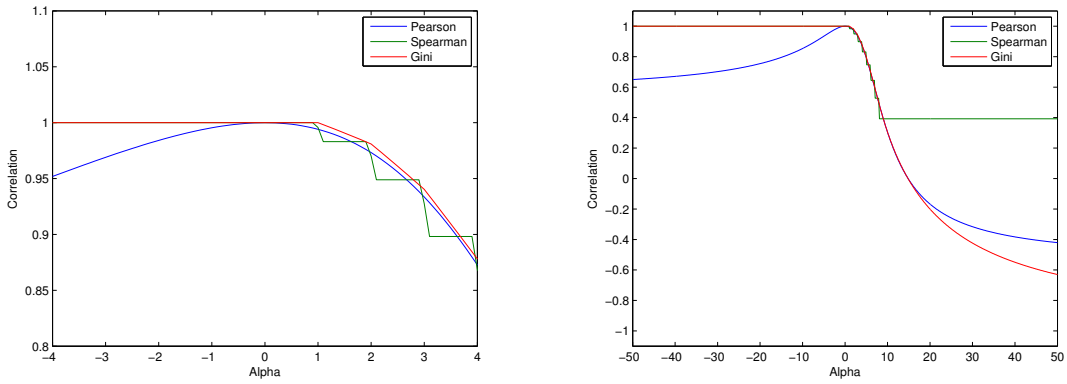


Figure 2.6: Leakage function of Sbox 0 (DPA contest v2).



(a) (b)

Figure 2.7: Leakage function of Sbox 0 extended in (b) to higher values of α. (DPA contest v2).

2.3.2.3 Pearson-Spearman Combination: An Empirical Approach

Why the combination works

As stated previously, the main difference regarding most SCAs relies on the measurements partitioning process and the used distinguisher. Otherwise, they usually run iteratively and a new ranking of all secret hypotheses is created at each iteration. Our starting argument to combine two different distinguishers, $(\Delta_k)_{sca'}$ and $(\Delta_k)_{sca''}$, involves four observations: the first observation is that the two distinguishers are equivalent (i.e. similar evolution), in terms of First-order success rate and Guessing entropy metrics, when performed in parallel, on the same set of Side-channel traces. In addition, we observe that the *secret key* mostly keeps the same temporal position for both distinguishers unlike the false key hypotheses. We define the *predicted key*[1]as the key hypothesis that has the best rank PK for the current iteration. Its value is updated for each trace processed. We observed that the two distinguishers often do not have the same predicted key ($PK_{(\Delta_k)_{sca'}} \neq PK_{(\Delta_k)_{sca''}}$). But more importantly, this emphasizes the fact that $(\Delta_k)_{sca'}$ and $(\Delta_k)_{sca''}$ are statistically different, even if they are exploiting the same dependency. Eventually, the last observation is the that *secret key* is always ranked among the best ranked key hypothesis for both distinguishers. In fact, when the attack succeeds, the *predicted key* is the actual *secret key*, as we are doing the correct partitioning of traces for each iteration. The *secret key* achieves a Guessing entropy of zero when the attack succeeds. This is not the case for *false keys* which should have an

[1]The *predicted key* is also known as *the best key*.

unstable (random) rank. For more details about key ranking behaviour analysis, refer to Sec. 2.4.

Combination formula

Consider two Side-channel attacks, sca' and sca'', that verify the empirical observations mentioned before. Let $\Delta_{vect_{sca'}} = ((\Delta_k)_{sca'})_{k=1}^{K}$ and $\Delta_{vect_{sca''}} = ((\Delta_k)_{sca''})_{k=1}^{K}$, respectively. We can combine sca' and sca" distinguishers by taking into account the scores given by both distinguishers for the same key hypothesis, k. We use aggregate functions for the combination (like the Max() and the Sum() functions). Similarly, Gini correlation can be imagined to use ratio as an aggregate function. Let Ψ be such function. For each key hypothesis, k, a new score is generated by computing $\Psi((\Delta_k)_{sca'}, (\Delta_k)_{sca''})$, which is the aggregate function of $(\Delta_k)_{sca'}$ and $(\Delta_k)_{sca''}$. This way, a new vector of scores $\Delta_{vect_{combi}}$ is built. An illustration of the combination mechanism is shown in Fig. 2.8.

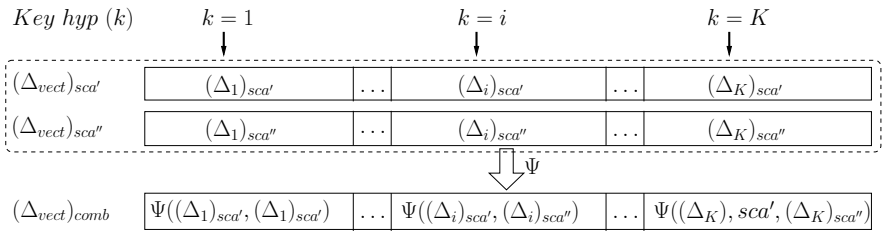
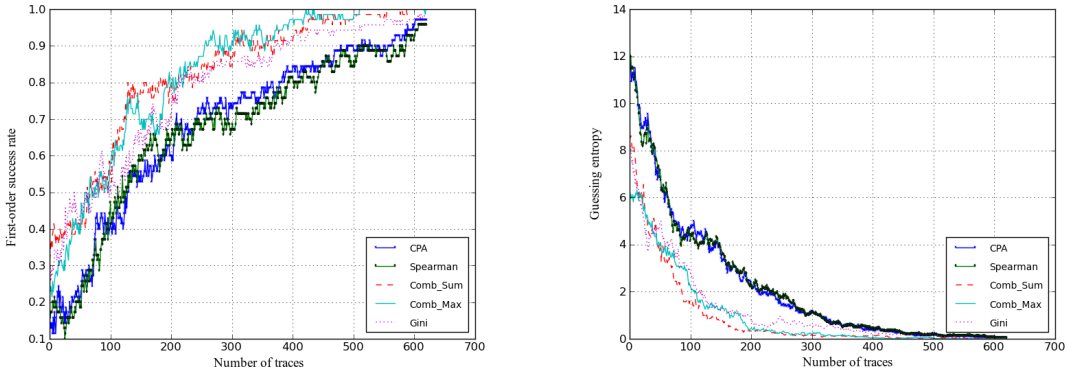


Figure 2.8: The mechanism of combination using an aggregate function Ψ.

2.3.2.4 Experimental Results & Discussion

In this experiment, we recorded 5000 Side-channel traces (averaged 256 times) related to the activity of an unprotected DES crypto-processor. As studied previously in Sec. 2.2, the analysis of the marginal distributions of both, the prediction X and the measurement Y, revealed that their joint distribution is not perfectly Gaussian. In fact, there is some deviation from the bivariate normal assumption; and therefore the Pearson correlation coefficient, ρ, might not be efficient. Moreover, Spearman Correlation Coefficient, r, might not be efficient too, because X takes a small number of different values which does not allow an reliable approximation to normal distribution. Indeed, this might create a problem of ties, which affects considerably the quality of r. The experiment that we have conducted involves five Side-channel attacks evaluated in term of their First-order success rate and Guessing entropy security metrics: CPA, Spearman rank correlation, Gini correlation, and two empirical combination attacks.

Indeed, two aggregate functions have been investigated: the Sum(), and the Max(). These two attacks are denoted by $Comb_{Sum}$ and $Comb_{Max}$, respectively. According to Fig. 2.9 (a) and Fig. 2.9 (b), CPA and Spearman attacks have similar behaviours.



(a) (b)

Figure 2.9: CPA, Spearman vs Combination: (a) Success Rate and (b) Guessing Entropy.

This agrees with our empirical statements stated previously. Clearly, the combined attacks (Gini Correlation, $Comb_{Max}$ and $Comb_{Sum}$) outperform CPA and Spearman attacks. As a matter of fact, for a First-order success rate threshold fixed at 80%, the number of traces needed to succeed the combined attacks is around 200 traces. CPA and Spearman attacks need much more traces to do so (400 traces) and thus the gain is about 100%. Unsurprisingly, the Guessing entropy shows a superior efficiency for the combined attacks as the rank of the *secret key* converges more rapidly toward the best rank, than CPA and Spearman attacks. Besides, for both metrics, Gini correlation is little bit less efficient than the empirical combinations, $Comb_{Sum}$ and $Comb_{Max}$. Let S_1, S_2 be two inputs of aggregate function with respective noise of standard deviation σ_1, σ_2. The SNR of the $Comb_{Sum}$ is $(S1+S2)/\sqrt{{\sigma_1}^2+{\sigma_2}^2}$. When S_1 and S_2 are equal, the SNR of combination using $Comb_{Sum}$ is increased by $\sqrt{2}$. Similarly, the increase in SNR when two distinguishers are combined using $Comb_{Max}$ can be computed. However, Gini Correlation is more **generic** and might be more efficient in other empirical circumstances.

2.3.3 Conclusion

In this part of Chapter, we have presented new methodologies of combined attacks. The methodologies proposed combine commonly used Side-channel distinguishers, Pearson

and Spearman coefficients, both theoretically (Gini correlation) and empirically (aggregate function). Please note that we intend to propose **generic** methodologies to accelerate attacks and not attacks in particular. Depending on the target, different distinguishers can be combined using appropriate aggregate functions. Choice of aggregate functions depends on the practical behaviour of distinguishers. For this purpose, in the next Section, we propose to give a special attention to such behaviour, by analysing the rank evolution of key hypotheses. Summarising, we would like to say that Side-channel attacks have significantly improved over the years. At this point, it should be interesting to combine the advantages of various attacks.

2.4 Secret Key Rank Correction

2.4.1 Introduction: Background Knowledge

2.4.1.1 Rank-based SCAs

As stated in the introduction of this Chapter, the main difference regarding most SCAs relies on the measurements partitioning process and the used distinguisher. Otherwise, they usually run iteratively and a new ranking of all key hypotheses is created at each iteration. Then, when the first ranked hypothesis is stable for a certain amount of iterations, it is returned by the SCA software. The attack is successful when it is, indeed, the actual secret. Although various biases and noises are introduced by implementations, architectures, and especially measurements acquisition tools, we will show that the rank of the secret key displays a specific behaviour with regards to other hypotheses. This part of Chapter is based on the study of such behaviours, and explains how to exploit them in order to enhance existing SCAs.

2.4.1.2 Notations

In the rest of this Chapter, we will use the following notations:

- RC is the *Rank Corrector*.
- SK is the secret key.
- PK, the predicted key, is the key hypothesis which has the best rank for the current iteration. The value of PK is updated for each new observation.
- PK_i denotes the predicted key at iteration i.
- FK represents a false key hypothesis (all but the secret key).
- R_k, $R_{k,i}$ are respectively the ranks of key hypothesis k for the current iteration and iteration i.
- S_{init} is the iteration number corresponding to the beginning of stability for a given PK.
- MTD, or Measurement To Disclosure, is the total number of traces needed to successfully perform the attack.

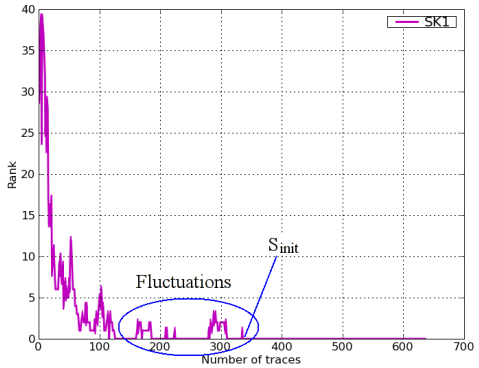
2.4.1.3 Key Rank Behaviours

In theory, for a great number of observations, SK should always be ranked first, as we are doing the correct partitioning of traces for each iteration. This is not the case for FKs which should have an unstable (random) rank. However, actual attacks are usually performed with a limited number of measurements, or aim, at least, to be successful using as few of them as possible. Therefore, we studied the behaviours of the ranks of both the secret key and false key hypotheses, by performing numerous DPAs and CPAs on four different architectures of DES and three of AES, implemented on Altera Stratix-II and Xilinx Virtex-II FPGAs. The goal of this study is to find an empirical method taking advantage of the distinctive behaviours between SK and the FKs.

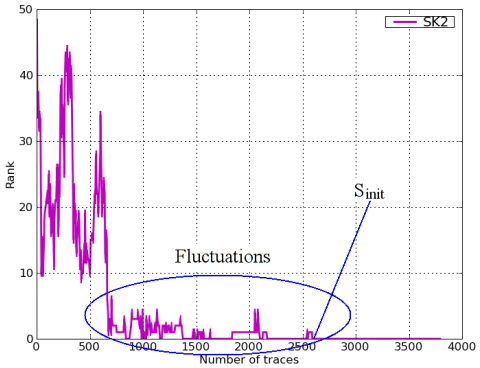
First of all, we observed that R_{SK} is always roughly decreasing until it reaches first position, considering that the best rank is 0. Fig. 2.10 shows examples of such behaviour during a DPA on 6 bits of the first S-Box of a DES coprocessor (a) and a CPA on 8 bits of the first S-Box of an AES 256(b), both implemented on FPGA. While in Fig. 2.10(a) R_{SK} decreases almost monotonically, in Fig. 2.10(b) it oscillates much more while doing so. Then, in both cases, R_{SK} clearly fluctuates within a short range of the first positions before definitely stabilising. This behaviour is observed most of the time, however, in rare cases, R_{SK} can stabilise as soon as it reaches the first position, without fluctuating.

Regarding false keys, we observed that, as the number of processed traces increases, they clearly tend to display more random behaviours. Fig. 2.11 shows two examples of false key rank evolution during the same DPA as Fig. 2.10(a). On the one hand, the leftmost one ($FK1$) is almost random and never ranks first, thus will not be treated as a potential secret key. This type of behaviour is easily differentiable from SK. On the another hand, the rank of the rightmost key ($FK2$) does reach the first position at some point and could therefore be concurrent to SK. However, with the increasing iteration number, R_{FK2} clearly raises, which would not be the case for SK.

In conclusion, our study shows that it should indeed be possible to differentiate between SK and the FKs based on the observation of their ranking. As a matter of fact, R_{SK} is roughly decreasing and then usually fluctuates between a few positions before stabilising, whereas the $R_{FK}s$, when they reach the first rank, usually become random a few iterations later.

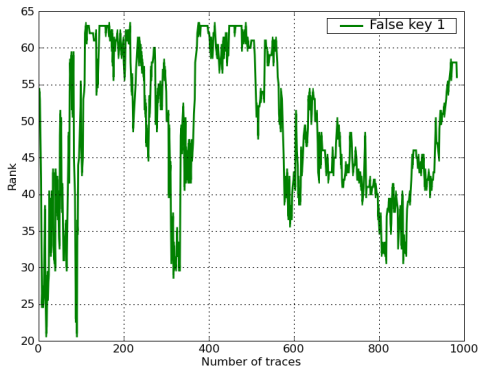


(a) Rank of SK during a DPA on DES

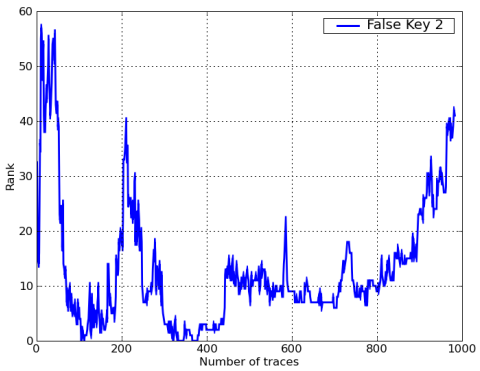


(b) Rank of SK during a CPA on AES

Figure 2.10: Examples of rank behaviours for the secret key.



(a)



(b)

Figure 2.11: Examples of rank behaviours for false keys.

2.4.2 Rank Corrector: Principle

2.4.2.1 Application Field

The main feature of the Rank Corrector is to be as generic as possible. It can therefore be applied to a wide range of attacks. Indeed, an attack scheme only needs to meet three requirements to be compatible, with our software:

1. It has to be iterative (for instance DPA iterates on a number of power consumption measurements).

2. A ranking of all key hypotheses must be produced at each iteration.

3. The SCA software must decide on the secret key by observing the stability of the first ranked hypothesis.

Up to now, the usual criterion employed to decide on SK is, indeed, the stability of the rank. For example, in the first edition of DPA contest v1 [148], a stability of 100 traces had to be achieved in order to validate the attacks [143], performed on an unprotected DES crypto-processor. Moreover, most of the passive SCAs, usually based on the exploitation of power consumption or electromagnetic measurements, present an iterative behaviour as, at some point, they process those traces one after the other.

Thereby, the Rank Corrector can be used to enhance a very large number of attacks, like DPA, CPA or MIA.

2.4.2.2 Basic Principle

The Rank Corrector (RC) is a **generic**[1] custom-made algorithm, which aims at exploiting the key behaviours described in Sec. 2.4.1.3 in order to significantly reduce the number of traces needed to perform a successful SCA.

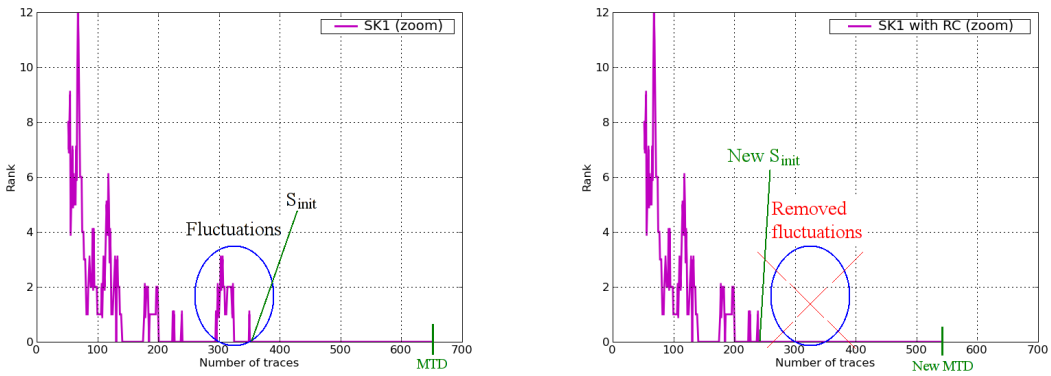
As a matter of fact, it studies in real-time, the evolution of an iterative ranking (for instance the one produced by a DPA software), in order to virtually reassign previous rank positions to the current PK, depending on past and current rankings. The detailed algorithm is described in Sec. 2.4.2.4. It is totally independent of the attack, given that it verifies the requirements described in Sec. 2.4.2.1. Indeed, it only modifies, on the fly, the stability of the target SCA, while creating a new ranking in parallel (for the sake of displaying the results). Eventually, RC can be seen as a plug-in, designed to enhance most existing SCAs.

Now suppose that we are performing a DPA on one sub-key of a cryptographic device implementing an algorithm like DES or AES, and that it will be successful, meaning that SK will eventually be ranked first and reach the given stability. Before stabilising, R_{SK} should be roughly decreasing (as stated in Sec.2.4.1.3). Then most of the time, after reaching the lowest position (*i.e.* best rank), it will fluctuate within a short range, and then stabilise, from S_{init} to MTD.

In this case, RC will detect those fluctuations in the proximity of the stabilisation, remove them and increase the stability counter by an equal amount (G) of traces. Thereby, G represents the gain of RC with regards to a simple DPA. Fig. 2.12 illustrates

[1] We insist that our methodology does not consist in trying to tune an attack on a given acquisition campaign so that it retrieves the key as fast as possible, as in [81]. Instead, we attempt to pre-characterize a set of parameters from a training campaign, and to use this prior knowledge subsequently in a positive view to speed up forthcoming attacks.

this scenario, by showing a zoom of the evolution of the R_{SK} displayed in Fig. 2.10(a), with and without the *Rank Corrector*. As we can see in this example, without RC the stability starts after 340 traces, whereas with RC, it does after only 240 traces. Thus we have a gain of 100 traces.



(a) Basic DPA.

(b) DPA with RC.

Figure 2.12: Rank of SK during a DPA, with and without RC.

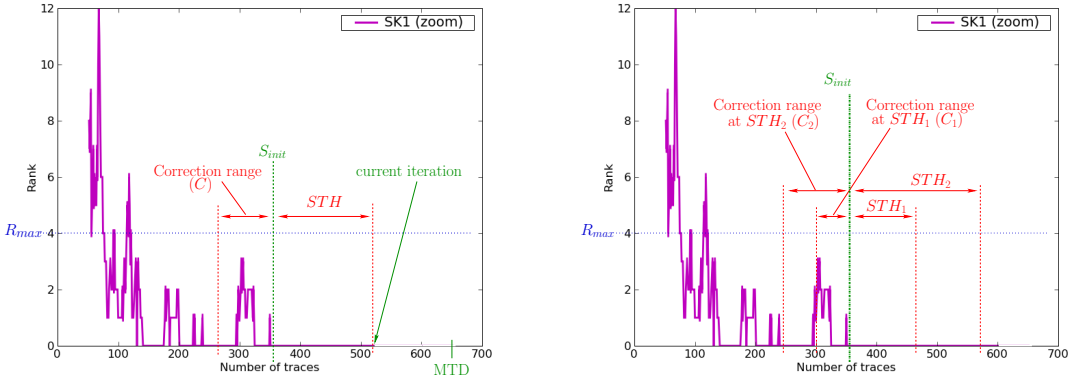
The main idea of RC is to locate and disqualify FKs that are ranked first during the fluctuations of SK. Therefore, it proceeds as follows:

1. When a PK has been stable for certain amount of traces STH (stability threshold), starting at S_{init}, it is considered as a potential SK, as shown in Fig. 2.13(a).

2. Then RC scans a small range of traces before S_{init} (called *correction range*) searching for fluctuations of the current PK (let's call it CPK). If they exceed a certain limit R_{max}, CPK is disqualified and will no longer be a candidate for SK.

3. In the other case, RC will check the ranks, at the current iteration, of all other PKs present in the correction range and discard all those that show a rank exceeding R_{max}. Then the rank of SK, within the correction range, is modified by removing the discarded keys.

Moreover, RC operates by using increasing values of STH. Each time the stability of PK reaches given values STH_n (with $n \in \mathbb{N}^*$), RC is launched. This threshold mechanism was chosen for two reasons. On one hand, it allows RC to easily discard any PK that isn't stable for at least STH_1 , and only take into consideration the potential



(a) RC principle.

(b) RC threshold mechanism.

Figure 2.13: Illustration of RC principle.

SKs. Moreover, for each new threshold, the correction range (i.e. the potential gain) increases, as shown in Fig. 2.13(b) which is coherent with the fact that the more stable a PK is, the more likely it is to be SK. On another hand, it keeps the computation time of the attack close to the original one as RC is only called a few times.

2.4.2.3 RC Parameters & Evaluation

In order to be as generic as possible, RC was designed as a **parametric** algorithm. It is thereby based on two main parameters, that allow RC to adapt to existing SCAs, independently of any bias introduced by either the architecture, the implementation, or the acquisition technique:

1. S is the minimum stability required to ensure that the attack will always decide on the actual secret key (SK).

2. R_{max} is the maximum fluctuation range of SK before the stabilisation (i.e. between the first time SK attains the first position, and its stabilisation).

These two parameters must be correctly evaluated for RC to work properly. For instance choosing S too small, could easily lead any SCA to decide on a false key, and using RC in this context would clearly increase the probability of doing so.

For this purpose, we use a clone device, which is often available from the security evaluation perspective, and undergo a profiling phase, computing multiple attacks for different SK. For each key and each attack, we record the evolution of the ranks of SK (R_{SK}) and every FK, in order to determine the best R_{max} and S. Indeed, the more representative these two parameters are of SK (and not of any FK), the more

efficient RC will be, as it will be able to disregard more FKs, and almost only consider the actual SK as a correction target.

Thanks to this profiling, we are then able to ensure that RC will not lead to finding a false key.
Aside from R_{max} and S, RC takes into account a few other secondary parameters: while these parameters do influence the maximum gain of RC, they do not have a major impact on the overall results. Therefore, and in order to simplify the notations, they will be fixed, in the rest of this study, to the values we used during our experiments.

1. n: the number of thresholds, $n \in 1, 2, 3$.
2. STH_n: the nth stability threshold, $STH_n = (n * S)/4$.
3. C_n: the nth correction range, $C_n = STH_n/2$.

Those empirical values were deduced from thorough studies on several cryptographic devices. Naturally, they may not be optimal for all SCAs and all implementations, and a finer study should be carried out, using the clone device, before every specific attack.

2.4.2.4 Algorithm Description

Algo. 1 gives a detailed description of RC. When launched, it starts by searching for an occurrence of the current PK (CPK), in the first rank of the C_n iterations before S_{init} (the current iteration number being $CIT = S_{init} + STH_n$). The search starts at $S_{init} - C_n$ down to $S_{init} - 1$ (step 2 and 3 of our algorithm). Finding CPK at iteration IT, means that R_{CPK} did actually reach first position and fluctuate before stabilising, meaning it is, as such, open to correction by RC (step 4).

All PKs between S_{init} and IT are then checked in the reverse order (from S_{init} to IT), and reassigned to CPK when possible (step 5). This way, whenever a rank that should not be corrected is found, RC is stopped. Several scenarios can then occur: the trivial one is when $PK_j = CPK$, with $j \in S_{init}$ *to* IT (though it is never true for $j = S_{init} - 1$). In this case, RC directly increases the stability by one iteration. When $PK_j \neq CPK$ (step 9), RC will look at $R_{CPK,j}$. If $R_{CPK,j} < R_{max}$ (i.e. $R_{CPK,j}$ is near the first position), that means CPK is a possible candidate to be SK (step 10). RC then checks if $R_{PK_j,CIT} \geq R_{max}$, and if this second condition is verified, PK_j is disqualified as a potential SK, and removed from the ranking (step 12 and 13). This check is mandatory, as, for the first thresholds, CPK could be a false key, in which case the real SK is likely to be one of the PK_j, with $j < S_{init}$. This step is repeated until $CPK = PK_j$ or a PK_j that cannot be disqualified ($R_{PK_j,CIT} < R_{max}$) is found.

Indeed, when $R_{PK_j,CIT} < R_{max}$ our algorithm considers PK_j to be a possible SK and thus no correction is made. Then, if $CPK = PK_j$, the stability is once again increased (step 15). As a matter of fact we suppose, based on the observations of Sec. 2.4.1.3, and the profiled value of R_{max} that R_{SK} will never go past R_{max} once it has been ranked first. Thus any PK that goes past R_{max} is definitively discarded.

These steps are repeated for each threshold, and each time the number of traces that might be corrected increases. As a matter of fact, the more stable a PK is, the more likely it is to be SK. Moreover, a FK that was not corrected at the first threshold, (for instance because it was ranked second), will usually not be in the same position at the second threshold, and will then be replaced by PK.

Consequently, the maximum gain of RC can be computed as shown in Eqn. 2.17:

$$GAIN_{max} = \sum_{n=1}^{3} \frac{STH_n}{2} \equiv \sum_{n=1}^{3} \frac{n * S}{8} \ . \tag{2.17}$$

2.4.2.5 Case Study

Fig. 2.14 illustrates the evolution of key ranks during an SCA using RC, when the stability of a given key reaches the first threshold. K represents our secret key, and S_{init} the iteration number marking the beginning of its stability, while K_0, K_1 and K_2 are three false keys. The process of RC can be described in three steps:

1. The rank of K (R_K) reaches the first threshold (i.e. a stability of $STH_1 = S/4$ traces), thus RC searches for K in the PKs of the $S/8$ prior traces. It is found at iteration IT, implying that R_K did actually reach rank 0 and fluctuate before stabilising.

2. Two different PKs, K_1 and K_2 are found respectively at iteration $S_{init} - 1$ and $S_{init} - 3$. For those iterations R_K is compared to R_{max}. As $R_K \leq R_{max}$ is true in both cases, RC enters the next step of the algorithm and checks the ranks of K_1 and K_2 at the current iteration $CIT = S_{init} + S/4$.

3. $R_{K_2,CIT}$ is greater than R_{max}, so K_2 is discarded. Then, K which was ranked second, becomes the new PK of iteration $S_{init}-1$ and the stability is increased. K is already ranked first at iteration $S_{init}-2$ so the stability is once again increased. $R_{K_1,CIT}$, on another hand, does not verify the condition, meaning it is a possible candidate for SK, and RC is therefore stopped.

Algorithm 1: RC detailed algorithm.

```
1: for each threshold STH_n (n ∈ 1,2,3) do
2:   for iteration in S_init − C_n to S_init do
3:     Search for an occurrence of the current CPK.
4:     if CPK is found at iteration = IT then
5:       for j in S_init to IT do
6:         Check the value of PK_j
7:         if PK_j = CPK then
8:           increase stability by 1
9:         else
10:          if R_{CPK,j} < R_max then
11:            while CPK ≠ PK_j and R_{PK_j,CIT} ≥ R_max do
12:              Remove PK_j from the ranking
13:            end while
14:            if CPK = PK_j then
15:              increase stability by 1
16:            end if
17:          else
18:            Exit.
19:          end if
20:        end if
21:      end for
22:    else
23:      Exit.
24:    end if
25:  end for
26: end for
27: return stability
```

In this example, RC produced a gain of 2 traces after the first threshold, and S_{init} is thereby updated as shown in Fig. 2.14.

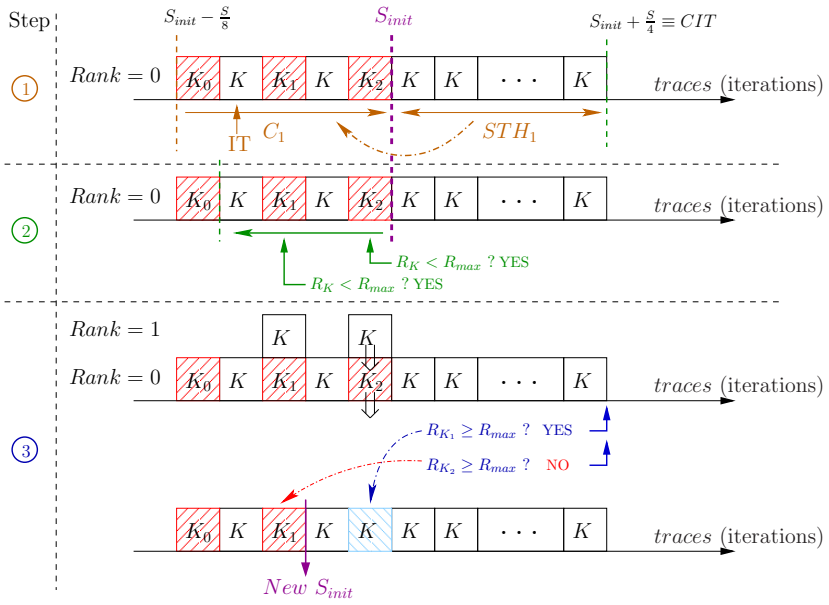


Figure 2.14: Illustration of an SCA using RC, at the first threshold.

2.4.2.6 Optimization

Although most of the time SK does fluctuate before stabilising, there are rare cases where it permanently stabilises as soon as it reaches the first position (this occurred in less than 4% of all the attacks we performed during our study). In this case, the gain should be null. In order to decrease the probability of having such a gain, we take advantage of the fact that R_{SK} is likely to be $< R_{max}$ just before stabilising. Thus RC will search for occurrences of SK in those ranks, before the stabilisation, and try to disqualify the corresponding PKs, in order to reassign SK to the first rank. This new search range, called $C2_n$, is another secondary parameter (like n and C_n), and will also be fixed in the rest of this study to our experimental value: $C2_n = STH_n/4$ (see Sec. 2.4.2.3).

Algo. 1 is then complemented as follows: step 18 is replaced by Algo. 2. Then when CPK is not present in the first search range (step 17 of Algo. 1) RC will check if $R_{CPK} < R_{max}$ for a smaller range of iterations: from $S_{init} - C2_n$ to S_{init}. The following process is similar to the former one, if all PKs can be disqualified ($R_{PK_k,CIT} \geq R_{max}$), CPK becomes PK_k and the stability is increased. Obviously SK will not always display such a behaviour, and there will thus be cases where the gain is null. A more

thorough study of these situations could certainly result in an improvement of our algorithm, but is out of the scope of this study.

Algorithm 2: RC optimization.

```
1: if CPK is not found then
2:   for k in S_init to S_init − I2_n do
3:     if R_CPK < R_max then
4:       while CPK ≠ PK_j and R_PK_j,CIT ≥ R_max do
5:         Remove PK_j from the ranking
6:       end while
7:       if CPK = PK_j then
8:         increase stability by 1
9:       end if
10:    else
11:      Exit
12:    end if
13:  end for
14: end if
```

2.4.3 Experiments & Results

Our experiments were conducted on Stratix-II FPGAs, soldered on two SASEBO-B boards provided by the RCIS [128] (one for the actual attack an one for the clone device). The target crypto-processor implemented in those devices is an unprotected DES. Power consumption measurements were acquired, using a differential probe plugged to the positive rail of the FPGA core power supply through a 1 Ω shunt resistor, coupled with a 54855 Infiniium oscilloscope from Agilent Technologies [5].

First of all, we estimated the parameters (S and R_{max}) with a profiling phase on the first FPGA. Using three different keys, we performed 100 attacks for each one. Eventually, $S = 110$ and $R_{max} = 5$ were deduced as the optimal values for these parameters. Then, we acquired 50000 traces in order to perform several DPAs on the real device, with and without using the *Rank Corrector*.

The improvement brought by our scheme is clearly visible on the curve of the First-order success rate in Fig. 2.15. For instance a Success rate of 80% is reached with less than 90 traces with RC, when the basic DPA needs more than 110 traces to do so.

While the Guessing entropy is also always lower with RC than without, the gap between the two is definitely thinner than for the Success rate. This is explained by the fact that the correction takes place when the rank of SK is lower than 5, so when computing the mean of ranks on a large number of attacks, it doesn't have a great

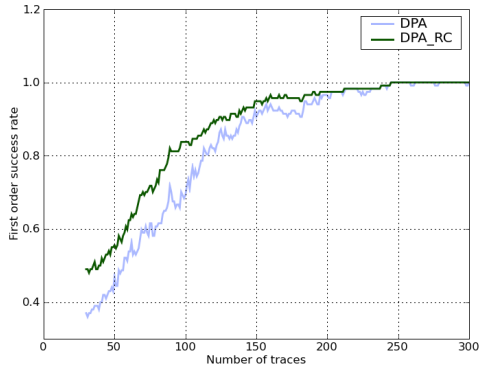


Figure 2.15: First-order success rate for DPA with and without RC.

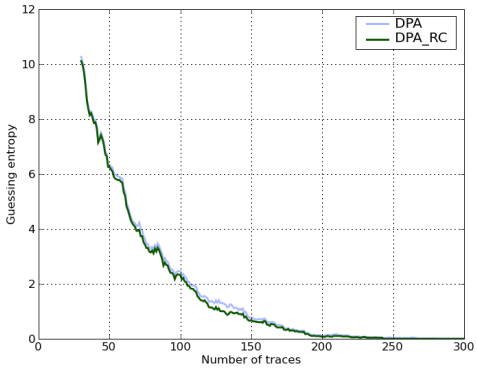


Figure 2.16: Guessing entropy for DPA with and without RC.

impact on the final results.

Moreover, after 300 complete attacks on random pool of traces, we obtain a mean gain of 43.7 traces. Considering that the basic DPA requires 250 to 300 traces to complete the attack, as shown in Fig. 2.15, we conclude that using RC results in a gain of $\sim 15\%$ in terms of MTD.

2.4.4 Conclusion

In this part of Chapter, the Rank Corrector algorithm which aims at enhancing most of SCAs has been presented and evaluated. The principle based on the distinguishable evolution of the ranking between the secret key and false keys has been observed on different SCAs. Two parameters which are the threshold of stability and the maximum range are necessary for the RC to work efficiently. We have shown that these parameters can be nicely determined using a clone device, which is often available in the case of the evaluator. When these parameters are well profiled, a significant gain in terms of MTD can be expected, especially when dealing with unprotected implementations, with regards to the classical DPA. Besides, we assume that the efficiency of RC is mainly dependent on the amount of noise affecting Side-channel traces and the features of used distinguisher (robust/non robust coefficient, univariate/multivariate). In the context of investigating new Side-channel distinguishers, in the next Section 2.5 we propose a new powerful multivariate distinguisher and show its efficiency with regards to existing ones like DPA, CPA and VPA.

2.5 First Principal Components Analysis for Secret Key Recovery

2.5.1 Introduction

In this part of Chapter, we outline the way how Principal Component Analysis (PCA [126]) could be used to extract the value of the secret key. PCA is a multivariate data analytic technique [126; 136] that has found application in fields such as computer vision [140], robotics [158], sociology and economics [138]. It is a way of identifying patterns in multidimensional data set, and visualising these data into a lower dimensional space, in order to highlight their similarities and differences. In the SCA techniques portfolio, PCA has already revealed its efficiency on Template attacks [116]. As described in [116], PCA improves the class of Template attacks by pre-processing the leakage traces before performing the attack on the real cryptographic device. Indeed, in the pre-processing phase the evaluator builds Templates in order to profile the clone device. Then, those templates are used to mount an attack on the real device. Our attack uses PCA no more as a pre-processing tool but as a distinguisher. Moreover, it follows the usual steps of a basic SCA algorithm that consists of only one phase and does not require a clone device for profiling, which makes the task of the evaluator easier. Also, in [55], the first eigenvalue of a PCA is used as a security metric to measure the leakage of several implementations protected against SCAs via *so-called* hiding countermeasures. Besides, as stated before, the leaked information can be statically defined by a continuous random variable for which the probability law P_{law} is unknown or uncertain; and therefore the main challenge of SCA is to make a sound estimation of P_{law} without loss of information. We assume that any cryptographic implementation could be attacked by exploiting its sensitivity against one Chosen Statistic (CS), that could be for instance the mean, the variance or any other statistic describing P_{law}, the leakage distribution. The higher the sensitivity, the greater the vulnerability of the implementation against attacks based on the considered CS.

In this study, first, we attempt to give some elementary background that is required to understand the process of PCA. Second, this background knowledge is taken advantage of, to outline the way how PCA could be exploited to mount an efficient attack by going through the different steps needed to perform the *First Principal Components Analysis* (FPCA). Third, we show and comment results given by the proposed FPCA attack. Indeed, we highlight the efficiency of FPCA by making a comparative analysis with existing attacks (DoM, DPA, CPA, VPA).

2.5.2 Principal Component Analysis: Background Knowledge

Let a data set of M quantitative variables describing N samples, arranged respectively in rows and columns. The goal of PCA is to ensure a better representation of the N samples by describing the data set with a smaller number M' of new variables. Technically speaking, PCA proposes to seek a new representation of the N samples in a subspace of the initial space by defining M' new variables which are linear combinations of the M original variables, and that are called principal components. Generally speaking, reducing the number of variables used to describe data will lead to some loss of information. PCA operates in a way that makes this loss minimal. For PCA to work properly, the data set should be centred. PCA starts by computing the covariance matrix of the data set in order to find the eigenvectors and eigenvalues which permit the capture of the existing dispersion in variables. In other words, it makes a change of orthogonal reference frame, the new variables being replaced by the Principal Components which are totally characterized by the associations of the eigenvectors and eigenvalues. But more importantly, these associations reveal the hidden dynamics of the data set. Determining this fact allows the evaluator to discern which dynamics are important and which are just redundant. The first component can be expected to account for a fairly large amount of the total variance. Each succeeding component will account for progressively smaller amounts of variance. In practice, the evaluator sorts eigenvectors by their eigenvalues, from the highest to the lowest. This gives the components in order of significance. Most of the time, only few M' components account for meaningful amount of variance. Thus, only these first M' components will be retained. The decision on the number of the M' best components could be achieved by performing some deciding tests such as the Kaiser criterion, the scree test or the cumulative variance criteria ([69]).

2.5.3 FPCA: Attack Process

In the SCA field, PCA has often been used as pre-processing tool to minimize the coding complexity by reducing the dimensionality of recorded traces [14; 142]. Our approach is different in the sense that PCA is used in the very core attack to retrieve the secret information. Indeed, FPCA uses the projection on the first principal components to tell good *secret key* candidates from incorrect ones. FPCA shares some key points with first-order SCA. As stated before, FPCA does not require a detailed knowledge about the cryptographic device to be performed (*i.e.* no clone device required). It only exploits data dependency of the power consumption of the device under attack. The

main difference with first-order SCAs resides in the way to distinguish the behaviour of the good key hypothesis. In fact, we remind that each attack has its own statistical test, referred to as distinguisher [51; 144], which allows the evaluator to detect the value of the secret key. In this context, FPCA comes with a new distinguisher for Side-channel analysis. In the rest of this section, we detail the different steps needed to perform a FPCA, while introducing our notations at the same time.

2.5.3.1 Preliminary Preparation Phase

This phase is common with differential and correlation power attacks. Suppose that T power consumption traces are recorded while a cryptographic device is performing an encryption or a decryption operation. Collected traces are L-dimensional time vectors. The evaluator chooses an intermediate result of the cryptographic algorithm that is processed by the cryptographic implementation. The intermediate value denoted by $v_{d,sk}$ is a function that takes two parameters. The first parameter d is a known data value that can be either the plain text or the cipher text. The number of data values is equal to T, the number of recorded traces. These known data values are represented by a vector $D_{vect} = (d_1, d_2, \ldots, d_T)$ of size T. The second parameter sk is secret, hence unknown. Indeed, sk is a small part of the cryptographic key and can take K possible values referred to as key hypotheses that we write as a vector $K_{vect} = \{k_j\}_{j=1}^{K} = (k_1, k_2, \ldots, k_K)$.

Thus, the *trace* can be written as a matrix of size $L \times T$. Given vectors D_{vect} and K_{vect}, the evaluator is able to compute, without difficulties, (hypothetical) intermediate values for all K key hypotheses and for all T executed cryptographic operations. He builds a matrix V of size $K \times T$: $V_{i,j} = v_{d_i,k_j}$ with $1 \leq i \leq T$ and $1 \leq j \leq K$. For each value $V_{i,j}$, the evaluator computes a hypothetical power consumption value $H_{i,j}$ based on a power consumption model. R being the number of possible values that the power consumption model could take, the traces are arranged in X ($X \leq R$) different partitions for each key hypothesis k_j. We denote these partitions as a vector $P_{k_j} = (P_{k_j,1}, P_{k_j,2}, \ldots, P_{k_j,X})$ with $1 \leq j \leq K$. For instance, suppose that our power consumption model is the Hamming Distance HD and that it can take integral values from 0 to 4: $HD = \{0, 1, 2, 3, 4\} = \{HD_i\}_{i=1}^{5}$. The trivial partitioning is to associate each HD_i value to one partition. Thus $X = R = 5$. One other possibility, described in [95], is to build only X=3 partitions in this way: first partition for $HD > 2$, second for $HD = 2$ and third for $HD < 2$. Intuitively, the more accurate the used power model is, the better our description of the secret information will be. Many

papers are dealing with the investigation of new power models and techniques for traces classification [2; 109]. The optimal choice of the power consumption model, including the partitioning process, is out of the scope of this study. In what follows, our study will focus on the HD model as it is one of the most commonly used, and often one of the most efficient.

2.5.3.2 References computation

Once traces are arranged in X partitions for each key hypothesis k_j, we propose to compute for each partition a statistical trace based on one Chosen Statistic CS and referred to as $reference$. For instance, if CS is the "mean" then the $reference$ will be the average of all traces that belong to the considered partition. Actually, the X references of one key hypothesis k_j will be used by PCA as criterions to highlight differences between the X partitions. For references computation, we notice that the same CS (the mean, the variance ...) is used for all partitions and for all key hypotheses k_j. One reference is an L-dimensional time vector. Thus we have one dataset of X references, for each k_j. We denote this set by $S_{ref_{k_j}}$. In what follows, our study will focus on analysing each dataset $S_{ref_{k_j}}$ corresponding to each key hypothesis k_j. This analysis will allow the attacker to discriminate the behavior of the secret key with regards to all other key hypotheses. Moreover, it will reduce the computational complexity of the PCA step.

2.5.3.3 FPCA distinguisher

For one key hypothesis k_j, the dependencies between references are made more eligible by PCA, when the references are projected to the new axes system composed by the principal components. The PCA is used to analyze these dependencies by measuring the dispersion of the references in the new coordinate space. Indeed, the larger the eigenvalue λ that corresponds to one eigenvector is, the greater is the dispersion of the references on this eigenvector. As stated by Eqn. (2.18), the total variance VAR_{tot} of one $S_{ref_{k_j}}$ is equal to the sum of all eigenvalues corresponding to all principal components:

$$VAR_{tot} = \sum_{j=1}^{L} \lambda_j \,. \tag{2.18}$$

Given a valid power consumption model and one CS, there are two cases to be discussed regarding the fluctuation of the total variance when increasing the number of

recorded traces. The first case is the one for which the cryptographic implementation is not sensitive to the considered CS. In this case, PCA could not discriminate references of the secret key as well for the other key hypotheses.

The second case happens when the implementation is sensitive to the chosen CS. In this case, VAR_{tot} related to $S_{ref_{k_{secret\ key}}}$ is getting high by increasing the number of recorded traces. This can be explained by the fact that the secret key partitioning is the one for which the references are the most different. Intuitively, for an infinity of traces, VAR_{tot} converges towards the real leakage dispersion. At the opposite, VAR_{tot} corresponding to one false key approaches the zero value when increasing the number of traces. This is due to the fact that PCA is not able to discriminate the references.

In order to highlight the dispersion of the references related to the secret key with regards to false keys, we carried out an experiment on DES [104] power consumption traces that are made freely available on line, in the context of the first version of DPA Contest competition [148]. The DES algorithm used for the competition is unprotected and easily breakable by first-order SCA. More details about this implementation could be found in [56]. For this purpose, we used the"mean" as CS and the HD model as power consumption model. Fig. 2.17 shows the dispersion of references related to the secret key and one false key, when projected to the first and the second principal components. These principal components are the most significant given that they cover a high rate of the total variance called the explained variance (EV). For the m^{th} principal component PC_m, this rate is defined by the following ratio:

$$EV(PC_m) = \lambda_m / VAR_{tot}\,,$$

where λ_m is the eigenvalue corresponding to PC_m. For m' principal components, we introduce the cumulative explained variance (CEV) that is defined by:

$$CEV(PC_1, \ldots, PC_{m'}) = (\sum_{i=1}^{m'} \lambda_i)/VAR_{tot}\,.$$

In practice, last principal components are usually considered to be related to the noise contribution and only few m' components are retained for analysis.

For this purpose, we used the cumulative variance criteria to extract the significant components. For instance, we keep only the m' first components which explain more than 95% of the total variance, for each key hypothesis k_j.

Then, we propose to compute an indicator (or distinguisher) $F_{k_j}^{CS}$ that is defined as follows:

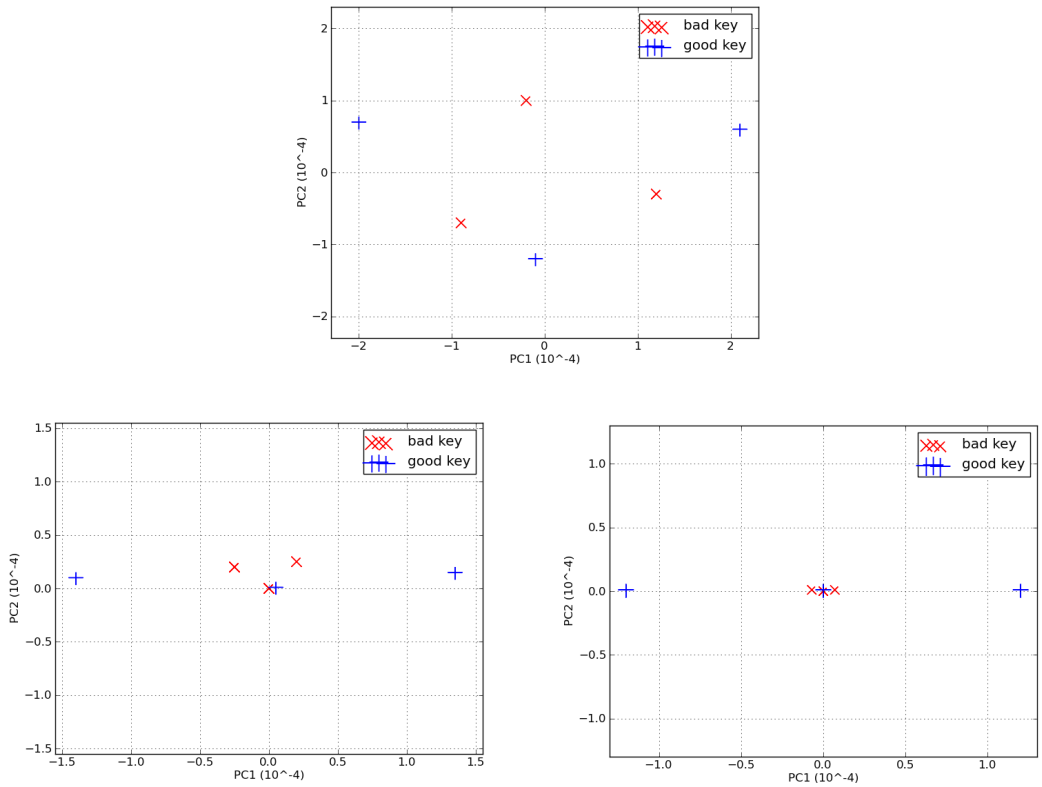


Figure 2.17: References dispersion for different number of traces :(top) 100 traces, (bottom left) 10000 traces, (bottom right) 81000 traces.

$$F_{k_j}^{CS} = \sum_{m=1}^{m'} (\lambda_m \cdot \Big| \ h(W, C^m)\Big|) = \sum_{m=1}^{m'} (\lambda_m \cdot \Big|\sum_{i=1}^{X} (w_i \cdot c_i^m)\Big|)\,, \tag{2.19}$$

where m' is the number of retained principal components, λ_m is the eigenvalue corresponding to PC_m, h is a linear combination function with $C^m = \{c_i^m\}_{i=1}^X$ is the centred coordinate vector of references when projected to PC_m and $W = \{w_i\}_{i=1}^X$ is the associated weight vector. Actually, this indicator takes two factors into consideration: the dispersion and the position of references in the new system coordinate which is composed by the principal components. The dispersion is quantified by the value of the eigenvalues λ_m and the position by the vector of weights W. The best key guess corresponds to the highest value of $F_{k_j}^{CS}$ regarding all key hypotheses ($arg\max(F_{k_j}^{CS})$). One schematic description of FPCA attack is depicted in Fig. 2.18.

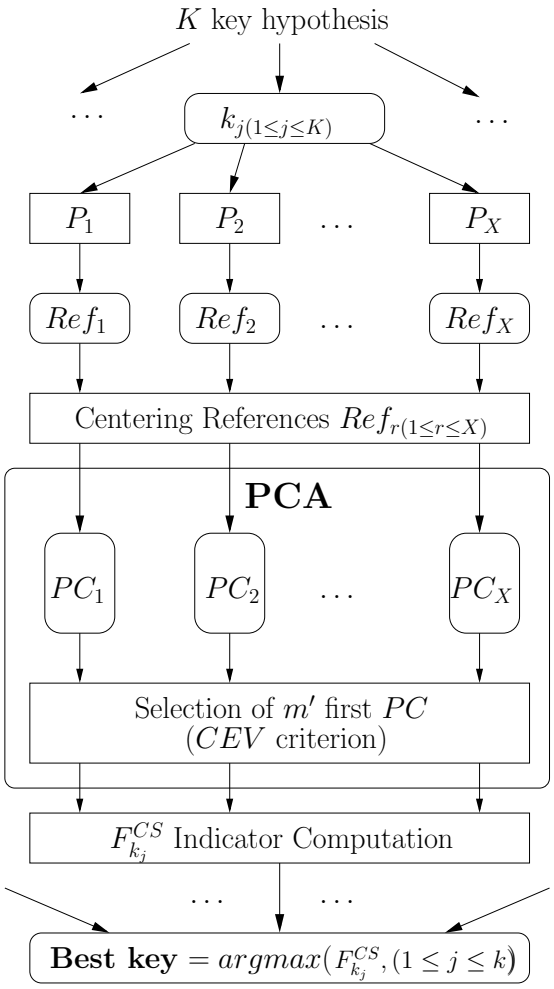


Figure 2.18: FPCA description.

One possible alternative is to consider only the factor of dispersion. This is useful in the case that the position factor is unknown. In fact, the dispersion factor represents a global description of the leakage without the need of more detailed knowledge about the encryption process. The idea is that, if the key guess is correct, PCA applied to the different partitions should be able to explain a big proportion of the variance with only a few components. At the opposite, if the key guess is wrong, Side-channel traces are sorted randomly, and PCA will need more components to explain the same proportion of variance. Thus, a reduced form of the indicator $F_{k_j}^{CS}$ is deduced from Eqn. (2.19) and defined as follows:

$$Red_F_{k_j}^{CS} = \sum_{m=1}^{m'} (\lambda_m) .$$

This indicator can be used for Dual rail Precharge Logic (DPL) architectures, like WDDL [55], which aim at making the activity of the cryptographic process constant independently from the manipulated data. In the context of this study, the idea behind DPL is to force all references to have the same probabilistic features, for all made partitions. However, in real life application, an ideal DPL implementation could not exist. In that sense, the reduced indicator $Red_F_{k_j}^{CS}$ can exploit the leaked information without the knowledge of the position factor.

2.5.4 FPCA on DES Implementations

All our experiments were conducted on real power consumption traces recorded from three different hardware implementations of a DES coprocessor. The first architecture is an unprotected DES. The second and the third ones concern two masking styles: USM [82] and Masked-ROM [82] DES implementation which are configured in an Altera Stratix II FPGA on the SASEBO-B evaluation board provided by the RCIS [128]. Moreover, we note that the length of acquired Side-channel traces covers only the first two rounds for all investigated DES implementations.

In this study, we deal with DoM, DPA, CPA, and VPA attacks. These attacks have shown their efficiency to break cryptographic implementations. Moreover, they are the basis of new derived distinguishers like the Spearman's rank correlation [19]; the correlation concept of Kendall is also of potential interest. Recently, Gierlichs *et al.* have presented an analysis dealing with the comparison of many existing distinguishers related to the aforementioned attacks [51]. The rest of the study deals with experiments on unprotected and masked implementations.

2.5.4.1 FPCA on Unprotected DES

In order to mount a successful FPCA on unprotected DES we fixed the "mean" as CS, as it is shown that such implementation is very vulnerable to differential attacks which are generally based on the "mean" in their calculations. In fact, when attacking the first round of the implementation, the leakage related to the mean is linearly correlated to the power consumption model $HD = \{0, 1, 2, 3, 4\}$. For this purpose, the weight vector W can be defined as follows: $W = \{-2, -1, 0, +1, +2\}$. One other alternative is to consider the probability that one trace belongs to one partition according to one power

consumption model. Hence $W = \{-0.25, -1, 0, +1, +0.25\}$. Results regarding attacks on unprotected DES implementation are depicted in Fig. 2.19 and Fig. 2.20. Indeed, the First-order success rate shows a superior performance of FPCA attack. This can be explained by the fact that DoM, CPA, and DPA are implicitly taking into account only the position factor relatively to our proposed attack. According to Fig. 2.20, FPCA needs around 160 traces to perform a successful attack. Unsurprisingly, the Guessing entropy metric depicted in Fig. 2.19 is in accordance with the First-order success rate results. One note is that FPCA is able to distinguish the *secret key* at an early stage. In fact, only 30 traces are required to get the *secret key* in the top ten of the key hypotheses averaged rank list.

2.5.4.2 FPCA on Masked DES

In the context of this study, an ideal masking implementation is one for which all references, for all made partitions, are the same when using the mean as CS. Formally, the probability distributions, that are related to leakage partitions, have all the same *expected value* (or mean). However, it has been proved that masking technique is still susceptible to first-order SCA as long as glitches problem remains not completely resolved [88]. For instance, authors in [82], have shown that one masked structure so-called "Universal Substitution boxes with Masking" (USM) is vulnerable to DPA. Moreover, masked implementations are not resistant against new variants of SCA like VPA: in fact it is shown that a full-fledged masked DES implementation using a ROM (Masked-ROM) is breakable by VPA, in spite of its high resistance against first-order attacks. In what follows, we use the same power consumption model as described in [82] to perform the FPCA on USM and Masked-ROM DES implementations.

First, in order to make a fair evaluation for our attack on USM DES structure we kept the "mean" as CS and we classified traces into five partitions for each key hypothesis k_j. For reasons of clarity, comparison is made between FPCA and DPA for which we noticed the best performances with regards to DoM and CPA. Results are deduced from Fig. 2.21 and Fig. 2.22. Obviously, according to the First-order success rate metric shown in Fig. 2.22, FPCA is more efficient than DPA. Indeed, 15000 traces are needed for DPA to achieve a rate of 80%. Whereas, for the same rate, FPCA attack requires only 10000 traces. The Guessing entropy metric, is quite equivalent for both attacks. Second, we targeted a Masked-ROM DES implementation. For this purpose, we chose the variance as CS, as it has shown that such implementation is sensitive to VPA, which is actually based on variance analysis [78; 82]. Fig. 2.23 shows that the

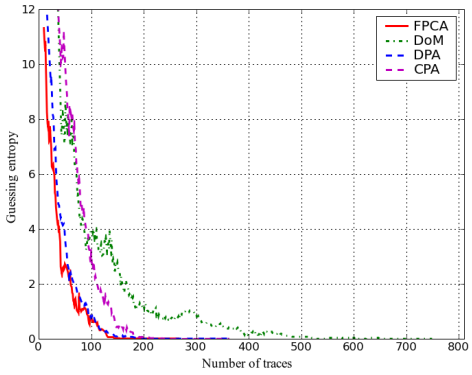


Figure 2.19: Unprotected DES Guessing entropy metric.

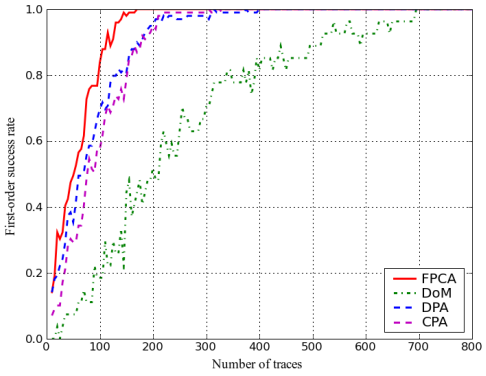


Figure 2.20: Unprotected DES First-order success rate metric.

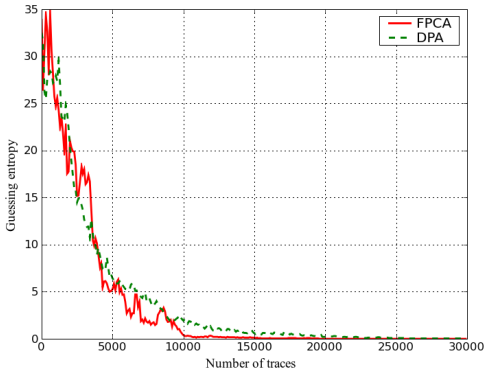


Figure 2.21: USM DES Guessing entropy metric.

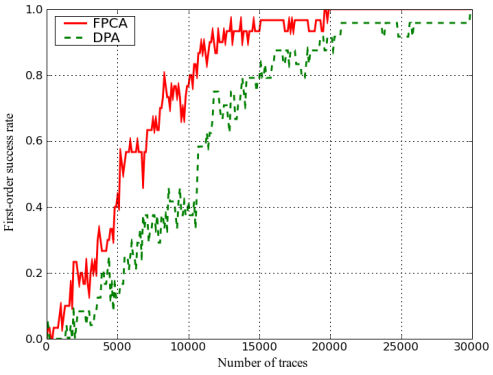


Figure 2.22: USM DES First-order success rate metric.

Guessing entropy curve, related to FPCA, approaches the best rank (*i.e.* the lowest rank) more rapidly than VPA. Moreover, the First-order success rate metric depicted in Fig. 2.24 reveals noticeable differences between both attacks. Clearly, FPCA has more chance to find the secret key more rapidly than VPA.

2.5.5 Conclusion

In this part of Chapter, we have proposed a new variant of SCA called FPCA, which is mainly based on Principal Components Analysis (PCA), the powerful multivariate data analytic tool. We have shown the efficiency of FPCA on unprotected as well as protected cryptographic implementations. Moreover, we have empirically shown

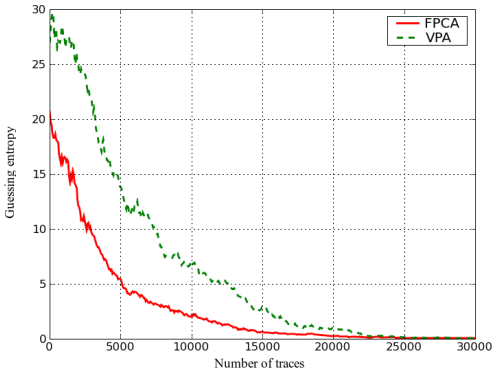


Figure 2.23: Masked-ROM Guessing entropy metric.

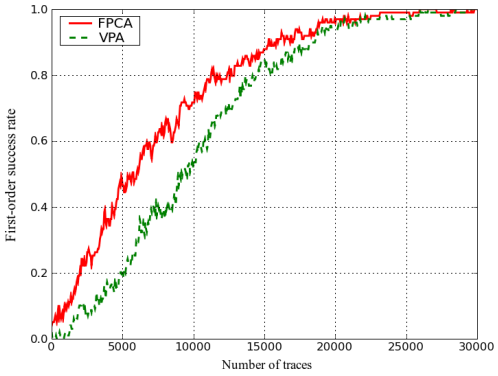


Figure 2.24: Masked-ROM First-order success rate metric.

its superior performance with regards to existing attacks (DoM, DPA, CPA, VPA). Besides, the originality of FPCA is to use PCA no more as a pre-processing tool, as used for Template attacks, but as a genuine distinguisher. In this insight, in the next Section 2.6 we propose to use Wavelets analysis, that was initially used to pre-process traces, to enhance the quality of distinguishers; and therefore improve Side-channel attacks.

2.6 Wavelets Transform based Side-Channel Attacks

2.6.1 Introduction

Basically, in the open literature, Side-channels are preferred to be performed in the time domain than in the frequency domain [86; 91]. Indeed, it is often reported that analysing a power or electromagnetic signal in the time domain is more efficient [89] than analysing it for its frequency content, based on the Fourier transform. The reason is that the Fourier transform is able to analyse the frequency content of a signal; whereas the time/phase information is not used. Actually, the time (phase) information and the frequency information are separated when performing a Fourier analysis. An alternative to this issue is the Short Time Fourier Transform (STFT) that computes the Fourier transform over different parts of the signal, called windows. The computation of the STFT provides the time-frequency information content of the analysed signal with a constant frequency and time resolution, because of the fixed window length. This is unlikely the most appropriate resolution. Indeed, when analysing low frequencies, a proper frequency resolution is often required. Whereas, for high frequencies, the time resolution is more important. An alternative tool with some attractive properties is the *Wavelet transform*. Although wavelets theory has been successfully applied in many applications in science and engineering, it has been rarely invoked in the SCA context. Actually, wavelets have only been used in two occasions: first, they are used as preliminary analysis to align Side-channel traces [110], and second used to estimate the probability density function of the leaked information [117]. Thus, wavelets have been used both as a pre-processing and an estimation tool. The originality of our work is to use Wavelet transform in the very core of the attack. The proposed wavelets based Side-channel attack takes benefits of both time and frequency domains. But more importantly, this analysis is **generic** as it can be seen as a plug-in to improve most of existing attacks.

2.6.2 An Understanding of the Multiresolution Analysis

In practice, signals acquired are usually analysed in the time domain. The information encompassed by signals can be described by different means or representations. Actually, thanks to these representations, more details about the signal, such as the frequency contents, can be emphasized very nicely in order to highlight the hidden dynamics related to the cryptographic process. The most commonly used representation

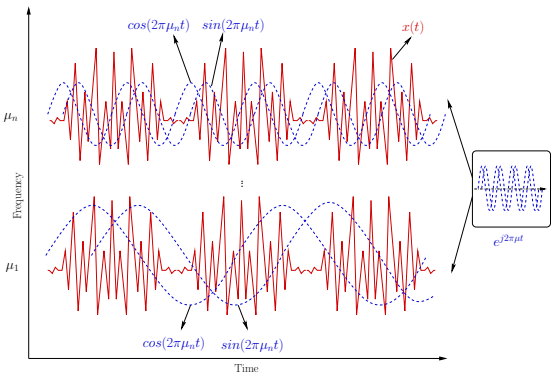


Figure 2.25: Fourier transform illustration.

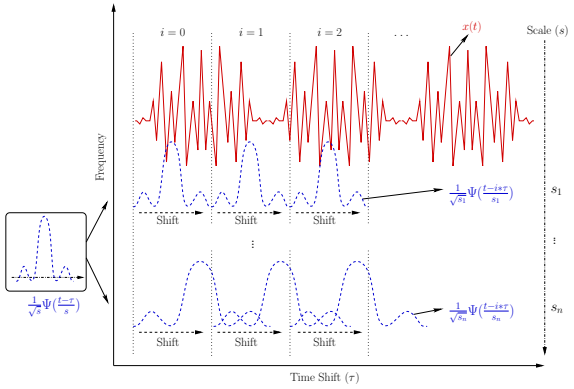


Figure 2.26: Wavelet transform illustration.

used to analyse a time signal for its frequency contents is the Fourier transform and its alternative the Short Time Fourier Transform (STFT). Understanding the Fourier transform and the STFT is necessary to understand the basics of the multiresolution analysis, which is often known as *Wavelet transform*.

2.6.2.1 Fourier Transform Overview

The Fourier transform is a mathematical representation that decomposes a function exactly into many components, each of which has a precise frequency. From the mathematical point of view, any periodic function $f(t)$, with period $2p$, that is $f(t) = f(t+2p) = ... = f(t+2np)$ for $n \in \mathbb{N}$, can be expressed as a linear combination of all *cosine* and *sine* functions, which have the same period:

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{+\infty}(a_n cos(\frac{n\pi t}{p}) + b_n sin(\frac{n\pi t}{p})) \ , \tag{2.20}$$

where

$$\begin{aligned} a_n &= \frac{1}{p}\int_{-p}^{p} f(t)cos(\frac{n\pi t}{p})dt \ ; \ n \in \mathbb{N} \ , \\ b_n &= \frac{1}{p}\int_{-p}^{p} f(t)sin(\frac{n\pi t}{p})dt \ ; \ n \in \mathbb{N} \ . \end{aligned} \tag{2.21}$$

These series of sines and cosines are called *Fourier Series*. Note that $cos(\frac{n\pi t}{p})$ or $sin(\frac{n\pi t}{p})$ is a periodic function, which period T_n is determined by the relation that

when t is increased by T_n, the function returns to its previous value,

$$cos(\frac{n\pi}{p}(t+T_n)) = cos(\frac{n\pi t}{p} + \frac{n\pi T_n}{p}) = cos(\frac{n\pi t}{p}) \ . \tag{2.22}$$

Thus,

$$\frac{n\pi}{p}T_n = 2\pi \ , \ T_n = \frac{2p}{n} \ . \tag{2.23}$$

The frequency μ is defined as the number of oscillations in one second. Therefore, each of them is associated with a frequency μ_n,

$$\mu_n = \frac{1}{T_n} = \frac{n}{2p} \ . \tag{2.24}$$

Often, the angular frequency, defined as $\omega_n = \frac{n\pi}{p} = 2\pi\mu_n$, is used to simplify the writing. The *Fourier Series* can be translated to complex numbers as a first step before formalizing the Fourier transform. Actually, it can be shown that the *Fourier Series* of a function repeating itself in the interval of $2p$ can be written as:

$$f(t) = \sum_{n=-\infty}^{+\infty} c_n e^{i\frac{n\pi}{p}t} \ , \ c_n = \frac{1}{2p}\int_{-p}^{p} f(t)e^{-i\frac{n\pi}{p}t}dt \ . \tag{2.25}$$

Hence

$$f(t) = \sum_{n=-\infty}^{+\infty} [\frac{1}{2p}\int_{-p}^{p} f(t)e^{-i\frac{n\pi}{p}t}dt]e^{i\frac{n\pi}{p}t} \ . \tag{2.26}$$

Now, if we denote $\delta\omega = \omega_{n+1} - \omega_n = \frac{\pi}{p}$, then the series can be expressed as:

$$f(t) = \sum_{n=-\infty}^{+\infty} [\frac{1}{2\pi}\int_{-p}^{p} f(t)e^{-i\omega_n t}dt]e^{i\omega_n t}\delta\omega = \sum_{n=-\infty}^{+\infty} \frac{1}{2\pi}\hat{f}_p(\omega_n)e^{i\omega_n t}\delta\omega \ , \tag{2.27}$$

where $\hat{f}_p(\omega_n) = \int_{-p}^{p} f(t)e^{-i\omega_n t}dt$.

"The Fourier transform may be regarded as the formal limit of the Fourier Series as the period tends to infinity" [115]. Indeed, if we let $p \to +\infty$, then $\delta\omega \to 0$ and ω_n becomes a continuous variable. Hence, we have:

$$\hat{f}(\omega) = \lim_{p\to+\infty} \hat{f}_p(\omega_n) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t}dt \ . \tag{2.28}$$

Besides, the original signal $f(t)$ can be reconstructed using the inverse Fourier transform

as follows:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{i\omega t} dt \ . \tag{2.29}$$

In practice, signals acquired experimentally are not continuous in time, but sampled as discrete time intervals δT. Besides, acquired signal's length is finite with a total acquisition time T, divided into $\frac{T}{\delta T}$ intervals. In this case, the frequency analysis is conducted through the *Discrete Fourier Transform* (DFT). An illustration of Fourier transform is shown in Fig. 2.25. Generally, the Fourier transform is useful at providing the frequency content that can not be easily detected in the time domain. However, the temporal structure of the analysed signal is not revealed; and therefore such transforms are not convenient to be used for rapidly time varying signals. Actually, using Fourier transform does not allow a time variation analysis of the signal's frequency contents. This may limit the merit of Fourier transform specially when both time and frequency information are required.

2.6.2.2 Short Fourier Transform (STFT) Overview

In order to overcome the limitations of the Fourier transform, a kind of time localised Fourier transform can be introduced. This concept, which has been initially proposed by D.Gabor in [44], is referred to as *Short Time Fourier Transform* (STFT). The STFT is capable to retrieve both frequency and time information from a signal. Technically, STFT employs a sliding window function $g(t)$ that is centered at time τ; and can be expressed by the following equation:

$$f_{STFT}(\tau, \mu) = \int_{-\infty}^{+\infty} f(t) g^*(t - \tau) e^{-i2\pi\mu t} dt \ , \tag{2.30}$$

where $*$ denotes the complex conjugated operator. When the window is moved by τ, a time-localised Fourier transform is performed on the original signal $f(t)$ within the window. This way, the STFT decomposes a time domain signal into a two dimensions time-frequency representation. Such time-frequency representation provides some information about if some frequency is continuously present over the whole signal or only at a specific time interval (just a part of the signal). However, using a sliding window with fixed length results in a new problem. Actually, the STFT analysis is critically dependent on the chosen window $g(t)$. Theoretically, it is not possible to know exactly what frequencies occur at a specific time, only some frequencies (an interval of frequencies) can be detected. In other words, it is not possible to get both a good time resolution and a good frequency resolution with STFT. In fact, on the one hand, a

short window, which is convenient for high frequencies detection, provides a good time resolution, but several frequencies are not revealed. On the other hand, a long window, which is convenient for low frequencies detection, provides an inferior time resolution, but a better frequency resolution.

2.6.2.3 Wavelet Transform

We have seen that the STFT is not sufficient to describe, with accuracy, both the time and the frequency content of a signal acquired. Recently, a powerful tool called *Wavelets analysis*, has been introduced to overcome the limitation of the STFT. In contrast to STFT, where the sliding window size is fixed, the Wavelet transform uses variable window sizes (or resolutions) in analysing the frequency content of a signal. The Wavelet analysis correlates the original signal $f(t)$ with a predetermined functions obtained from the scaling (*i.e.* dilation and compression) and the shift (*i.e.* translation along the time samples) of a wavelet function ψ, referred to as *the mother wavelet*. In comparison with Fourier transform, Wavelet transform offers more flexibility as no restrictions are required when choosing the analysing function. In other words, there is no need of using only sines and cosines forms as for Fourier transform. The similarity between the original signal and the wavelet function is calculated separately for different time intervals, producing a two dimensional representation. Basically, when dealing with Wavelet analysis, two types of transforms can be used: the *Continuous Wavelet Transform (CWT)* and the *Discrete Wavelet Transform (DWT)*.

2.6.2.4 Continuous Wavelet Transform (CWT)

As defined in statistics books, the Continuous Wavelet transform (CWT) is *"the sum over all time of scaled and shifted versions of the wavelet function ψ (or mother wavelet)"*. Indeed, ψ is a more complex function (relatively to sines and cosines in Fourier transform) that aims at detecting more details about the patterns of analysed signal. Examples of analytical wavelets are the Paul, Morlet, b-spline, and Shannon mother wavelets, which are defined by the following equations [150] respectively:

$$\psi_{Paul}(x) = \frac{2^n n!(1-ix)^{-(n+1)}}{2\pi\sqrt{\frac{(2n)!}{2}}} \ . \tag{2.31}$$

$$\psi_{Morlet}(x) = \frac{1}{(f_b^2\pi)^{1/4}} e^{2\pi i f_c x} e\frac{-x^2}{2f_b^2} \ . \tag{2.32}$$

$$\psi_{b-spline}(x) = \sqrt{f_b} e^{(2\pi i f_c x)} [sin_c(\frac{f_b x}{m})]^m. \tag{2.33}$$

$$\psi_{Shannon}(x) = \sqrt{f_b} e^{(2\pi i f_c x)} sin_c(f_b x). \tag{2.34}$$

Where n is the order of the Paul mother wavelet, f_b is the variance of the analysed window, f_c is the mother wavelet central frequency and m is an integer order parameter (where $m > 1$). The CWT, when applied on the original signal $f(t)$, is expressed as:

$$CWT_f(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} f(t) \psi^*(\frac{t-\tau}{s}) dt \ , \tag{2.35}$$

where $*$ denotes the complex conjugated operator. The calculation of the CWT over a signal results in many *coefficients*, which are functions of the translation parameter τ and the scale parameter s. In fact, τ, is proportional to time information. It indicates the time location of the wavelet. The variation of τ leads to shifting the wavelet over the signal. The scale s is inversely proportional to the frequency information. The variation of s modifies principally the window length. The larger scales, the more low frequencies detected; and the more the general shape of the signal revealed. Conversely, the smaller scales, the more high frequencies detected; and the more the finer details of the signal revealed. Moreover, the signal energy remains constant at every scale as it is normalized by $\frac{1}{\sqrt{|s|}}$. Although the high accuracy provided by the CWT in analysing a signal, the computation of CWT is redundant and very time consuming. Usually the calculation of the CWT is performed by taking discrete values for the scaling parameter s and the translation parameter τ. An illustration of CWT is shown in Fig. 2.26.

2.6.2.5 Discrete Wavelet Transform (DWT)

In practice, signals acquired experimentally are not continuous in time, but sampled as discrete time intervals. Previously, we have seen that the CWT performs a time-frequency resolution (or multiresolution) by scaling and shifting a wavelet function. Recently, it has been shown that such analysis can actually be performed using multiresolution filter banks and wavelet functions, resulting in the *Discrete Wavelet Transform* (DWT). We note that the DWT is not the discretized version of the CWT, which just uses a discretized version of the scale and the translation parameters. The DWT provides the information necessary to reliably analyse the content of a signal acquired; and more importantly it is much faster than CWT calculations. One level DWT is basically composed of what we call *wavelet filters*, which involve two basic concepts: the filter banks and the down- and up-sampling operations. The filter banks aim at changing

the resolution by separating the sources of the signal into frequency bands. Actually, a discrete time signal is first filtered by the filters L and H which separate the frequency content of the analysed signal in frequency bands of equal length. The filters L and H are thus respectively a low-pass and a high-pass filter. Therefore, the signal is effectively decomposed into two sub-signals called the *details* (or detail wavelet coefficients) and the *approximations* (or approximation wavelet coefficients), respectively. The approximations correspond to low frequencies and the details to high frequencies. Note that the output sub-signals each contains half the frequency content, but the same amount of samples as the original signal. In other words, the two sub-signals together contain the same frequency content as the original signal, however the amount of samples is multiplied by two (doubling). At this point, a down-sampling of factor two is applied to each sub-signal, which indeed doubles the scale (*i.e.* make the analysing window larger), doubles the frequency resolution; and therefore avoid redundancy. This way, the resolution and the scale have been changed in a manner to increase the frequency resolution and reduce the time resolution. In the second level of DWT, the approximation sub-signal is used as the original signal and put through a wavelet filter (*i.e.* filter bank plus down-sampling), until reaching the required level of decomposition. For a p-level decomposition, the range of frequency content included by the approximations and the details, denoted by f_{Approx} and f_{Det} respectively, can be determined as follows:

$$f_{Approx} = [0, \frac{f_s}{2^{p+1}}] \ , \ f_{Det} = [\frac{f_s}{2^{p+1}}, \frac{f_s}{2^p}] \ , \tag{2.36}$$

where f_s is the sampling frequency of the original signal. Eventually, the original signal can be represented by the final approximation, related to the last level of decomposition, and the accumulated details corresponding to all levels. The process can be expanded to an arbitrary level, depending on the desired resolution. An illustration of a 3-levels DWT is shown in Fig. 2.27. Note that the relation between CWT and DWT is similar. In fact, on one hand, wavelets in the CWT behave as a band-pass filter when correlating the mother wavelet function with the original signal; on the other hand, the DWT process composed of low-pass filter, high-pass filter and down-sampling behaves also as a band-pass filter. Now, for the reconstruction of the original signal, the same mechanism is used for the transformation. Actually, the reconstruction is possible thanks to the final approximation and the accumulated details. The obtained sub-signals are first upsampled and then passed through filter banks, which are related to the initial filter banks L and H. For more technical details about the reconstruction, we refer the reader to [27].

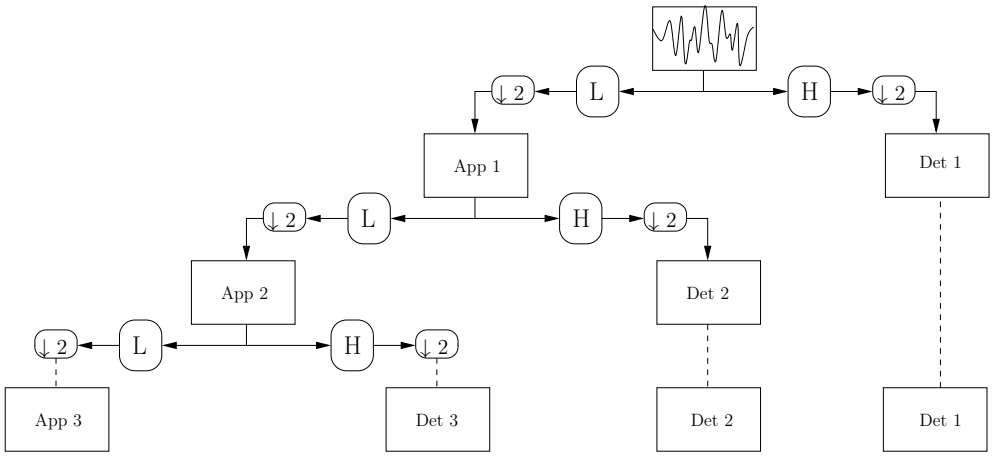


Figure 2.27: An illustration of 3-levels DWT decomposition.

2.6.3 Wavelets for Secret Key Recovery

In SCA, the interest is to find some ways to accelerate the attack, taking into account that the scarce resource is the number of traces acquired. As stated before, the importance of wavelets relies on separating the different sources (*i.e.* an electrical activity with different frequencies content) composing the signal acquired. We recall that the signal content can be represented differently thanks to the concatenation of the approximations of the last level of decomposition with accumulated details. From the SCA perspective, such representation is very helpful as the main goal of SCA is to separate and extract the sources related to the cryptographic process and responsible for the manipulation of the secret key. Our proposition is to perform the Side-channel analysis directly on the wavelet coefficients: the approximations and the details. In other words, the leakage is not anymore represented by the time samples of the acquired signals but simply by the value of the wavelet coefficients. This way, in one sense, we may improve the efficiency of Side-channel distinguishers. In another sense, we avoid the problem of signal's reconstruction. Actually, wavelets theory says that a loss of information could happen only when reconstructing the signal (*i.e.* the transition from wavelet coefficients to temporal domain).

2.6.3.1 Wavelets based CPA

In our first experiment, we have considered a CPA performed on real unprotected DES traces. These traces were averaged by the oscilloscope to reduce the amount of noise.

This has a pure practical flavour for us, as the idea behind the experiment is to show that wavelets are not acting as a noise filter but as a sources separation tool. More precisely, three analyses have been involved: CPA on the original traces (*orig*), CPA on the first-level of wavelet decomposition, *i.e.* the approximations and the details (*AppDet*) and a CPA on only the approximation coefficients (*App*). In what follows, we used *Daubechies wavelet* [103] as mother wavelet function. The First-order success rate, which is depicted in Fig. 2.28, shows that analysing the traces using wavelets is clearly more efficient than the basic analysis in the time domain. Actually, the First-order success rate is converging faster towards the maximal rate for *AppDet* and *App* than for *orig*. For instance, to reach a Success rate of 90%, we need around 150 traces for *orig*, and only 75 traces for *AppDet* (*i.e.* a gain of 50%). Besides, we noticed that for low levels of decomposition, like in this example (first level used), the useful information regarding unprotected implementations is essentially located within the approximations; in contrast to the details that represent the noise source component of the Side-channel measurement. Nonetheless, during our experiments, we have remarked that *AppDet* often performs better then *App* especially when the implementation is protected. Therefore, the evaluator is required to conduct his analysis on *AppDet*, specifically when the deployed countermeasures are unknown. In the following experiments, all coefficients (*i.e.* *AppDet*) are considered. In Fig. 2.29, the Guessing entropy, unsurprisingly confirms the results found for the First-order success rate. In fact, the rank of the secret key when performing an ordinary CPA takes more time to converge to the lowest and best rank, compared to *AppDet* and *App* analyses.



Figure 2.28: CPA First-order success rate.

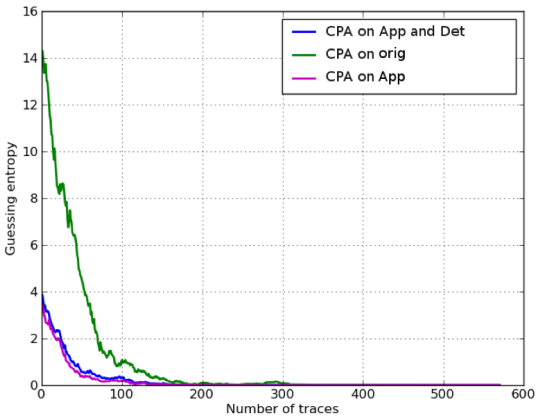


Figure 2.29: CPA Guessing entropy.

Generally, from the evaluation perspective, the evaluator has at his disposal power-

ful tools, usually known as profiling analyses, such as *Template analysis* and *Stochastic models* [133], which mainly aim at revealing the finer details about the leakage; and therefore allowing the evaluator to reliably assess the robustness of the analysed cryptographic implementation.

2.6.3.2 Wavelets based Template attacks

In our second experiment, we are interested in Principal Components Analyses (PCA) based template analysis and examine the effect of its combination with Wavelet transforms on SCA. We note that, in this experiment, the same DES traces used previously were again involved. However, a Gaussian noise was added to these traces. Actually, one other goal of this experiment is to highlight the efficiency of the attack when increasing the level of wavelets decomposition. In the open literature of wavelet analysis, the combination between PCA and Wavelet transform has been recently proposed in [17]; and has turned out to be efficient in many engineering fields [10; 77]. The main idea behind combining PCA with wavelets is to remove redundancy from wavelet coefficients; and therefore keep only de-correlated ones. According to Fig. 2.30 and Fig. 2.31, first we notice that wavelet analysis outperforms, in terms of Success rate and Guessing entropy, the standard analysis, *i.e.* when performed on the original trace; and this for all levels of decomposition. Moreover, the performance of the analysis is getting better when the level is increased. In principle, the more scale levels are used, the higher the performance is. Nonetheless, we can not endlessly centralize the energy. Hence, a limit scale wavelet decomposition level can be determined. Beyond this level, performance remains constant. Besides, more computations and time complexity are required when increasing the level of decomposition. Therefore, a proper scale level should be selected based on some points such as the amount of noise affecting the traces and the frequency sampling used for the acquisition. When performing a Template analysis, the selection of a proper level of decomposition can be easily determined thanks to the clone device used for the profiling.

2.6.3.3 Wavelets vs FFT based Template Attacks

Our third experiment consists in comparing the efficiency of DWT to FFT based Templates analysis. As stated before, Side-channel attacks based on the frequency domain (*e.g.* FFT) are usually less performant than those based on the time domain [86; 91]. In this experiment, the averaged version of DES traces was used. According to Fig. 2.32 and Fig. 2.33, DWT based template attack is clearly more rapid than both FFT and

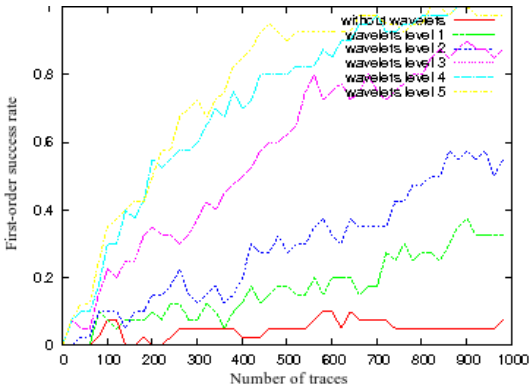


Figure 2.30: Template attack First-order success rate.

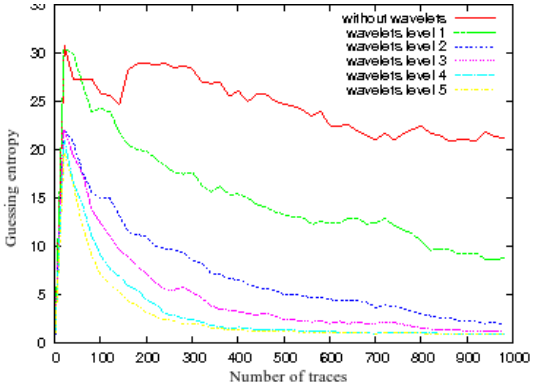


Figure 2.31: Template attack Guessing entropy.

original time domain attack (or basic attack, termed by "REAL" in figures), in achieving a First-order success rate of 100%. We note that these results concern one Sbox of the DES implementation. Nonetheless, during our tests, we noticed that approximately same results (*i.e.* same First-order success rate and Guessing entropy evolution) were observed for the rest of Sboxes.

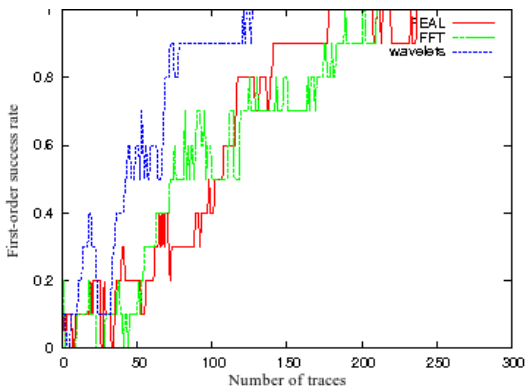


Figure 2.32: DWT vs FFT based Template attack: First-order success rate.

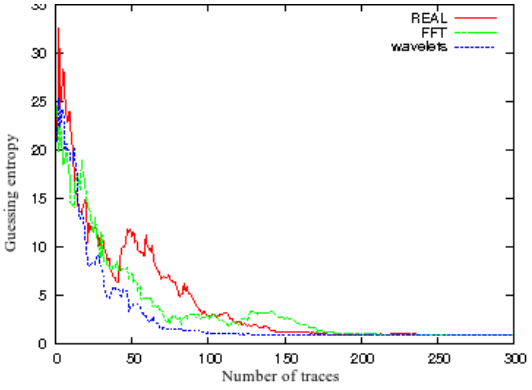


Figure 2.33: DWT vs FFT based Template attack: Guessing entropy.

2.6.4 Conclusion

In this part of Chapter, we have presented a new way to use Wavelet transform in the very core of the attack. The proposed Wavelets based Side-channel attack takes benefits of both time and frequency domains. But more importantly, this analysis is **generic** as it can be seen as a plug-in to improve existing SCAs.

In Chapter 3, we will provide the evaluator with more innovative applications of Wavelets transform to be applied in the Side-channel analysis field.

Chapter 3

Side-Channel Signal Processing

3.1 Introduction & Contributions

Signal Processing theory has been always an important topic in computer science. Obviously, Signal Processing is nowadays used everywhere to extract and analyse information from signals. In the modern litterature of Signal Processing, signals are digitally acquired by oscilloscopes and calculations are usually made by computers or integrated circuits. Thus, we often employ the term *DSP* for *Digital Signal Processing*. From the Side-channel security evaluation perspective, Signal Processing can be considered as the backbone of the evaluation; as the main idea behind Side-channel analysis is to best detect and extract the secret information from signals. Indeed, in the Side-channel context, signals, which are often called *traces* or *measurements*, are digitally acquired from an electronic circuit while performing a cryptographic operation. This Chapter aims at proposing, in a methodological manner, new Signal Processing techniques and practices to provide Side-channel evaluator with the best working conditions. Basically, when assessing the robustness of a secure implementation, the evaluator might be faced with two Signal Processing problems: the noise problem affecting Side-channel traces and the de-synchronization (or misalignment) of these traces.

3.1.1 The Noise Problem

In practice, whenever one tries to measure a signal, there is always some form of noise to be accounted for. Basically, noise can be defined as "unwanted disturbances superposed upon a useful signal that tends to obscure its information content" [154]. Indeed, noise affects negatively the quality of electrical signals. In the Side-channel context, only few techniques have been proposed in order to improve the quality of power consumption signals [76; 110]. In this part of thesis, we propose new solutions for denoising Side-

channel measurements; and therefore we markedly improve working conditions of the evaluation aspects. Techniques for noise reduction are particularly important for the evaluator because Side-channel analyses are very sensitive to the magnitude of these signals acquired to retrieve the secret information. Averaging traces is the traditional method used to reduce noise. Nonetheless, it is necessary to investigate more advanced techniques and strategies, that should follow the advances being made in the Side-channel field.

3.1.1.1 Our contributions

In this Chapter, we first propose an algorithmic solution, based on the well known Kalman filtering theory [156]. Moreover, we make the comparison with *the High Order Statistics* based noise filtering technique (HOS) [75; 76], one of the first tools proposed in the context of denoising Side-channel traces. Second, we investigate the Expectation-Maximization algorithm (EM) to get a better estimation of the Kalman filter (KF) parameters. Third, we propose the use of *the Wavelet transform* to best pre-process traces. Indeed, using Wavelet analysis for Side-channel noise filtering have been first proposed by Pelletier [110]. In this thesis, we develop a new algorithm to improve the existing technique of Pelletier; and propose other applications of Wavelet transform in pre-processing acquired signals, such as the compression of traces and the detection of cryptographic patterns. Eventually, we give more in depth view of adapting the basics of *Electromagnetic Shielding* theory, which particularly aims at decreasing the contribution of external noise sources, to the Side-channel context.

3.1.2 The De-synchronization Problem

From the Side-channel evaluation perspective, the alignment of acquired traces is of a great concern since deployed analyses are very sensitive to the magnitude of these traces. In the real life context, it is almost impossible to perfectly get aligned traces due to many factors. For instance, a frequent situation is that the trigger signal, which is precisely synchronized with the cryptographic process, is removed by the designer for security reasons. However, even if the access to a signal indicating the start of the cryptographic process is possible, a jitter is often observed. Therefore, in both situations, secret information can be lost due to errors induced by the displacement of traces. In some other cases, the misalignment results from some implemented counter-measures, such as instruction shuffling [118] or random delay instruction [34]. Basically, these countermeasures aim at hiding the instant where the cryptographic process is be-

ing performed. For instance, one may implement additional and useless operations or varies the internal clock frequency responsible for controlling the cryptographic process. Generally, any factors responsible for the temporal misalignment of traces, are usually connected to either the acquisition environment (*e.g* oscilloscope resolution, nearby noise, *etc.*) or to the cryptographic architecture itself. The most traditional solutions to get round this problem is to average the acquired traces or acquire as many traces as possible. However, we assume that these solutions are not suitable for overall misalignment cases. Actually, averaging is not possible for masked implementations since the mask is updated from an acquisition to another. Besides, the evaluator might be limited by the number of traces that he can acquire. One other solution, namely Differential Frequency Analysis [151], has been proposed to overcome the temporal misalignment problem. Indeed, this analysis, which aims at transposing the Side-channel analysis to the frequency domain, is principally based on the time shifting property of Discrete Fourier Transform (DFT) for periodic signals. Basically, this property states that a shift in time is similar to a linear phase shift in frequency. The frequency content is unchanged in a time shift, since it depends only on the shape of a signal. Only the phase content will be affected. From the view point of the evaluator, first it is important to get rid off the misalignment so as to make its evaluation in the best conditions. Second, it is better to know if the countermeasure used to misalign traces can be broken by an attacker.

3.1.2.1 Our contribution

In this Chapter, we give particular attention to investigating new algorithmic solutions to the common problem of aligning Side-channel traces. Indeed, first we highlight the importance of aligning SCA traces and survey existing techniques to get round the problem of misalignment. Second, we put forward an innovative re-synchronization algorithm, called RM, and show its efficiency compared to existing techniques.

3.2 Side-Channel Filtering & Patterns Detection

3.2.1 Kalman Noise Filtering

Kalman filtering has been the subject of extensive research and application such as mobile robotics [41], radar tracker [26], weather forecasting [23], satellite navigation systems and economics [31]. For example, in biomedical fields the Kalman filter is used to denoise the electrocardiogram (ECG) signals [16] that are obtained by a non intrusive method as well for power analysis. In the open litterature, it has been often shown the efficiency of Kalman filtering specifically when dealing with random signals, known as *white noises*. Basically, when performing a Side-channel analysis, four different types of noise, that can usually be approximated by a Gaussian distribution, can be distinguished:

- **Extrinsic noise** which is induced by the external source,
- **Intrinsic noise** which is induced by the device under test itself,
- **Quantization noise** which is induced by the activity of the analog-to-digital converter used to sample the power signals,
- **Algorithmic noise** which is induced by the activity of the cryptographic process.

In what follows, we give more details about the theoretical background necessary to understand the basics of Kalman filtering; and empirically show its efficiency in denoising Side-channel measurements.

3.2.1.1 Kalman Filter Model

State Space Model Overview

The state space model is a widely used notion in Signal Processing domain. This notion offers a mathematical description of a system behaviour. Most of physical processes, such as sinusoidal radio-frequency carrier signal, can be approximated as a linear systems [137]. In the general case, whatever the mechanism of the system is, the linear approach, which can be represented by a series of samples defined in discrete points of time (or time samples), can serve as a first approximation. Let $\mathbf{y} = Y_L = (y_0, \ldots, y_\ell, \ldots, y_L)$ being the observation vector taken by one measurement,

and $\mathbf{x} = (x_0, \ldots, x_\ell, \ldots, x_L)$ the unknown vector that we aim to estimate. The model under consideration can be described by the following two equations: a state equation Eqn. 3.1 (or the process model equation) that defines the evolution of the process through time and a measurement equation Eqn. 3.2 (or the observation equation) that describes how the hidden state (or internal state) is observed.

$$x_\ell = Ax_{\ell-1} + Bu_\ell + w_\ell \ , \tag{3.1}$$

$$y_\ell = Hx_\ell + v_\ell \ , \ \ell > 0 \tag{3.2}$$

where x_ℓ is the true state vector, y_ℓ is the measurement vector of the true state vector, u_ℓ is the optional input control vector, w_ℓ is the process noise vector, v_ℓ is the measurement noise vector, H is the observation matrix model which maps the true state vector into the measurement vector, A the state transition matrix model which is applied to the previous true state vector and B is the optional control input matrix model which is applied to the control vector u_l. An illustration of the Kalman process is shown in Fig. 3.1.

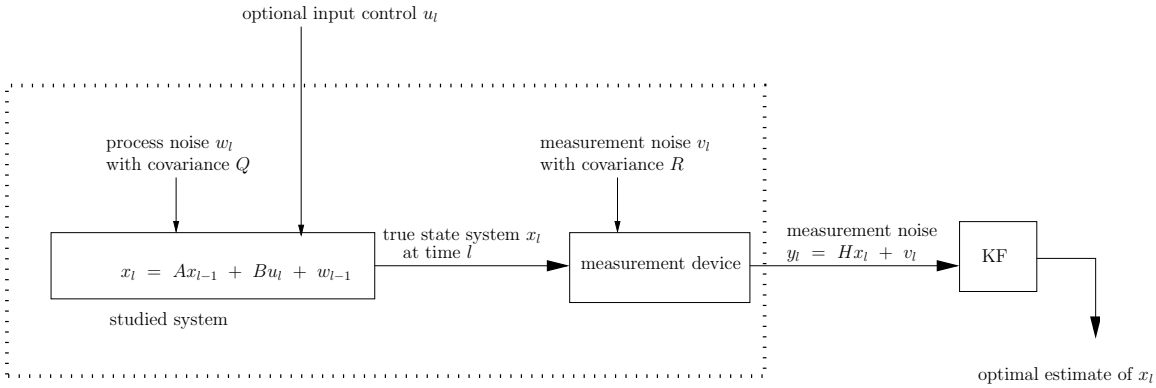


Figure 3.1: Kalman process description.

Kalman Filter Equations

The state space realization discussed above can also be called Discrete time Kalman model if we consider that w_ℓ and v_ℓ of covariance $Q = \sigma_w^2$ and $R = \sigma_v^2$ respectively are zero mean Gaussian processes, *i.e.* their amplitude can be modelled as a normal distribution and $\mathbb{E}(w) = \mathbb{E}(v) = 0$, where $\mathbb{E}$ is the mathematical expectation operator

. The noise vectors $(v_0, \ldots, v_L)$ and $(w_1, \ldots, w_L)$ are assumed to be uncorrelated, *i.e.* mutually independent. Note that, in the real world, this model can not be fitted perfectly. In this case, the KF is still functional, but it may not lead to optimal results [139], particularly if the filter parameters are not accurately estimated. In what follows we denote by $\hat{x}_{\ell|\ell-1}$ the *a priori* state vector estimate at time ℓ given knowledge of the process up to time $\ell - 1$, $\hat{x}_{\ell|\ell}$ the *a posteriori* state vector estimate at time ℓ given measurement y_ℓ, $P_{\ell|\ell-1}$ the covariance matrix of *a priori* estimate error, $P_{\ell|\ell}$ the covariance matrix of *a posteriori* estimate error, I the identity matrix and A^T the transpose of the matrix A. The Kalman filtering technique is depicted in Fig. 3.2. The KF estimates a process using a prediction-correction form. In fact, the filter estimates the process state at some time then gets feedback in the form of noisy measurement. This way, the KF has two distinct phases: Prediction and Correction.

Prediction phase equations:

$$\hat{x}_{\ell|\ell-1} = A\hat{x}_{\ell-1|\ell-1} + Bu_\ell \ , \tag{3.3}$$

$$P_{\ell|\ell-1} = AP_{\ell-1|\ell-1}A^T + Q \ . \tag{3.4}$$

Eqn. 3.3 updates the estimate of the true state vector at time ℓ based on the knowledge of the previous estimation, *i.e.* at time *ℓ-1*, and the input of the system. Eqn. 3.4 expresses the manner in which our knowledge about the system's state gradually decays over time. It updates the *a priori* error covariance. The prediction phase is corrected according to the feedback obtained from the measurement.

Correction phase equations:

$$K_\ell = P_{\ell|\ell-1}H^T(HP_{\ell|\ell-1}H^T + R)^{-1} \ . \tag{3.5}$$

$$\hat{x}_{\ell|\ell} = \hat{x}_{\ell|\ell-1} + K_\ell(y_\ell - H\hat{x}_{\ell|\ell-1}) \ . \tag{3.6}$$

$$P_{\ell|\ell} = (I - K_\ell H)P_{\ell|\ell-1} \ . \tag{3.7}$$

Eqn. 3.5 computes the Kalman gain K in order to minimize the *a posteriori* error covariance matrix P computed by Eqn. 3.7. Eqn. 3.6 computes the *a posteriori* estimates of the true state vector given measurement up to time ℓ.

KF Parameters Adjustment

Basically, the KF gain K and its evolution through time depends only on the following parameters:

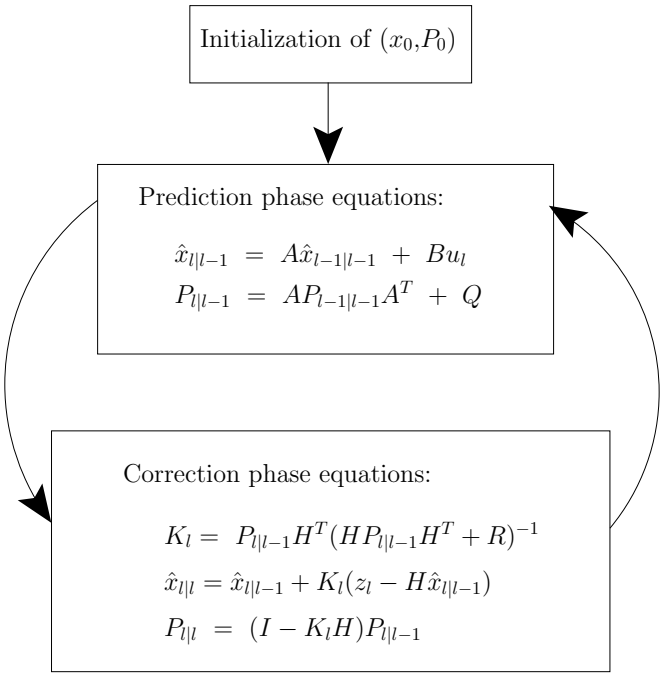


Figure 3.2: Kalman filter algorithm.

1. The process noise covariance Q which represents the confidence we have in the process model. The higher the value attributed to Q, the lesser is the confidence level in the process model.

2. The measurement noise covariance R which represents the confidence we have in the measurement. Higher the value of R, greater is the impact of noise on the measurement.

3. The couple (x_0,P_0) which represents the confidence we have in the initialization.

In the KF implementation, it is usually possible to determine the measurement noise covariance R. In fact, we just need to take some offline measurements, *i.e.* when the process is disabled, in order to get a reliable variance value of the measurement noise. However, the determination of Q is generally more difficult since we do not know the real process model. In fact, it is not possible to directly observe the process we are estimating. In the case of uncertainty about the process model equation, it should be better to give a high value to Q. In the general case, the tuning of R and Q is usually performed by running the filter offline. Regarding the initial estimate of the true state x_0 and the error covariance P_0, it is preferable to fix P_0 large if the correct value of x_0

is uncertain. This means that the KF will take a long time to start taking into account of any measurement sample.

3.2.1.2 Experiments & Results

As discussed before, the main idea behind using KF is to denoise Side-channel traces; and therefore improve the quality of the analysis to best assess the robustness of secure implementations against SCA. Previously, in Chapter 2, we have seen that the traditional metric often used by the evaluator to decide for the correct hypothesis is the number of power consumption signals for the key guess to stabilise. The higher the number of traces to recover the secret key, the more robust the implementation. For sake of clarity, the Kalman model used in the experiment is straightforward and assumes that a power consumption signal can be expressed as a mono-dimensional system with no optional control input. This means that all input matrices are reduced to scalars and the input control vector u_l is omitted. In the open litterature, such basic model has usually shown its efficiency to reduce the amount of white noise, without altering the useful information.

Kalman Filtering for Patterns Detection

In this experiment, we acquired a noisy trace corresponding to one encryption process related to the activity of an AES-128 hardware implementation. This AES implementation needs exactly 51 clock cycles to perform an encryption operation: 44 clock cycles for the 11 rounds operations and 7 ones for the 7 controls operations. Fig. 3.3 shows the original trace as it is acquired by the oscilloscope. Obviously, before using the filter, the encryption process was totally hidden by noise. After filtering, as shown in Fig. 3.4, we can distinguish the UART activity before and after the encryption process. Moreover, we easily recognize our 51 clock cycles corresponding to the AES encryption operation. From the evaluation point of view, the cryptographic patterns detection plays a crucial role in determining the nature of the algorithm (standard or non-standard) and building properly the set of traces before starting performing statistical analyses. Besides, it is an important point to succeed a Simple Power Analysis (SPA).

One other example of Kalman filtering is shown through Fig. 3.5 and Fig. 3.6. Actually, Fig. 3.5 represents the first two rounds (noisy) of a DES implementation. When using KF (Fig. 3.6), clearly the signal is smoothed; and therefore the DES patterns are more visible.

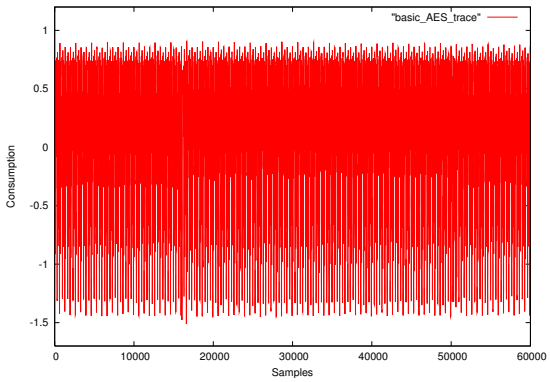


Figure 3.3: An illustration of a noisy AES trace (basic trace).

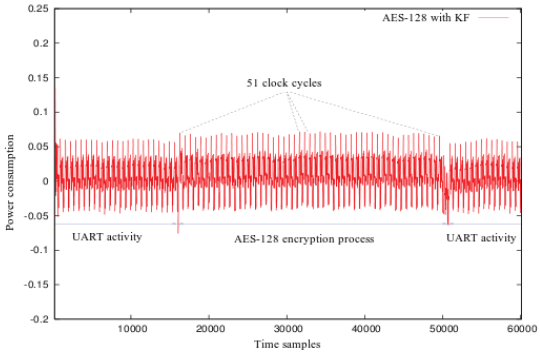


Figure 3.4: AES trace filtered with the Kalman technique.

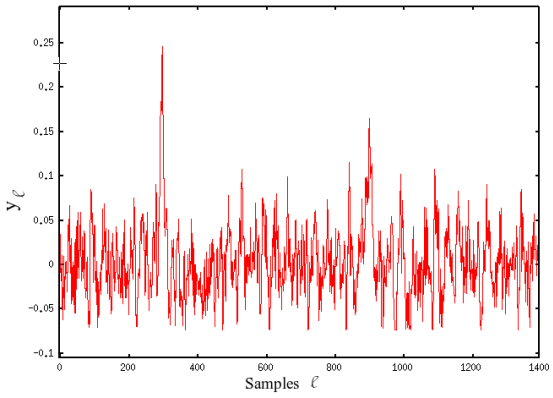


Figure 3.5: DES first round measurement without any filtering.

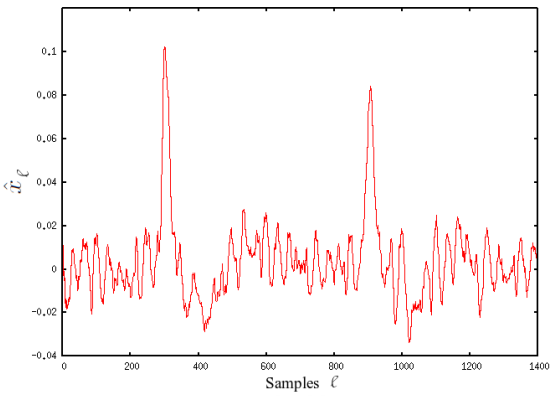


Figure 3.6: DES first round measurement with Kalman filtering.

Effect of the Kalman Filter on Power Analyses

As mentioned before, the effectiveness of the KF on the Differential Power Analysis (DPA) can be evaluated by determining the number of traces needed to retrieve the secret key of the encryption algorithm. For this purpose, we tested our technique on public traces made available by DPA contest v1 [148]. Since these traces were already available, it was not possible to get an available offline white noise measurement. Actually, measuring the variance of the offline noise during the acquisition is the best method, as it is specified in the open litterature of Kalman filtering, to estimate with accuracy the amount of white noise affecting the Side-channel traces. To overcome this problem we propose to exploit the noise already present within the trace itself. In fact,

the extraction of a noise measurement sample without destroying the useful information related to the secret key is a delicate operation. For instance, in our case, this useful information is represented by the activity of the state register corresponding to each round operation, and it takes place at the rising edge of the clock over the encryption process. First, as the architecture of the DES design on which we performed the DPA was known, we were able to determine the path from an input to an output DES round design with the largest delay. This operation is called *critical path identification* and can be easily performed using timing analysis tools. Indeed, such tools are capable to analyse every path from the input to the output, and give the delay on each path. Once critical path identification was done, we were able to precisely localise where the information related to the state register activity takes place over time. Therefore, as shown in Fig. 3.7, noise was extracted without destroying the useful information. We recall that the variance of this noise measurement sample represents the value of the input parameter R of the KF.

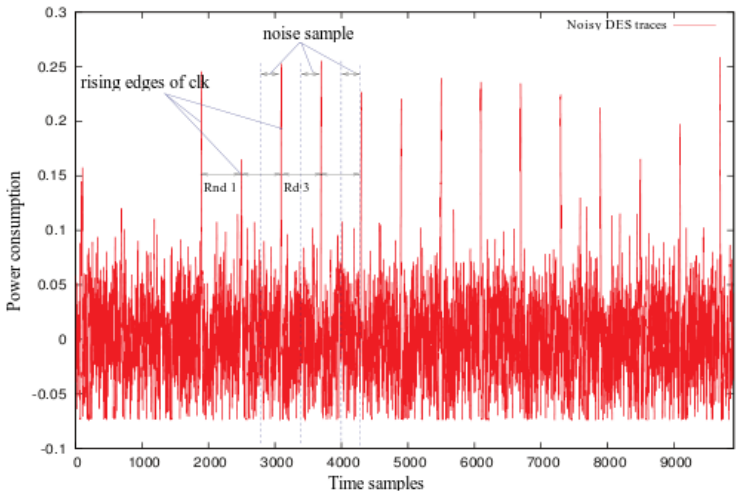


Figure 3.7: An illustration of noise extraction from a DES encryption measurement.

Regarding the other parameters of the KF, we attributed a high value to the input parameter Q which represents the confidence we have in the Kalman process model that we had already fixed. The state measurement matrix H is constrained as a scaled identity matrix since measurements are taken at LOS (Line-Of-Sight) at short distance from the target circuit ($2cm$). The state transition parameter A is chosen such that it provides the Minimum Mean Square Error (MMSE) 2.2.4.2 between the filtered signal and the ideal signal. Previously, in Chapter 2, we have seen that the MMSE is a

commonly used tool to quantify the estimation's goodness of an unknown parameter. The estimation here is the filtered signal and the hidden state is the ideal signal. The MMSE can be defined as:

$$MMSE = arg\min_{\theta}(\mathbb{E}(\theta - \hat{\theta})^2) \ , \tag{3.8}$$

where $\mathbb{E}$ is the Expectation mathematical operator, θ is the ideal signal and $\hat{\theta}$ is the filtered signal. In practice, a good estimation of the parameter A can be determined by fixing the rest of parameters and varying the value of A until reaching the minimum mean square error. From the evaluation perspective, the computation of the ideal signal can be equal to the average of all traces acquired. Nonetheless, the accuracy of the average trace is essentially dependent on the alignment of all traces. In real life, getting a perfect alignment of traces is virtually impossible due to many factors. More in depth discussion is given in Sec. 3.3. In such situation, the ideal signal can not be computed; and more advanced techniques should be investigated. To get round the problem of traces misalignment from the filtering perspective, next in 3.2.2, we propose a new algorithmic solution based on the mixing of the Kalman filtering with the Expectation-Maximization (EM) algorithm [39]. Moreover, such mixing aims at estimating the KF parameters in a real time manner. In our experiment, the average of traces was computed (we verified that traces are not misaligned). Thus, the initial state estimate x_0 was assumed to be equal to the initial value of that computed trace. Moreover, we note that the basic traces were noisy, as they were not averaged by the oscilloscope during the acquisition process. In what follows, we compare the efficiency of the KF and HOS [76] techniques when a DPA is performed. Basically, the HOS filtering is a method which takes power from two notions: high-order cumulants, in particular the fourth-order cumulant known as *kurtosis*, which is a powerful tool to test the normality of an unknown distribution (*i.e* test whether a process is Gaussian), and the sliding window initially related to the moving average filtering. However, in practice, the problem of choosing the optimal size of the sliding window has been often discussed, as no generic rule can be proposed. For each DES Sbox, our DPA tool determines the average number (30 random attacks ran) of traces needed to retrieve the corresponding subkey. The subkey is considered to be recovered only when a certain stability is observed (500 traces). The lesser the number of traces needed, the more efficient the filtering technique. We denote by $DPA/CPA_{(KF/HOS)}$ the DPA or CPA performed on traces filtered either by KF or HOS technique. Results can be depicted in Tab. 3.1. We note that, results regarding the HOS technique correspond to the optimal

size of the sliding window that was determined empirically after exhaustive testing of numerous size values. Moreover, the problem encountered concerning analysed traces relied on the fact that useful signal was totally hidden by a large amount of noise. A gain, G_{filt}, in term of number of traces using a filter, $filt$, was computed as follows:

$$Gain_{filt}(\%) = \frac{N_{basic} - N_{filt}}{N_{basic}}.100(\%) \ , \tag{3.9}$$

where N_{basic} and N_{filt} are respectively the averaged number of noisy electromagnetic (Em) measurements and the averaged number of filtered measurements, required for a complete attack.

Table 3.1: Needed traces to break DES Sboxes.

Sbox	Basic DPA	DPA (KF)	DPA (HOS)		Basic CPA	CPA (KF)	CPA (HOS)
0	1910	740	863		815	712	819
1	5680	5236	6854		5725	5134	5689
2	1675	664	845		1302	585	882
3	4290	3048	4086		3115	2904	3110
4	1793	1078	923		1901	763	1601
5	2602	437	1269		1118	478	462
6	2341	813	1165		822	639	986
7	485	219	517		229	208	228
Avg	2597	1530	2065		1878	1429	1722
G_{filt}	-	41%	20.48%		-	23.90%	8.3%

According to Tab. 3.1, clearly the KF not only improves the Basic DPA for all Sboxes, but also outperforms the HOS technique. For instance, the KF gain is roughly twice the HOS gain, when performing a DPA. In the same way as for DPA, we performed the Correlation Power Analysis (CPA) that usually shows better results than DPA. Indeed, according to Tab. 3.1, the number of traces needed to find all subkeys, when performing a basic CPA, is much lower than performing a basic DPA (*i.e.* on original traces). Besides, both filtering techniques are still efficient but with lower gains, compared to the gains found for DPA. This can be explained by the CPA's capability of reducing the amount of noise thanks to its normalization factor, as shown previously in Chapter 2. In the general case, according to Tab. 3.1, the KF is more powerful than the HOS technique. However, it is noteworthy that the KF is more complicated to implement (parameters estimation, rapidity) than HOS filtering that principally needs only one parameter which is the size of the sliding window.

3.2.1.3 Conclusion

In this part of Chapter, we have proposed an algorithmic solution, based on the well known Kalman filtering theory [156], to reduce the amount of noise affecting Side-channel traces; and therefore improve the analysis. Moreover, we have presented the ways to efficiently adapt Kalman filtering to the SCA context. In the next Section 3.2.2, we investigate the Expectation-Maximization algorithm (EM) to get a better estimation of the Kalman filter (KF) parameters.

3.2.2 Kalman Combined Expectation Maximization Algorithm

As explained previously, the Kalman filter provides a state estimate conditioned on the past and present observations, and so is a causal estimator. This estimator is a good solution for real-time purposes, which might be helpful from the attacker point of view, in contrast to the evaluator. Such solution would be nicely improved if time constraints were relaxed. Nonetheless, we recall that this thesis essentially aims at providing the evaluator with new tools to reliably assess his analysis. In this part of thesis, we propose to improve the estimation of KF parameters thanks to a new complementary tools: *Kalman Smoother* followed by *Expectation-Maximization algorithm* (EM). In fact, in practice it is possible to postpone the calculation of the estimate (filtered signal) until future data are measured. The new estimate ought to construct a smoothed, or non causal, estimator from future, as well as the past, data.

3.2.2.1 An Overview of Kalman Smoother

In what follows, we denote by $Y_s = (y_1, \ldots, y_s)$ the observation vector where $s \leq L$ (with L the size of the whole measurement), the estimation is equal to:

$$\widehat{x}_{\ell|s} = E[x_\ell|Y_s] \ . \tag{3.10}$$

We denote by $P_{\ell_1,\ell_2|s}$ the following expression:

$$P_{\ell_1,\ell_2|s} = E\left[(x_{\ell_1} - \widehat{x}_{\ell_1|s})(x_{\ell_2} - \widehat{x}_{\ell_2|s})|Ys\right] \ . \tag{3.11}$$

Thus, the estimate error covariance of $\widehat{x}_\ell$ according to $\widehat{x}_{\ell|\ell-1}$ knowing $\ell - 1$ observation elements, *i.e.* $Y_{\ell-1}$, is equal to:

$$P_{\ell|\ell-1} = E\left[(x_\ell - \widehat{x}_{\ell|\ell-1})^2|Y_{\ell-1}\right] \ . \tag{3.12}$$

We have seen previously that the Kalman filter predicts an estimation of x_ℓ with regards to the previous estimation $\widehat{x}_{\ell-1|\ell-1}$ and the current observation component y_ℓ. Thus, the Kalman filter is able to display $\widehat{x}_{\ell|\ell}$ knowing $\widehat{x}_{\ell-1|\ell-1}$ and y_ℓ according to *prediction* and *correction* equations, studied previously. In what follows, these equations are simplified by assuming that the state transition matrix A and the observation matrix H are scalars, noises v_ℓ and w_ℓ are additive white Gaussian noises of variance σ_v^2 and

σ_w^2, respectively. w is called the process noise and v is called the measurement noise. Hence, the equations can be written as follows:

$$\widehat{x}_{\ell|\ell-1} = A\widehat{x}_{\ell-1|\ell-1} \ . \tag{3.13}$$

$$P_{\ell|\ell-1} = P_{\ell-1|\ell-1}A^2 + \sigma_w^2 \ . \tag{3.14}$$

$$K_\ell = \frac{P_{\ell|\ell-1}H}{H^2 P_{\ell|\ell-1} + {\sigma^2}_v} \ . \tag{3.15}$$

$$\widehat{x}_{\ell|\ell} = \widehat{x}_{\ell|\ell-1} + K_\ell \ .(y_\ell - H\widehat{x}_{\ell|\ell-1}) \ . \tag{3.16}$$

$$P_{\ell|\ell} = (1 - K_\ell H)P_{\ell|\ell-1} \ . \tag{3.17}$$

The initial state conditions $\widehat{x}_{0|0}$ and $P_{0|0}$are respectively denoted by μ an Σ, assuming that x_0 is drawn from a Gaussian distribution (*i.e.* $\mathcal{N}(\mu, \Sigma)$).

Kalman Smoother:

As we have discussed before, calculations will be executed after measurement has been taken. The Kalman Smoother (KS) is applied on the observation to take advantage from the past, the present and future observation for data estimation. Thus the estimate of ℓ^{th} component knowing the observation vector Y_L 3.2.1.1 is given by

$$\widehat{x}_{\ell|L} = E\left[x_\ell|Y_L\right] = \widehat{x}_{\ell|\ell} + J_\ell(\widehat{x}_{\ell+1|L} - \widehat{x}_{\ell+1|\ell}) \ , \tag{3.18}$$

where

$$J_\ell = \frac{P_{\ell|\ell}A}{P_{\ell+1|\ell}} \ , \tag{3.19}$$

$$P_{\ell|L} = P_{\ell|\ell} + J_\ell^2(P_{\ell+1|L} - P_{\ell+1|\ell}) \ . \tag{3.20}$$

Obviously, the difference between KF and KS relies on the fact that KS takes into consideration the whole observation to decide upon the current time sample estimation. Whereas, the KF is based only on previous time samples to estimate the current one. Theoretically, applying KS should lead to better results for data estimation than KF. In fact, literature has proved that the estimates knowing the observation of past, present and future are better than those knowing only the present and past [97].
From the practical point of view, KS is more efficient than KF in detecting breaks and discontinuities within a signal. Besides, the use of KS here is necessary for the application of the Maximization-Expectation algorithm (EM), which needs to analyse

the entire observation in order to return a reliable estimation of the Kalman input parameters. The estimate error covariance $P_{\ell|L}$ is useful for the EM update.

3.2.2.2 Updating Kalman Parameters with EM Algorithm

The Kalman filter efficiency relies on the values attributed to the input parameters that define the system model. Basically, we want to use the available measurement, *i.e.* the vector $\mathbf{y}$ 3.2.1.1 in order to evaluate the initial parameters required for Kalman initialisation. Thus, we need to determine and estimate the mean and the variance of the initial state x_0 (*i.e.* μ and Σ), such that $x_0 \sim \mathcal{N}(\mu, \Sigma)$.

The EM algorithm [39; 96] is applied to find the maximum likelihood $\log p(\mathbf{x}, \mathbf{y}|\theta)$ and it is especially effective when the likelihood of the incomplete data is much more difficult to maximize than the likelihood of the complete data. We denote by incomplete data the observed vector $\mathbf{y}$ (or the measurement), by missing data the hidden state vector $\mathbf{x}$, by complete data the couple $\Upsilon = (\mathbf{x}, \mathbf{y})$ and by θ the parameter to be estimated. In our case θ is the set of parameters $\{\mu, \Sigma, A, H, \sigma_v^2, \sigma_w^2\}$.
Indeed, if the complete data Υ is available, the maximum-likelihood (ML) estimation of θ is obtained by maximizing the likelihood or equivalently the log-likelihood:

$$\hat{\theta}_{ML} = arg\max_{\theta} \log\left(p(\Upsilon|\theta)\right) = \log\left(p(\mathbf{x}, \mathbf{y}|\theta)\right). \tag{3.21}$$

In our study, the incomplete data (or the measurement) is the only available observation to the evaluator. The real leakage corresponding to the activity of the cryptographic implementation, $\mathbf{x}$, is missing, and ML estimation as in 3.21 cannot be computed. In such case, as detailed in [66], ML estimation becomes:

$$\hat{\theta}_{ML} = arg\max_{\theta} \log\left(p(\mathbf{y}|\theta)\right). \tag{3.22}$$

Now, one problem arises: in many cases such as in the Side-channel context, the maximization of $\log\left(p(\mathbf{y}|\theta)\right)$ for a ML estimation is very difficult, since the analytical expression of the log-likelihood is not available. To get round this problem, the EM algorithm is a proper solution to 3.22. Given a current parameter value $\theta^{(i)}$ at iteration *(i)*, the EM algorithm computes an update $\theta^{(i+1)}$. The final EM estimate depends on the initial value $\theta^{(0)}$. Moreover, the convergence of the algorithm is relatively rapid. In our case, the use of EM algorithm aims to reliably estimate the KS parameters; and therefore to improve the Side-channel noise filtering process. The EM algorithm starts

from an initial value of $\theta^{(0)}$ and it improves this value in an iterative manner. This algorithm involves two steps at each iteration: the expectation step (**E-step**) and the maximization step (**M-step**). Given a current parameter value $\theta^{(i)}$ at iteration *(i)*, the *EM* algorithm computes an update $\theta^{(i+1)}$. The final *EM* estimate depends on the initial value $\theta^{(0)}$. Formally:

1. $\theta^{(0)}$ is determined (arbitrarily).
2. **for** each iteration (i) (i=1,2,...), the following steps are processed
   (a) **E-step**: the expectation value of log-likelihood of complete data conditioned by observed samples and the current value of $\theta^{(i)}$ are computed:

$$\Lambda(\theta|\theta^{(i)}) = \mathbb{E}_{\mathbf{x}}[\log\left(p(\mathbf{x},\mathbf{y}|\theta)|\mathbf{y},\theta^{(i)}\right)] \ . \tag{3.23}$$

   (b) **M-step**: $\theta^{(i+1)}$ that maximize the auxiliary function $\Lambda(\theta|\theta^{(i)})$ is calculated

$$\theta^{(i+1)} = arg\max_{\theta} \Lambda(\theta|\theta^{(i)}) \ . \tag{3.24}$$

In the case of unknown source distribution and by the means of Bayes' rule and considering that $\mathbf{x}$ and θ are independent, we get

$$p(\mathbf{x},\mathbf{y}|\theta) = p(\mathbf{y}|\mathbf{x},\theta)p(\mathbf{x}|\theta) = p(\mathbf{y}|\mathbf{x},\theta)p(\mathbf{x}) \propto p(y|x,\theta) \tag{3.25}$$

Hence, the new auxiliary function 3.23 can be expressed as

$$\begin{aligned} \Lambda(\theta|\theta^{(i)}) &= \mathbb{E}_{\mathbf{x}}[\log p(\mathbf{y}|\mathbf{x},\theta)|\mathbf{y},\theta^{(i)}] = \sum_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x},\theta) P(\mathbf{x}|\mathbf{y},\theta^{(i)}) \\ &= \sum_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x},\theta) APP_i(\mathbf{x}) \ , \end{aligned} \tag{3.26}$$

where $APP_i(\mathbf{x}) = P(\mathbf{x}|\mathbf{y},\theta^{(i)})$ is the *a posteriori probability* of $\mathbf{x}$ at the i^{th} iteration of the EM algorithm.

$$\Lambda(\theta|\theta^{(i)}) = \mathbb{E}\left[\log p(\mathbf{x},\mathbf{y}|\theta)|\theta^{(i)}Y\right] = \mathbb{E}_{X|Y}\left[\log p(\mathbf{x},\mathbf{y}|\theta)|\theta^{(i)}\right] \ . \tag{3.27}$$

The Log-likelihoods function inside the Auxiliary function has the following expression

$$\log p(\mathbf{x},\mathbf{y}|\theta) = \log p(\mathbf{x}) + \log p(\mathbf{y}|\mathbf{x}\theta) \ , \tag{3.28}$$

$$= \log p(x_0,\ldots,x_L) + \log p(y_1,\ldots,y_L|x_0,\ldots,x_L,\theta) \ . \tag{3.29}$$

Since x_ℓ depends only on $x_{\ell-1}$, we have then

$$p(\mathbf{x}) = p(x_0,\ldots,x_L), \tag{3.30}$$

$$= p(x_L|x_0,\ldots,x_{L-1})p(x_0,\ldots,x_{L-1}), \tag{3.31}$$

$$= p(x_L|x_{L-1})p(x_{L-1}|x_0,\ldots,x_{L-2})p(x_0,\ldots,x_{L-2}), \tag{3.32}$$

$$\vdots$$

$$= p(x_0)\prod_{\ell=1}^{L} p(x_\ell|x_{\ell-1}) \ . \tag{3.33}$$

As for Eqn.3.33, we apply the same method to $p(\mathbf{y}|\mathbf{x})$. Indeed, y_ℓ depends only on x_ℓ. Hence, from the independence property of $\{y_\ell\}$ (*i.e* y_ℓ depends only on x_ℓ), we have:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{\ell=0}^{L} p(y_\ell|x_\ell) \ . \tag{3.34}$$

Applying 3.33 and 3.34 to the Auxiliary function 3.27, this leads to

$$\Lambda(\theta|\theta^{(i)}) = \mathbb{E}_{X|Y}\left[\log p(x_0) + \sum_{\ell=1}^{L}\log p(x_\ell|x_{\ell-1}) + \sum_{\ell=0}^{L}\log p(y_\ell|x_\ell)|\theta^{(i)}\right] \ , \tag{3.35}$$

$$= \underbrace{\mathbb{E}_{X|Y}\left[\log p(x_0)|\theta^{(i)}\right]}_{\alpha} + \underbrace{\sum_{\ell=1}^{L}\mathbb{E}_{X|Y}\left[\log p(x_\ell|x_{\ell-1})|\theta^{(i)}\right]}_{\beta} + \underbrace{\sum_{\ell=0}^{L}\mathbb{E}_{X|Y}\left[\log p(y_\ell|x_\ell)|\theta^{(i)}\right]}_{\gamma} \ . \tag{3.36}$$

The theoretical equations development of α, β and γ components are given in the **Appendix** B. Actually, the expressions of α, β and γ can be rewritten, respectively, as:

$$\alpha = \mathbb{E}_{X|Y}\left[\log p(x_0)|\theta^{(i)}\right] = C^{te} - \frac{\Sigma}{2} - \frac{1}{2\Sigma}\left(P_{0|L} + (\widehat{x}_{0|L} - \mu)^2\right) \ , \tag{3.37}$$

where C^{te} is a constant.

$$\beta = \sum_{\ell=1}^{L} \mathbb{E}_{X|Y}\left[\log p(x_\ell|x_{\ell-1})|\theta^{(i)}\right] = \\ C^{te} + -\frac{L}{2}\log\sigma_w^2 - \frac{1}{2\sigma_w^2}\sum_{\ell=1}^{L}\left[P_{\ell|L} + A^2 P_{\ell-1|L} + (\widehat{x}_{\ell|L} - A\widehat{x}_{\ell-1|L})^2 - 2AP_{\ell,\ell-1|L}\right]. \tag{3.38}$$

$$\gamma = \sum_{\ell=0}^{L} \mathbb{E}_{X|Y}\left[\log p(y_\ell|x_\ell)|\theta^{(i)}\right] = C^{te} - \frac{L}{2}\log\sigma_v^2 - \frac{1}{2\sigma_v^2}\sum_{\ell=0}^{L}\left[(y_\ell - \widehat{x}_{\ell|L})^2 + H^2 P_{\ell|L}\right]. \tag{3.39}$$

Therefore, according to the equations development of α, β and γ components, the Auxiliary function Λ can be expressed as:

$$\begin{aligned}\Lambda(\theta|\theta^{(i)}) =& C^{te} - \frac{\Sigma}{2} - \frac{1}{2\Sigma}\left(P_{0|L} + (\widehat{x}_{0|L} - \mu)^2\right) - \frac{L}{2}\log\sigma_w^2 - \frac{L}{2}\log\sigma_v^2 \\ &- \frac{1}{2\sigma_w^2}\sum_{\ell=1}^{L}\left[P_{\ell|L} + A^2 P_{\ell-1|L} + (\widehat{x}_{\ell|L} - A\widehat{x}_{\ell-1|L})^2 - 2AP_{\ell,\ell-1|L}\right] \\ &- \frac{1}{2\sigma_v^2}\sum_{\ell=0}^{L}\left[(y_\ell - \widehat{x}_{\ell|L})^2 + H^2 P_{\ell|L}\right].\end{aligned} \tag{3.40}$$

In order to get the update of parameters required for Kalman filter, we maximize the expression 3.40 with respect to $\theta = \{\mu, \Sigma, A, H, \sigma_w^2, \sigma_v^2\}$ as follows:

$$\theta^{(i+1)} = arg\max_{\theta} \Lambda(\theta|\theta^{(i)}) \tag{3.41}$$

The update of parameters are obtained by deriving Eqn. 3.40 with respect to θ. This

yields to solve the following derivations

$$\frac{\partial \Lambda(\theta|\theta^{(i)})}{\partial \mu} = 0 \ . \tag{3.42}$$

$$\frac{\partial \Lambda(\theta|\theta^{(i)})}{\partial \Sigma} = 0 \ . \tag{3.43}$$

$$\frac{\partial \Lambda(\theta|\theta^{(i)})}{\partial A} = 0 \ . \tag{3.44}$$

$$\frac{\partial \Lambda(\theta|\theta^{(i)})}{\partial H} = 0 \ . \tag{3.45}$$

$$\frac{\partial \Lambda(\theta|\theta^{(i)})}{\partial \sigma_w^2} = 0 \ . \tag{3.46}$$

$$\frac{\partial \Lambda(\theta|\theta^{(i)})}{\partial \sigma_v^2} = 0 \ . \tag{3.47}$$

These equations lead to the set of update of parameters value at the i^{th} iteration of the EM algorithm. Hence we have:

$$\mu^{(i+1)} = \widehat{x}_{0|L}^{(i)} \tag{3.48}$$

$$\Sigma^{(i+1)} = P_{0|L}^{(i)} \tag{3.49}$$

$$A^{(i+1)} = \frac{\sum_{\ell=1}^{L} \left(P_{\ell,\ell-1|L}^{(i)} + \widehat{x}_{\ell|L}^{(i)} \ \widehat{x}_{\ell-1|L}^{(i)} \right)}{\sum_{\ell=1}^{L} \left(P_{\ell-1|L}^{(i)} + \left(\widehat{x}_{\ell-1|L}^{(i)} \right)^2 \right)} \tag{3.50}$$

$$H^{(i+1)} = \frac{\sum_{\ell=0}^{L} \left(y_\ell \ \widehat{x}_{\ell|L}^{(i)} \right)}{\sum_{\ell=0}^{L} \left(P_{\ell|L}^{(i)} + \left(\widehat{x}_{\ell|L}^{(i)} \right)^2 \right)} \tag{3.51}$$

$$\sigma_w^{2\,(i+1)} = \frac{1}{L} \sum_{\ell=1}^{L} \left[P_{\ell|L}^{(i)} + \left(\widehat{x}_{\ell|L}^{(i)} \right)^2 - A^{(i^+1)} \left(P_{\ell,\ell-1|L}^{(i)} + \widehat{x}_{\ell|L}^{(i)} \ \widehat{x}_{\ell-1|L}^{(i)} \right) \right] \tag{3.52}$$

$$\sigma_v^{2(i+1)} = \frac{1}{L+1} \sum_{\ell=0}^{L} \left[y_\ell^2 + \left(H^{(i+1)} \right)^2 \left(P_{\ell|L}^{(i)} + \left(\widehat{x}_{\ell|L}^{(i)} \right)^2 \right) - 2 \ y_\ell \ H^{(i+1)} \ \widehat{x}_{\ell|L}^{(i)} \right] \tag{3.53}$$

In our study, the EM algorithm and KS were jointly implemented. The EM algorithm is first used to estimate the initial parameters of the KS. After a first iteration of the EM algorithm, KS is applied on the received data. The initial parameters are

iteratively improved after each KS application.

3.2.2.3 Experiments & Results

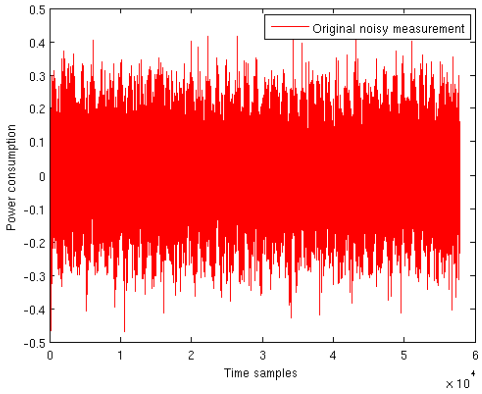


Figure 3.8: Original trace of three AES-128 encryptions in CBC mode.

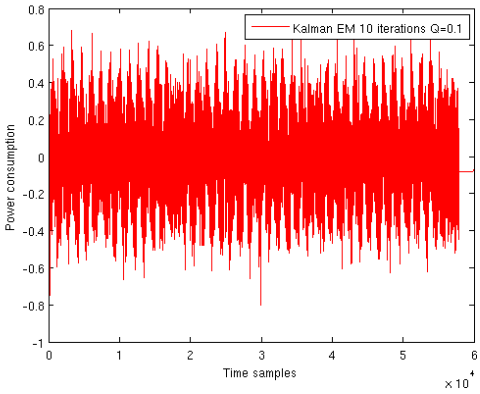


Figure 3.9: KF with 10 EM iterations with $Q = 0.1$, $R = 0.5$.

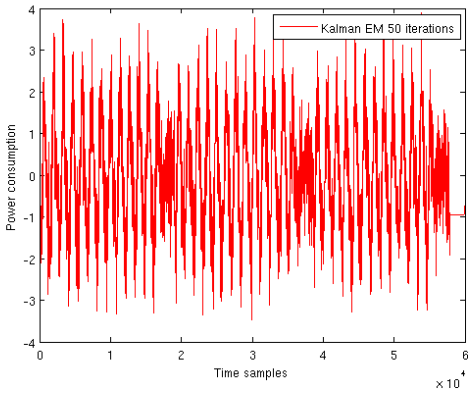


Figure 3.10: KF with 50 EM iterations with $Q = 0.1$, $R = 0.5$.

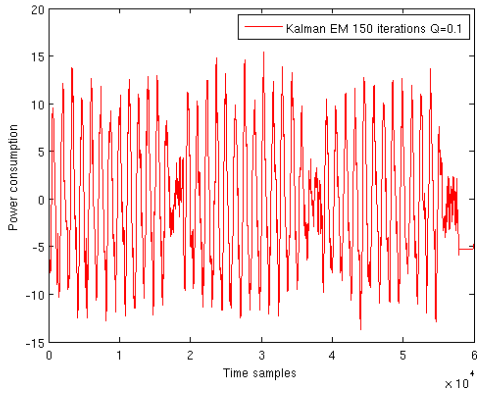


Figure 3.11: KF with 150 EM iterations with $Q = 0.1$, $R = 0.5$.

In this experiment we have acquired a real measurement of three AES-128 encryptions in CBC mode. Fig. 3.8 shows the original trace as it was acquired by the oscilloscope. In order to apply Kalman filter combined with EM algorithm, the input parameters ($A = 1$, $H = 1$, $Q = \sigma_w^2 = 0.1$ and $R = \sigma_v^2 = 0.5$) were fixed arbitrarily. We recall that these values represent the initial value $\theta^{(0)}$, taken by the EM algorithm. Obviously, according to Figures. 3.9, 3.10, 3.11, 3.12 and 3.13, the efficiency of the

Kalman combined EM noise filtering process is improved when the number of iterations is getting higher.

But more importantly, we note that the efficiency of the filtering starts stabilising when the algorithm converges to the optimal estimations. For instance, the shape of the filtered signal shown in Fig. 3.13 (after 300 iterations processed) is slightly the same as the signal shown in Fig. 3.14 (after 500 iterations processed). In this experiment, the actual value (*i.e.* obtained after the convergence of the algorithm) of noise variance ($R = 0.9$) turned out to be bigger than the one we fixed initially (*i.e.* R=0.5). Now, in order to see the effect of attributing proper parameters values to the initial state of the algorithm, we took the latter obtained (optimal) value (*i.e.* $R = 0.9$) as the initial noise variance. Unsurprisingly, as depicted in Fig. 3.15, the noise within the signal is clearly reduced after only 10 EM iterations. Moreover, during our analysis, we noticed that the EM algorithm needed around 90 iterations to converge. Consequently, the more accurate the input values (*i.e* A, H, $Q = \sigma_w^2$ and $R = \sigma_v^2$), the lesser the number of iterations required for EM convergence.

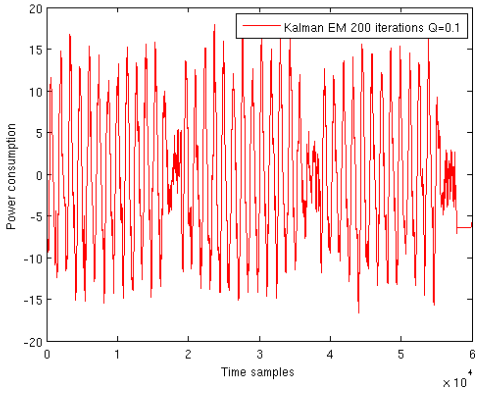


Figure 3.12: KF with 200 EM iterations with $Q = 0.1$, $R = 0.5$.

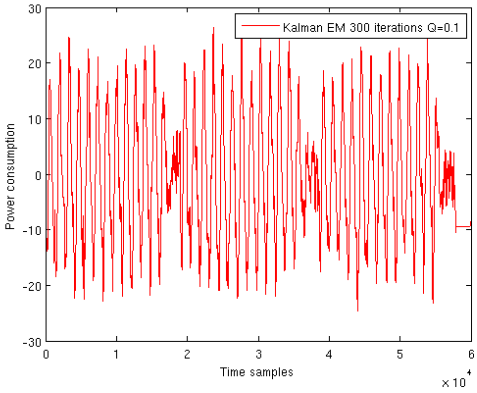


Figure 3.13: KF with 300 EM iterations with $Q = 0.1$, $R = 0.5$.

3.2.2.4 Conclusion

In this Chapter, we have presented an efficient solution to improve the basic Kalman filtering algorithm, proposed previously in Sec. 3.2.1. The solution proposed is based on the powerful Expectation-Maximization iterative algorithm combined with the Kalman Smoother (KS) filtering. The overall goal has been to accurately estimate the Kalman Filter (KF) parameters; and therefore to improve the pre-processing of Side-channel traces.

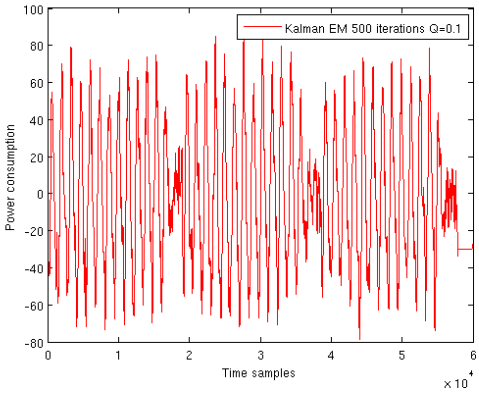


Figure 3.14: KF with 500 EM iterations with $Q = 0.1$, $R = 0.5$.

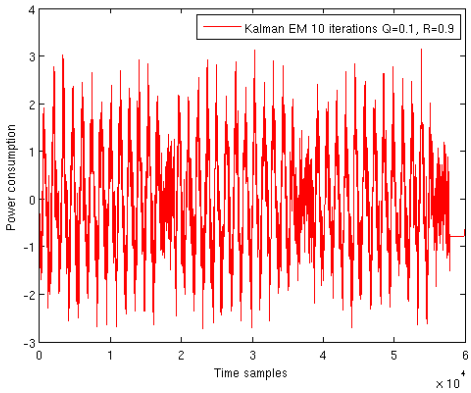


Figure 3.15: KF with 10 EM iterations with $Q = 0.1$, $R = 0.9$.

3.2.3 Wavelets: A Multiresolution Time-Frequency Analysis

Today's evaluator already has at his disposal several tools to properly analyse Side-channel traces. Up to this point, this thesis has dealt only with the pre-processing of time information included within the signal acquired. Actually, the pre-processing of traces can be also performed in the frequency domain using for instance the well known *Fourier Transform*. Previously, in Chapter 2, we have stated that Side-channels are preferred to be performed in the time domain than in the frequency domain. Moreover, we have shown that Side-channel attack algorithms can be markedly improved thanks to *Wavelet transforms*, known as *multiresolution analysis*, that advantageously deal with both time and frequency domains. In this Chapter, we will provide the evaluator with new Wavelets based Side-channel applications. More precisely, we will be interested only in the pre-processing aspect. For this purpose, we first highlight the ability of wavelets to detect and extract the patterns of a cryptographic process when performing a simple Side-channel analysis (*i.e.* a direct interpretation of the traces acquired). Second, we propose an improvement of the basic method of Pelletier *et .al*, proposed in [110] and used for Side-channel traces noise filtering. Indeed, we discuss the problem of noise filtering from the SCA context and propose to employ an information theoretic approach as a complementary tool and an enhancement for the basic method. Moreover, we show how wavelets can be properly used to make real-time compression of Side-channel traces without loss of information. Indeed, it is known that, when acquiring Side-channel signals, cryptographic patterns need a high sampling rate to be detected by the existing means, especially when the implementation is protected. Therefore, a large memory depth and storage are required. Obviously, the goal of compressing SCA traces is to relax the space constraint to store, without loss, the sensitive information.

3.2.3.1 Wavelets for Cryptographic Patterns Detection

In the Side-channel domain, the main challenge of the evaluator is to best analyse and exploit dependencies between the manipulated data and the electric (power, electro-magnetic, *etc.*) consumption leaked from a CMOS circuit. In practice, the evaluator analyses a set of Side-channel consumption signals (or traces). Each trace includes a block of operations occurring during a cryptographic process (*i.e.* encryption or de-cryption). In real life applications, encrypted data bits are divided into small blocks of bits, called blocks of operations, depending on the specification of the cryptographic

algorithm. For instance, Block ciphers, like AES algorithm, are related to different modes of operations (*e.g.* ECB, CBC, CFB, OFB)(refer to 1) that aim at dividing the encrypted data and managing the way of operation of the obtained blocks. Moreover, in the context of SCA, the alignment of traces is of great concern since deployed analyses are very sensitive to the magnitude of acquired traces. Consequently, in order to build a proper set of traces, the evaluator should be able to detect the start and the end of each block of operation. Unfortunately, in practice, it is almost impossible to perfectly collect aligned traces due to many factors. A frequent situation is that the trigger signal, which is precisely synchronized with the cryptographic process and used in functional testing or academic cases, is removed by the designer for security reasons. Additionally, the acquired traces are very often disturbed by the presence of noise. In such situation, we propose to use the Continuous Wavelet Transform (CWT) 2.6.2.4 to reveal the global information involved by the cryptographic process. For this purpose, we used the same measurement recorded previously for Kalman-EM analysis 3.2.2.3. We recall that signal acquired involves the activity of three AES-128 encryptions in CBC mode and implemented in a Virtex-5 FPGA soldered on a XILINX LX30 platform. Fig. 3.16 is a capture generated from MATLAB [150] (*wavemenu toolbox*), the powerful numerical computing environment. The top of this figure shows the original signal as it is acquired by the oscilloscope. Clearly, the signal is disturbed by a high amount of noise and no information about the cryptographic process can be revealed. At the center of the figure, we can see the two dimensional representation of the CWT (*Coiflet* mother wavelet [103] used). Obviously, thanks to this representation, we can easily detect the limits of the three measured AES blocks. Actually, we have added the dashed lines to highlight these limits. We note that these limits are detected for high scales (low frequencies) of CWT. Moreover, we can reveal more details about the information content of the signal, such as the number of rounds composing each block of AES. This may be very helpful for the evaluator, specially when the analysed algorithm is unknown. Besides, as shown at the bottom of the figure, MATLAB tool generates an approximate shape (*i.e.* extracting the global information) of the analysed signal based on the CWT coefficients. More precisely, it represents the wavelet coefficients computed for the fixed scale 400, and which actually corresponds to low frequencies.

3.2.3.2 Wavelets Combined Mutual Information for Side-Channel Traces Filtering

Side-Channel Traces Noise Filtering Using Wavelets

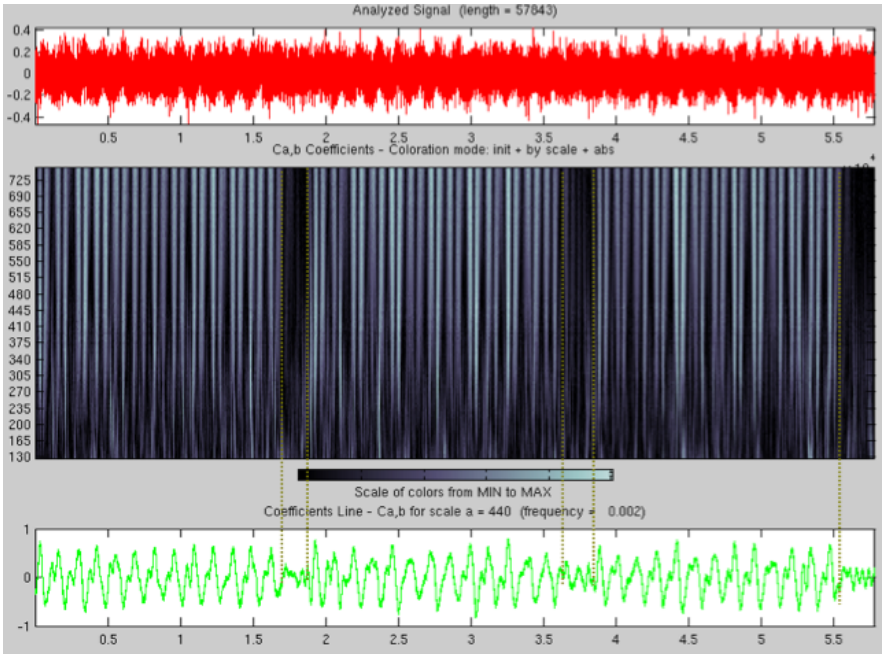


Figure 3.16: An illustration of the CWT on three AES encryption blocks.

In SCA litterature, the procedure of denoising Side-channel traces, using Wavelets transforms, was first presented in [110]. The authors in [110], proposed the very general method to remove noise from signals as it is described in Signal Processing books, without detailed explanations. The method is based on the Discrete Wavelet Transform (DWT). Actually, we have seen that a DWT decomposition leads to a set of coefficients: the approximations for low frequencies and the details for high frequencies. In the literature of wavelet analysis, it has been often shown that the details coefficients can be discarded (set to zero) if their amplitude is not significant; and this without loss of sensitive information. Actually, discarded details are often those related to the noise. This becomes the main idea behind *thresholding* all frequencies that are less than a particular threshold to zero and use these coefficients to reconstruct the denoised version of the original signal. The threshold value is estimated using the well known formula of Donoho [80]. The Donoho's threshold, λ_{Donoho}, can be computed as follows:

$$\lambda_{Donoho} = \sigma\sqrt{2\log(len)}\ ,\ \sigma = \frac{median|details|}{0.6745} \tag{3.54}$$

$$\begin{cases} Th_{hard}(c) & = c \quad if \quad |c| > \lambda_{Donoho} \ , \\ & = 0 \quad if \quad |c| \leq \lambda_{Donoho} \ . \end{cases} \tag{3.55}$$

where σ and $'len'$ are respectively the noise dispersion and the length of the details coefficients. This threshold is applied only on the details coefficients for a specific wavelet scale level. In fact, a new threshold is computed for each scale and used in a *thresholding function*. There is two basic types of thresholding functions: a hard thresholding Th_{hard}, defined in Eqn. (3.55), that sets to zero all details coefficients that are below the threshold value λ_{Donoho}; and a soft thresholding Th_{soft}, defined in Eqn. (3.56), for which the details coefficients with values smaller than λ_{Donoho} are set to zero, but the retained coefficients are also manipulated to reduce the amount of noise affecting wavelet coefficients.

$$\begin{cases} Th_{soft}(c) & = sign(c)(|c| - \lambda_{Donoho}) \quad if \quad |c| > \lambda_{Donoho} \ , \\ & = 0 \quad if \quad |c| \leq \lambda_{Donoho} \ . \end{cases} \tag{3.56}$$

Improvement of the Noise Filtering Basic Method In the open literature of

Wavelet analysis, many scientific papers deal with the efficiency of Donoho's threshold to treat digitally acquired signals. However, we assume that such thresholding techniques are very general and not sufficient to treat the problem of noise reduction from the SCA context. Actually, from the security evaluation perspective, the evaluator usually knows the value of the secret key. His goal is to make his analysis in the best conditions; and thus to know at which point a secure implementation can be resistant to Side-channel attacks. Therefore, the knowledge of the secret key can be a crucial advantage to the security evaluation analysis. In this context, we propose to use the powerful *information theoretic approach* as a complementary tool to Donoho's threshold. This information theoretic approach is basically used to measure the amount of the useful information, which is leaked from the cryptographic implementation. Technically speaking, this metric is mainly based on the mutual information theory. Previously, in Chapter 2 and in the context of SCA, we have evaluate the amount of information in the Side-channel leakages with the mutual information(MI) 3.17, measured in bits. Thus, such tool allows the evaluator to detect the time instants that are directly related to the secret key, during the cryptographic process. This has a pure practical flavour for the analysis, as the evaluator will be able to improve the basic threshold of Donoho. An illustration of MI computation is shown in Fig. 3.17. The MI is computed

over the first-level DWT decomposition of an unprotected DES implementation (500 noisy traces used and only the first two rounds were considered). We remark that for an unprotected implementation, the sensitive information is mainly located within the approximations which are related to the low frequencies, specifically when a low level of decomposition is used. However, during our experiments, we generally noticed that, for several implementations in particular the protected ones, sensitive information can not be negligible in the details' region. In this study, the mutual information is used as a complementary tool to Donoho's threshold in order to add more information about the real features of traces acquired; and therefore to improve the reliability and the accuracy of the initial thresholding. Our proposed mutual information based Donoho's threshold, can be stated in four steps as follows:

- **step 1** Applies the DWT over all traces acquired for a desired level of decomposition.
- **step 2** Computes the MI over the set of DWT coefficients obtained.
- **step 3** Keeps in memory (table T) the temporal indexes of all coefficients which values are located below a certain threshold. (*i.g.* 90% of Max(MI)).
- **step 4** Pre-processes the original traces as follows: **for each** trace, the DWT is first computed (with the same level of decomposition as used previously). Second, λ_{Donoho} is calculated based on the obtained DWT coefficients. Finally, **for each** coefficient c that value is above λ_{Donoho} **and** that index belongs to the table T, **then** the value of c is set to zero.

Clearly, the new algorithm aims at discarding more coefficients that are really related to noisy time samples; and therefore favorising only the actual coefficients that are related to the sensitive information. Basically, the Donoho's threshold is computed over the details; nonetheless it is noteworthy that in the open litterature of Wavelet analysis other thresholds have been proposed to deal with both the approximation and the details like proposed in [134]. We note that our proposed algorithm is **generic** and can be plugged to any proposed threshold. In order to compare the efficiency of both filtering methods (*i.e.* 'Donoho' versus 'Donoho combined MI'), we performed a DPA on an unprotected DES noisy traces. The Guessing entropy metric is depicted through Fig. 3.18. We can easily verify that DPA when performed with 'Donoho combined MI' filtering is more efficient than 'Donoho' alone. Actually, we note that for 'Donoho' filtering the key rank is evolving much more slowly towards the best rank than 'Donoho

combined MI'. Moreover, Fig. 3.18 shows, in the same context, the efficiency of Kalman filter combined with EM algorithm (termed by KF-EM in the figure), which has been proposed in Sec. 3.2.2. Globally, 'KF-EM' technique slightly outperforms the 'Donoho' filter.

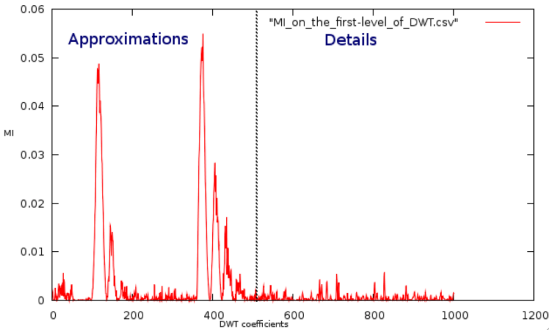


Figure 3.17: Computation of the mutual information (MI) on the first-level DWT of unprotected DES traces.

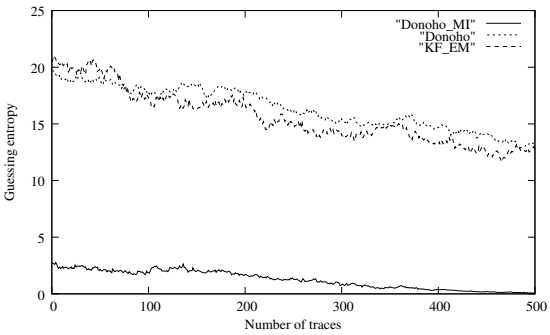


Figure 3.18: Comparison of 'Donoho', 'Donoho combined MI' and 'Kalman combined EM' efficiency.

A logical result from thresholding is the compression of signals. Indeed, the evaluator is required to collect as much traces as he can in order to know whether the cryptographic implementation can be broken by an attacker. In practice, the acquisition of SCA traces, which is usually performed by an oscilloscope, requires a high storage capacity especially when countermeasures are implemented. Indeed, from the security evaluation view point, the whole cryptographic operations (*e.g.* the sixteen rounds for a DES implementation) should be included within each trace acquired. This increases the need for storage capacity specially when taking into account all traces acquired. Moreover, a high sampling rate is usually used when acquiring traces; thus more memory resources are needed. In this insight, the compression of signals that involve a big amount of information, has been recently the focus of extensive research. One of the most efficient solutions is to use DWT. The idea behind compression is to remove redundancies from the signal and keep only the useful information, in a real time manner (*i.e.* before storage). From the SCA perspective, the useful information can be defined as the power or electromagnetic consumption related directly to the activity of the cryptographic process, resulting in the recovery of the secret key. In the litterature, Wavelet transforms based signals compression is basically performed using the notion of *threshold*, detailed previously. We note that 'Donoho combined MI' algorithm that we have proposed for the denoising context, can not be applied here. Indeed, our algorithm needs the entire set of traces acquired to be computed,

unlike the λ_{Donoho}'s threshold that is computed for each trace acquired. Several metrics such as *the zero ratio* and *the retained energy ratio* [134], have been proposed in the litterature, to assess the compression quality. However, when dealing with signals compression, the thresholding usually requires additional coding techniques like *the Huffman coding* [152], which slow down the compression process. From the SCA perspective, and specifically for unprotected implementations, we propose to keep only the approximation coefficients and reject the details. This way, the compression process is fast and the storage capacity gain is fixed. As experiments, we have measured the mutual information, in the same way as done for generating the Fig. 3.17, for different implementations (DES and AES) and for different levels of DWT decomposition. As a result, we have realized that the sensitive information, within SCA traces, is basically represented by low frequencies. Therefore, the compression of traces, when taking into account only the approximations, is performed without loss of information. Indeed, in the litterature of multiresolution analysis, it has been often reported that a loss of information could happen only when reconstructing the signal. For this purpose, the so-called *bi-orthogonal wavelet family* is often recommended for the reconstruction. Such compression can be very useful when used for unprotected implementations. However, during our experiments, we have noticed that, for protected implementations, sensitive information can not be negligible in the details' region.

3.2.3.3 Conclusion

The theoretical basis of Wavelet transforms have been detailed in the first Chapter 2.6 of this thesis, in which Wavelets were used in the very core of the attack. In this Chapter, we have shown how Wavelet transforms based Side-channel existing pre-processing techniques can be improved. First, we have proposed to use the 2D-representation, generated by Continuous Wavelet Transform (CWT), to efficiently detect and extract the cryptographic patterns from Side-channel traces. Second, we have presented a methodology to improve the pre-processing of signals acquired, from the Side-channel context. Indeed, this methodology is a combination between Donoho's threshold and mutual information (MI). Eventually, we have highlighted the way to reduce the storage capacity, needed to memorise Side-channel traces.

3.2.4 Electromagnetic Shielding

3.2.4.1 Electromagnetic Signals: General Background

The evaluator, when performing the acquisition of Side-channel traces, has basically at his disposal two possible technical solutions to spy the information leaked from a cryptographic implementation: a power or an electromagnetic (Em) acquisition. In the open litterature of SCA, scientific papers have often dealt with power acquisition although the usefulness of Em acquisition in improving the effectiveness of the analysis. In this part of thesis, we give more in depth discussion about Em acquisition, study the undesirable effects of external noise sources and provide the evaluator with a material solution to improve his measurements acquisition.

Digital integrated circuits are built out of individual transistors which dissipate power by charging the various capacitances whenever they are switched. The current, that flows across the transistor substrate when charge is applied to the gate, produces electromagnetic emanations. The generated electromagnetic (Em) field can be easily eavesdropped by the evaluator using inductive probes (antennas) which are sensitive to the electromagnetic impulses. According to Faraday's laws of induction, the Em antenna output voltage is computed as $V = -\frac{\partial \phi_m}{\partial t} = \oiint \overrightarrow{B} \cdot \mathrm{d}\overrightarrow{S}$, where ϕ_m is the magnetic flux through the surface, $\overrightarrow{B}$ is the magnetic field and $\overrightarrow{S}$ is the surface of the antenna. The electromagnetic wave theory involves two types of zone (or regions): the far zone, where electric and magnetic fields are mixed together and obey to the relationship $\frac{E}{H} = 377\ \Omega$, and the near zone, where, according to the topology of the source, one of the two electric or magnetic fields will be dominant. If we call $'d'$ the distance from the antenna to the source, the limit between the two zones is depends on the dominant wavelength λ_m emitted by the source and the features of the used antenna. Generally, the limit between the two zones is considered to be at a distance $d = \frac{\lambda_m}{2\pi}$. Therefore, when setting up an Em measurement, the evaluator is required to take into consideration the type of the used antenna and the distance from the antenna to the source. In fact, the antenna should be placed as close as possible to the source in order to decrease the contribution of surrounded external sources and because the source itself emits with few power. As discussed before, Em measurements must be often made in the presence of magnetic, electric or both fields which can induce an electromagnetic noise. Moreover, differential electromagnetic analyses are very sensitive to the magnitude of Em signals acquired. From the evaluator point of view, as discussed

in the previous part of this thesis, when performing an electromagnetic acquisition, it is necessary to minimize the contribution of the external sources, relatively to the considered source. Those external sources concern nearby circuits of the board, exterior lighting, electrical wiring, *etc.* Basically, we distinguish two independent types of electromagnetic noise: electrostatic and magnetic. Generally, the external noise sources combine the two noise types, which makes the noise reduction problem complicated. In academic cases, we recall that the noise is reduced by averaging the *Em* signals, since we are free to acquire as many measurements we want to perform a successful attack. However, in real life work conditions, the evaluator might be limited by the number of measurements. Furthermore, for some protected cryptographic implementations such as masked algorithms [9], the evaluator is not allowed to average the *Em* signals.

3.2.4.2 Electromagnetic Shielding Overview

In the Side-channel context, the leakage is passively and particularly observed via *Em* emanations [6], *etc.* In order to decrease the contribution of external sources, we propose to surround the considered source (*i.e.* the circuit during the encryption process) with what is known as a *Faraday cage*. Then, the *Em* acquisition is performed inside the shield. We note that the Faraday cage (or shield) has been already mentioned and/or used in the Side-channel context for different applications [67; 100]. Conceptually, the shield is a barrier to the transmission of electromagnetic fields. Moreover, the effectiveness of the shield can be defined as the ratio of the magnitude of the incident magnetic (or electrostatic) field on the barrier to the magnitude of the transmitted magnetic (or electrostatic) field through the barrier. When building a shield, three points should be taken into consideration by the evaluator:

1. The rifts of the cage that allow the penetration of *Em* emanations, which *"is limited to oscillations that have wavelength shorter than two times the diameter of the opening"* [46].

2. The discontinuity of the shield: the access through an opening might corrupt the proper functioning of the shield .

3. The electrostatic and magnetic features of the cage.

Our cage is built with steel material and covered internally by a thin aluminium sheet. This combination between two materials aims to improve the efficiency of the shield against the two types of noise. Indeed, the steel material is known for its high magnetic permeability, which ensures the magnetic shielding [114]. In the litterature, the

magnetic permeability is defined as *"the material's ability to acquire high magnetisation in a magnetic field"*. In addition, the aluminium has a high conductivity, which ensures the electrostatic shielding. In fact, the electrical conductivity is a measure of the material's ability to conduct an electric charge. Besides, the *Faraday cage* can be used for protection purposes. In fact, one can imagine a cryptographic circuit blinded by such cage in order to block any *Em* emanations dissipated form the circuit. This way, the circuit is protected against *Em* acquisition-based SCA.

3.2.4.3 Experiments & Results

Our measurement setup consists of one Xilinx Virtex–II Pro FPGA soldered on SASEBO platform [129], an 54855 Infiniium Agilent oscilloscope with a bandwidth of 6 GHz and a maximal sample rate of 40 GSa/s, antennas of the HZ–15 kit from Rohde & Schwarz. One picture of our *Em* measurement setup is shown in Fig. 3.19. The board is taken backside, because we noticed that the most leaking components were not the FPGA itself, but the decoupling capacitors that supply it with power. Those capacitors are surface mounted components (CMS) that have a fast response time, and thus radiate useful information about every distinct round of the algorithm. Moreover, they are easily accessible altogether with a large coil-shaped antenna. Therefore the *Em* leakage of the entire FPGA is captured without precise knowledge of the placement information within the FPGA.

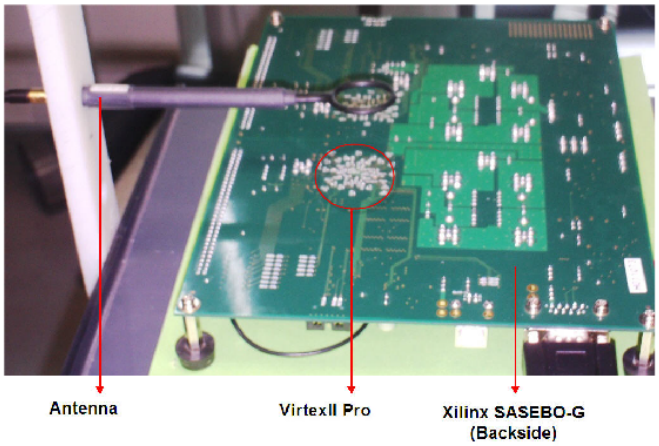


Figure 3.19: Em measurement setup.

As discussed before, the space surrounding the considered source can be divided into a far zone and a near zone. For this purpose, we show the effectiveness of the shield against electromagnetic noises by minimizing the contribution of far sources. Then, in

the near zone, we describe how the shield can be useful to enhance a DPA performed on Em Side-channel traces (this is often termed by DEMA: Differential Electromagnetic Attack) by decreasing the noise, generated by nearby sources.

In order to achieve the first part of the experiment, we used a dosimeter [12] to measure the source's exposure to the electromagnetic field in the experimental environment. Fig. 3.20 and Fig. 3.21 shows the detected frequencies with and without the shield. Obviously, the shield removes totally or partially the contribution of high frequencies (> 80 MHz), in the far zone.

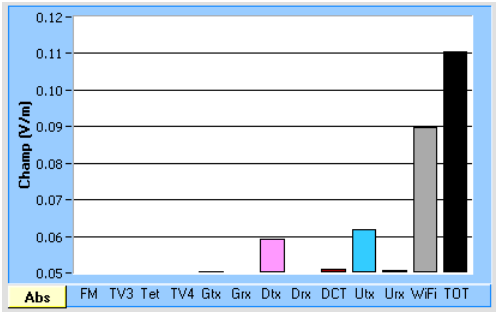


Figure 3.20: Detected frequencies without shielding.

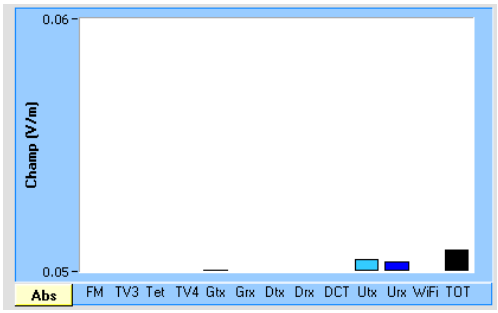


Figure 3.21: Detected frequencies with shielding.

The second part of the experiment is devoted to the DEMA. For this purpose, Em measurements were performed inside the shield, which is connected to the board electrical ground. As shown in Fig. 3.22, the penetration of the probe through the shield is allowed by a small opening. The DEMA was performed on an unprotected AES-128 implementation. In order to find the entire 128-*bit* key, the analysis was performed on the sixteen AES Sboxes. For each attacked Sbox, we retrieved one 8-*bit* subkey.

Measurements were performed for 10 000 Em signals, then we launched an *Em* analysis. As shown in the Fig. 3.23, for non averaged Em signals (denoted as "1x avg"), five subkeys over sixteen were revealed when using the shield. Whereas, for the basic attack (*i.e.* without shielding), we could not retrieve any subkey.

Besides, we note that the number of revealed subkeys is getting high with averaging, for both cases. After averaging *Em* signals 1024 times, all subkeys were retrieved. Thanks to shielding, we realized that we need only 696 *Em* signals to break all Sboxes (here, Stability Criteria metric was used 1.5.4). On the other hand, 1579 measurements were necessary to perform a complete (successful) *Em* analysis. Obviously, the shield enhances the quality of measurements. Therefore, DEMA with shielding is more efficient than the basic attack. Now, we recall that the o-th order success rate is defined

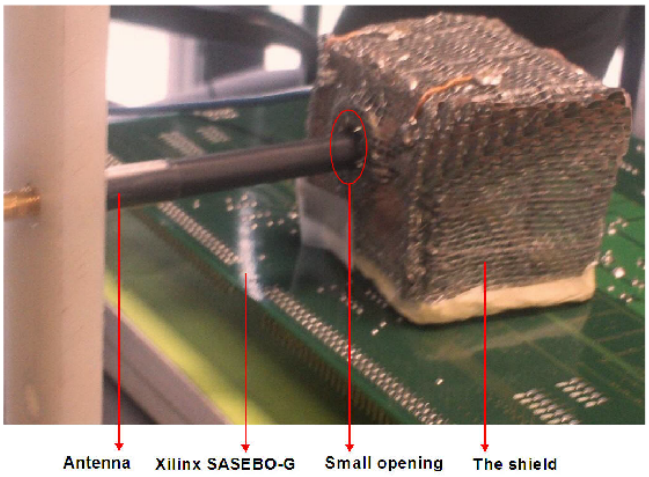


Figure 3.22: Shielding technique setup.

as the probability that an attack succeeds in recovering the correct key amongst the best o guesses. In our setup, this means that the targeted subkey shall be ranked at least o-th amongst the 2^8 candidates. The o-th order success rate can be estimated heuristically by repeating several analyses on different measurement campaigns comprised of the same number of traces. In our evaluation, we have carried out only series of 10 000 measurements with a single key, considered arbitrary. However, given that the attacked algorithm is an AES, where all the substitution Sboxes are identical in functionality (they all compute the SubBytes look-up described in the NIST standard FIPS 197 [4]) and that are evaluated in parallel, we can consider that the attack on each Sbox is independent. As for our experiments, the guessed round key (that of the last round) is different for each of the sixteen bytes of the state, we are actually conducting sixteen similar attacks concomitantly. This statement is a consequence of an "ergodicity" property: it holds because the outcome of an evaluation is not expected to change depending on the date at which it is conducted. So leading all the evaluations in parallel or sequentially should yield equivalent results. Therefore, the number of broken Sboxes (*i.e.* that for which the exact key byte has been recovered) amongst the sixteen ones when the number of observations is equal to 10 000 (displayed with hachures in Fig. 3.23) can be reinterpreted as the success rate of order $o = 1$ (multiplied by sixteen). As the number of broken Sboxes is always larger with a Faraday cage than without, we can thus conclude that the First-order success rate is greater with a Faraday cage. This result is shown in Fig. 3.24.

Eventually, we emphasize that our prototyping experiments were done with a home-

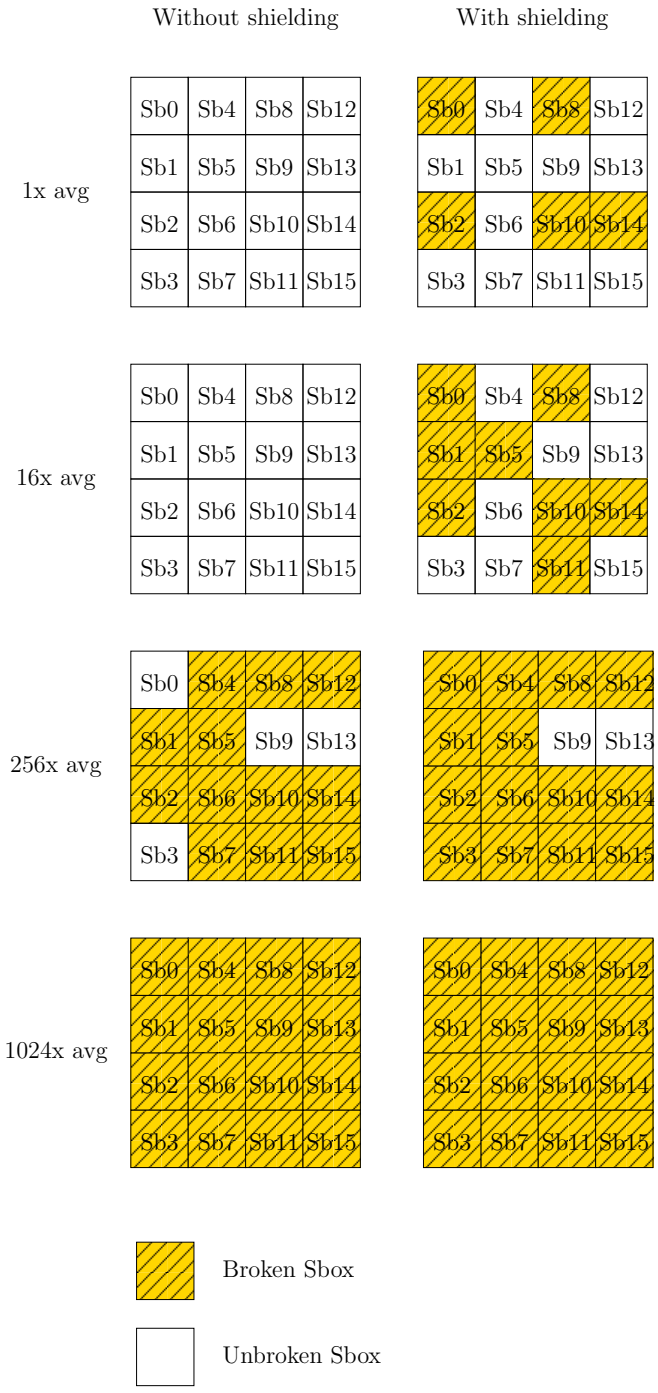


Figure 3.23: AES Sboxes attack using the electromagnetic shielding.

made shield. Fig. 3.25, 3.26 and 3.27 show a new shield prototype, that we have

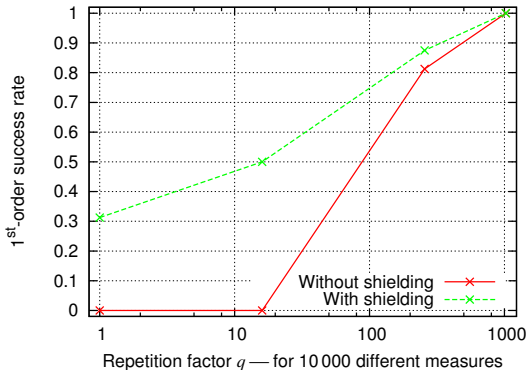


Figure 3.24: Firs-order success rate metric.
Success rate when breaking AES with $10\,000 \times q$ measurements, for a repetition factor $q \in \{1, 16, 256, 1\,024\}$.

conceived. This prototype provides more options to the evaluator, as it allows the use of different types of antennas (Fig. 3.28) with different positioning (horizontal/vertical). Moreover, it allows the evaluator to check the right positioning of the antenna through small openings located in the top of the shield. The difference with the previous shield (Faraday cage) relies on the rifts that allow the penetration of the *Em* emanations; and therefore to cancel the amplifications generated by the inner wall (*i.e.* the interior of the cage). Nonetheless, the amplification, caused by the interior reflections, could be useful when the undesired sources signal are negligible compared to the source signal containing the sensitive information. Indeed, when using the new prototype, we noticed the amplification of the signal; but we could not know whether this amplification would be useful for the analysis. Such prototype can be nicely improved by covering the interior surfaces with what is known as *Radiation Absorbent Material (RAM)* to cancel the interior reflections. Such design is known as *Anechoic chamber* [60]. An example of an *Anechoic chamber* is shown in Fig. 3.29. Eventually, all results obtained with different shields could be enhanced with an improved commercial-grade shield.

3.2.4.4 Conclusion

In this Chapter, we have given more in depth view of adapting the basics of Electromagnetic Shielding theory, which particularly aims at decreasing the contribution of external noise sources, to the Side-channel context. Generally, a perfect shielding is considered to be the most appropriate material solution to reduce the electromagnetic noise. All the same, shielding needs to have space to accommodate the experiment.

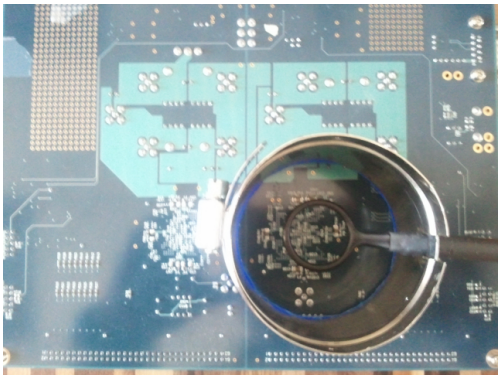


Figure 3.25: Top view of a horizontal positioning Em antenna.

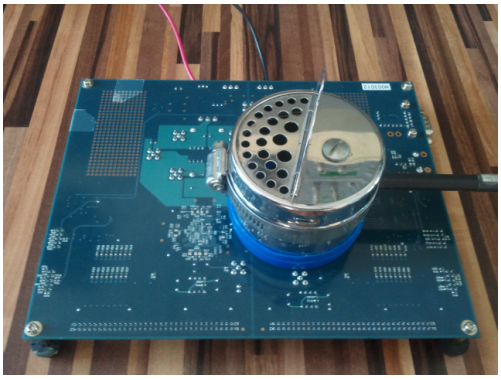


Figure 3.26: Front view of a horizontal positioning Em antenna.

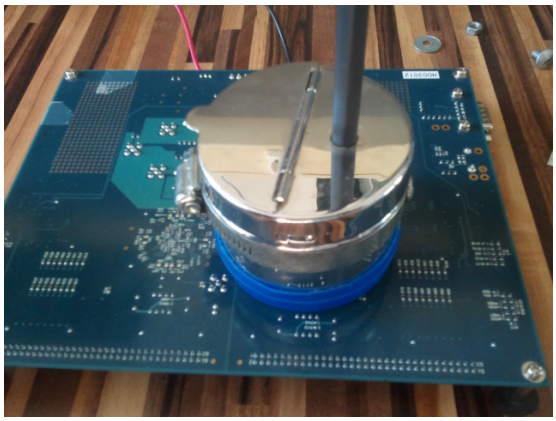


Figure 3.27: Front view of a vertical positioning Em antenna.



Figure 3.28: An example of several Em antennas.

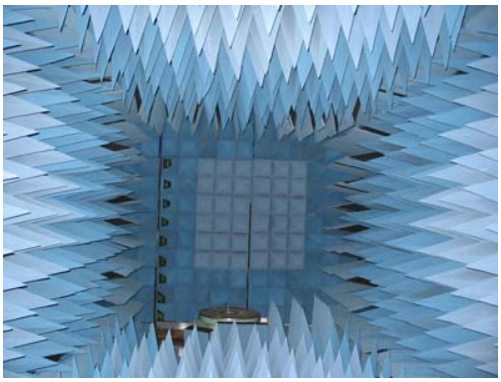


Figure 3.29: An example of an Anechoic chamber (this picture is taken from [13]).

3.3 Side-Channel Signals Re-synchronization

As stated in the introduction of this Chapter, the evaluator might be essentially faced with two major SCA problems that are intimately connected: the noise reduction problem and the misalignment of traces. In this part part of thesis, we will give particular attention to investigating new algorithmic solutions to the common problem of aligning Side-channel traces. Indeed, we put forward an innovative re-synchronization algorithm, called RM, and show its efficiency compared to existing techniques.

3.3.1 Related Work

In SCA literature, only few re-synchronization algorithms have been offered on the common problem of aligning SCA traces. Moreover, the most interesting algorithms have only recently been presented to the cryptographic community. We mention the Phase-Only Correlation (POC), which is presented by Homma *et al.* [61]. This technique, which was initially used in the computer vision and fingerprint recognition field, employs phase components in the frequential domain using the Discrete Fourier Transform and makes it possible to determine the displacement errors between signals by using the location of the correlation peak. Recently, a new technique, based on the Dynamic Time Warping (DTW) algorithm, has been presented by J. van Woudenberg *et al.* [153]. DTW is an approach that was historically used for speech recognition that has the advantage to work with traces that have different sizes. However, there is no common method to align a set of traces with this algorithm since it is basically used to measure the similarity only between pairs of traces. Moreover, it needs a parameter to trade off between the speed and the quality of the re-synchronization. Recently, S. Guilley *et al.* have proposed a cross-correlation based re-synchronization algorithm [57], namely AOC (Amplitude Only Correlation), and an intermediate algorithm called threshold-POC (T-POC). T-POC involves a parameter $\epsilon \in \mathbb{R}^+$; depending on the value of ϵ, T-POC is rather close to POC or to AOC. This re-synchronization algorithm shows its efficiency particularly when cryptographic countermeasures are deployed. In what follows, we will focus only on the most commonly used re-synchronization algorithms that are POC and AOC.

3.3.2 Effect of Traces Misalignment on SCA

In order to highlight the effect of the re-synchronization process, we created three increasing displacements $disp_1 < disp_2 < disp_3$ between initially aligned traces, as shown in Fig. 3.30 (displacements are highlighted with yellow color). Then we performed a CPA and observe the differences. We define a displacement ($disp_i$) as a random number of time samples shifted left or right. We carried out our experiment on unprotected DES power consumption traces. Obviously, as depicted in Fig. 3.31(Guessing entropy metric), the DES implementation is easily breakable by CPA. Indeed, according to CPA_{ref}, around only 250 traces are needed to reach the best rank. Although the CPA still manages to recover the secret key for a small displacement value ($disp_1$), its sensitivity is clearly getting higher when increasing the value of the displacement ($disp_2$, $disp_3$).

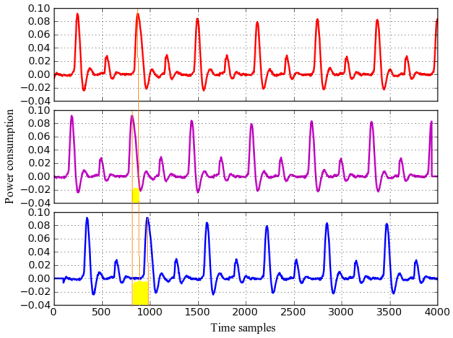


Figure 3.30: An illustration of misaligned traces

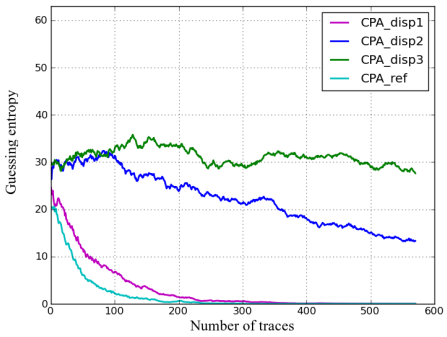


Figure 3.31: CPA Guessing entropy on misaligned traces.

From the theoretical point of view, we assume the misalignment results from a displacement of the traces by a number of time samples in the interval $[\![0, t[\![$. We say that $t \in \mathbb{N}^*$ is the size of the misalignment window. Then, in the extreme case where the misalignment is uniformly distributed over $[\![0, t[\![$ (which is almost achieved by [34]), the correlation ρ between the traces and a leakage model with the misalignment is equal to $1/\sqrt{t}$ times that without any misalignment. Now, the speed of a CPA is directly linked to these correlation coefficients. In fact, S. Mangard in [85; 86] proposed an interesting hypothesis testing approaches based rule that aims at estimating the correlation coefficients that occur in CPA without actually performing the attack in practice. This rule is evoked previously in this thesis and given by Eqn. (1.16). For low values of Pearson coefficient ρ, the Eqn. (1.16) is $\propto \rho^{-2}$. Therefore, all in one,

the number of traces N to break a cryptographic implementation with a misalignment window t is roughly multiplied by $\left(1/\sqrt{t}\right)^{-2} = t$. Consequently, the higher the size of the misalignment window, the lower the efficiency of CPA is.

3.3.3 Re-synchronization by Statistical Moments

An interesting statistical moments based technique for signals alignment has been firstly developed by James [65], in the context of Signal Processing. In what follows, we explore this technique and we test its adequacy for Side-channel Analysis.

3.3.4 Statistical Moments Based Jame's Method Principle

Let $X_0(t)$ and $X_1(t)$ be two misaligned traces. By considering the acquisition process, each trace has a temporal basis. Formally, X_0 and X_1 are the discrete time digital representations of different continuous quantities, denoted respectively by $\mathcal{S}_0(t)$ and $\mathcal{S}_1(t)$. We consider that acquired traces are misaligned when their temporal basis are different. In the general case, the traces can be represented by a triplet $(\mathcal{S}_i(t), t_i, d_i)$ with $i \in \{0, 1\}$, where t_i is the instant such as $\mathcal{S}_i(t_i) = X_i(0)$ and d_i is the number of clock cycles within $X_i(t)$, which is related to the resolution of $\mathcal{S}_i(t)$). Thus, $X_i(t)$ is seen as a window, with t_i and d_i which respectively set the shift and the zoom on $\mathcal{S}_i(t)$. In what follows, the couple $\mathcal{B}_i = (t_i, d_i)$ will be called the temporal basis of $X_i(t)$. In this case, there exists two functions W_0 and W_1, called warping functions, such as $X_0(W_0(t))$ and $X_1(W_1(t))$ have the same temporal basis. In our context, the warping functions are first order polynoms, $W = a + b \cdot t$, where a and b are the shift and the zoom coefficients respectively. For sake of convenience, $\mathcal{S}_i(t)$ can be reduced to the corresponding union of all traces:

$$\begin{aligned} &\text{Let } t_0 = \min_i (t_i) \text{ and } t_1 = \max_i (t_i + d_i) \ , \\ &\text{hence:} \\ &S_i(t) = 0, \ \forall t \in]-\infty, t_0] \cup [t_1, +\infty[\ . \end{aligned} \tag{3.57}$$

Basically, the problem of re-synchronizing two traces is to set them with a common temporal basis. We suppose now that the misaligned traces have different temporal basis but have exactly the same support $\mathcal{S}(t)$. The main idea behind the James method is to mark a point of reference R_i on each $X_i(t)$. R_i is a temporal mark, which is different for each acquired (discrete-time) trace, but corresponds to the same continuous time on $\mathcal{S}_i(t)$. Thanks to R_i, the warping functions are deduced and the different temporal basis are transformed to a single one. In [65], the author proposes to use the statistical

moments to unwarp all the traces to a single temporal basis. If the warping function W is linear, only the first and second moment (mean and variance) are needed. Let $\{X_i, 0 \leq i < N\}$ be the misaligned traces and $\{Y_i, 0 \leq i < N\}$ be the re-synchronized one. First, each trace is smoothed and weighted by some filters and weighting functions. Thus, a new set $\{\tilde{X}_i, 0 \leq i < N\}$ of traces is built. This step aims to emphasize the characteristics of the main pattern within traces X_i. James proposed some weighting functions as $I_X^{(m)}$, I_X^{min} or I_X^{max} defined as follows:

$$I_X^{(m)}(t) = \frac{|X^{(m)}(t)|}{\int |X^{(m)}(s)| \, \mathrm{d}s} \ , \tag{3.58}$$

where $X^{(m)}$ is the m^{th} derivative function of X.

$$I_X^{min}(t) = (\max(X(t)) - X(t))^r \ , \tag{3.59}$$

$$I_X^{max}(t) = (X(t) - \min(X(t)))^r \ . \tag{3.60}$$

$I_X^{(m)}$ allows us to concentrate weights on the shape of the pattern, while I_X^{min} and I_X^{max} respectively weight on the global minimum and maximum of X when r tends to infinity. The second step is to compute the two first moments $\mu_{X_i}^{(1)}$ and $\mu_{X_i}^{(2)}$ of each $\tilde{X}_i$. They are defined as follows:

$$\mu_{X_i}^{(1)} = \int t \cdot \tilde{X}_i(t) \, \mathrm{d}t. \tag{3.61}$$

$$\mu_{X_i}^{(2)} = \int (t - \mu_{X_i}^{(1)})^2 \cdot \tilde{X}_i(t) \, \mathrm{d}t \ . \tag{3.62}$$

Note that $\mu^{(1)}$ is the way chosen in [65] to find a point of reference R, introduced above. With $\{(\mu_{X_i}^{(1)}, \mu_{X_i}^{(2)}), 0 \leq i < N\}$, we can compute the reference moments $(\mu_{\text{ref}}^{(1)}, \mu_{\text{ref}}^{(2)})$, *e.g.* as follows:

$$\mu_{\text{ref}}^{(1)} = \frac{1}{N} \sum_{i=0}^{N-1} \mu_{X_i}^{(1)} \ , \quad \mu_{\text{ref}}^{(2)} = \left(\frac{1}{N} \sum_{i=0}^{N-1} \sqrt{\mu_{X_i}^{(2)}} \right)^2 \ . \tag{3.63}$$

We now aim at finding $W_i(t) = a_i + b_i.t$ such as $\tilde{X}_i(W_i(t)) = \tilde{Y}_i(t)$ and $(\mu_{\tilde{Y}_i}^{(1)}, \mu_{\tilde{Y}_i}^{(2)}) = (\mu_{\text{ref}}^{(1)}, \mu_{\text{ref}}^{(2)})$. In other words, we have to find a_i and b_i such as the statistical moments of X_i tends to $(\mu_{\text{ref}}^{(1)}, \mu_{\text{ref}}^{(2)})$. We can determine them as follows:

$$b_i = \sqrt{\frac{\mu_{\tilde{X}_i}^{(2)}}{\mu_{\text{ref}}^{(2)}}} \ , \qquad a_i = \mu_{\tilde{X}_i}^{(1)} - b_i \cdot \mu_{\text{ref}}^{(1)} \ . \tag{3.64}$$

3.3.4.1 Adequacy for Side-Channel Analysis

Some difficulties arise when James method is applied in the SCA context. Indeed, all traces are different from each other. Actually, the data manipulated during an encryption is dependent on the plain text in input, which varies from one trace to another. Even if the same plaintext is used, noticeable differences between the acquired traces are often observed due to noise fluctuations. In practice, these differences imply a variation of $\mu^{(1)}$ and therefore an inaccuracy on the point of reference R. These noticings bring us to consider more carefully the step of smoothing and weighting. In fact, we can establish the two following criterions:

$$\tilde{\mathcal{S}}_i \approx \mathcal{S}_{\text{ref}},\ \forall i \in \{0, 1, \ldots, N-1\}, \tag{3.65}$$

and

$$\begin{gathered} \forall i \in \{0, 1, \ldots, N-1\},\ \forall t \in]-\infty, t_i[\cup]t_i + \hat{d}_i, +\infty[, \\ \tilde{\mathcal{S}}_i(t) \approx 0\,, \end{gathered} \tag{3.66}$$

where $\hat{d}_i$ is the delay for d_i clock cycles. With the choice of suitable filters and weighting functions, we aim at transforming each trace such as the corresponding weighted support $\mathcal{S}_i(t)$ are very similar from one to the others (according to the criterion (Eqn. 3.65)). The higher the similarity is, the lower the error of $\mu^{(1)}$ is. In the SCA context, traces are mainly periodic (*e.g.* with the measured activity of the clock) and in this case, there exists no weighting function which emphasizes a pattern, common to all traces. Indeed, a periodic trace can not satisfy the criterion (Eqn. 3.66) unless the trace is a null vector.

3.3.4.2 Resynchronization by Moments (RM): Proposed Algorithm

Since the problem comes only with periodic traces, a logical approach is to work, for each trace X_i, with a sub-window $\mathcal{T}_i$, such as $|\mathcal{T}_i| = T_i$ and $\mathcal{T}_i(0) = X_i(t_0)$, where T_i is the period of X_i and t_0 is a chosen index in $[\![0, n-1]\!]$ ($|X_i| = n$). From a set of traces $\{X_i, 0 \leq i < N\}$, we compute a new set of averaged and weighted period $\{\tilde{\mathcal{T}}_i, 0 \leq i < N\}$. As shown in Fig. 3.32, each $\tilde{\mathcal{T}}_i$ is represented around an origin point O, by associating polar coordinates to each point as follows:

$$\begin{gathered} \forall i \in [\![0, N-1]\!],\ \forall t \in [\![0, T_i - 1]\!]\ , \\ \tilde{\mathcal{T}}_i(t) \mapsto P_{i,t} = (\tilde{\mathcal{T}}_i(t), \tfrac{t \times 2\pi}{T_i})\ . \end{gathered} \tag{3.67}$$

Thus, all the $\tilde{\mathcal{T}}_i$ are similar up to a rotation. To each $\tilde{\mathcal{T}}_i$, we associate a new triplet

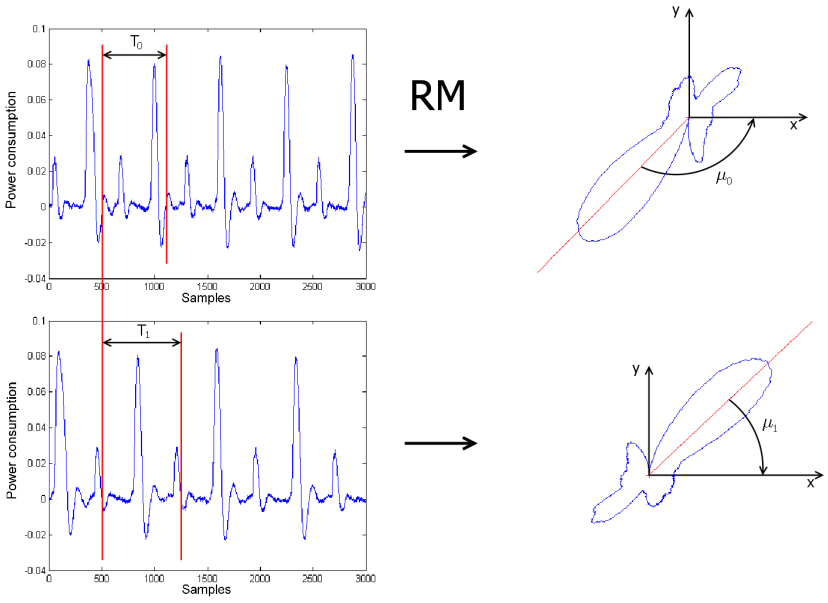


Figure 3.32: An example to computing $\mu^{(1)}$.

$(\mathcal{S}_i(t), t_i, d_i)$, where $\mathcal{S}_i(t)$ is a circular support, t_i is the angular distance from $\tilde{\mathcal{T}}_i$ to $\mathcal{S}_i(t)$ and d_i is now equal to T_i. With the knowledge of t_i and T_i, all the traces can be transformed such as they have a single temporal basis. Indeed, the warping function $W_i(t) = a_i + b_i.t$ is deduced with:

$$a_i = \frac{t_i \times T_i}{2\pi}, \quad b_i = \frac{T_i}{T_{\text{ref}}} \ , \tag{3.68}$$

where $T_{\text{ref}} = \frac{1}{N}\sum_{i=0}^{N-1} T_i$. We find t_i by searching a point of reference R_i on each period. We propose to find R_i by computing a 'circular mean', that is:

$$\overrightarrow{OR_i} = \sum_t \overrightarrow{OP_{i,t}} \ . \tag{3.69}$$

As depicted in Fig. 3.32, the vector $\vec{R_i}$, which is represented in red, coincides approximately with the maximum of the trace over the period. But, looking more in details, it is little offset: the reason is that the Resynchronization by Moment's method (RM) considers all the information contained in one clock period, thus gaining accuracy. By setting $\mathcal{S}_{\text{ref}}$ such as $R_{\mathcal{S}_i}$ has its angular coordinate equal to zero, the angular coordinate of R_i is then equal to t_i. Note that the RM method operates within a clock period.

Indeed, all clock periods involved are synchronized only when the possible shift is inferior than $\frac{T_{\text{ref}}}{2}$. In order to rectify these errors, the evaluator can use a giant step phase. It consists in choosing one trace X_{ref} as a reference and computing a cross-correlation between all the traces, by considering a step of T_{ref} temporal points. This phase is necessary to enhance SCA attacks. More details will be given in Sec. 3.3.5 that puts forward the efficiency of RM without the giant step phase by using the method of the least square modulo clock.

3.3.4.3 Link With POC

In the case of periodic traces, the circular mean defined in Eqn.(3.69) can be written as follows:

$$\mu_{\mathcal{T}_i}^{(1)} = \sum_{t=1}^{T_i} \tilde{\mathcal{T}}_i(t) e^{\frac{-2\pi j}{T_i} t} \, \mathrm{d}t \ . \tag{3.70}$$

Thus, $\mu_{\mathcal{T}_i}^{(1)}$ is equal to $\text{DFT}(\tilde{\mathcal{T}}_i(t))_k$, where $k = \frac{n}{T_i}$. Thus, while POC uses the phase of all component of $\text{DFT}(X_i)$, RM uses the phase of only one component of $\text{DFT}(\tilde{\mathcal{T}}_i)$, which corresponds to the highest amplitude (since X_i is periodic). Then, a_i is deduced with $a_i = \frac{\arg\left(\mu_{\tilde{\mathcal{T}}_i}^{(1)}\right) \cdot T_i}{2\pi}$. This implies that RM is $\log(n)$ times faster than POC. Furthermore, the RM benefits from any additional knowledge the evaluator has about the measurements. Typically, the evaluator is generally able to identify periods of interest. Thus, he can focus the analysis on them, which eventually leads to a reduction of the noise, especially when compared with a blind POC that would accumulate the noise of the whole trace.

3.3.5 Experiments, Results & Discussion

3.3.5.1 Evaluation Metrics

In order to compare the efficiency and the genericity of involved algorithms (AOC, POC and RM), we studied two kinds of time warp: time-shift and time-shift combined with a dilation (time-stretching). Moreover, for both cases of time warp, we simulated three levels of misalignment. From our point of view, an appropriate metric to evaluate the re-synchronization is to compute the standard variance of the re-synchronization error:

$$S = \frac{1}{n} \sum_{i=0}^{n-1} (s_i + a_i - \overline{m})^2 \ , \tag{3.71}$$

where s_i are the simulated shift, a_i are the shift deduced with the evaluated method and the new reference $\overline{m}$, which corresponds to the ideal a_i when $s_i = 0$. To validate this metric, we check the results provided by performing a CPA as a security metric. However, in practice, the S metric can not be computed as the evaluator does not know the ideal set. In this case, the problem is to quantify the level of misalignment of a given campaign and estimate whether a re-synchronization process is necessary. For this purpose, Gini coefficient (or index) [52], that is used as a metric to measure the statistical dispersion, is a suitable solution since it provides a value between 0 and 1. Indeed, Gini coefficient can be used to evaluate the amount of power consumption disparity within a set of traces. The higher the disparity is, the lower the value of Gini coefficient is. Hereinafter the formula:

$$G_t = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} |y_{it} - y_{jt}| \ , \tag{3.72}$$

where n is the number of traces and y_{it} is the set of power consumption values at a given instant t. The metric used in this study is defined by $G = \frac{1}{s} \sum_{t=0}^{s-1} G_t$, where G is the Gini coefficient averaged over all s time samples.

3.3.5.2 Experiments & Results

Our measurement setup consists of one Altera Stratix-II FPGA soldered on an SASEBO-B platform. We recorded two sets of 5000 Side-channel traces related to the activity of an unprotected DES crypto-processor. The averaging (256x) was performed on only one set of traces. The involved re-synchronization algorithms are R$\hat{\text{M}}$, RM and AOC presented in Sec. 3.3.5.2. R$\hat{\text{M}}$ is RM without the weighting phase (*i.e.* $\mu^{(1)}$ is directly computed with $\Im_i$). During our experiments, we have empirically noticed that the weighting function described in Eqn. 3.62 with r equals to 3 is a proper choice in the analysis.

Only Time-Shifting

Results regarding the "Only time-shifting" case are depicted in Tab. 3.2. The method of least square is computed modulo clock. We notice that AOC and POC provides a perfect re-synchronization when the traces are averaged before being processed. However, AOC's efficiency is lower than RM's one if the traces are noisy. POC is still a little better with noisy traces, but this is strongly dependent on the campaign (as seen in Fig. 3.34 with a different set of traces). Note that RM is always better when

applied with a suitable weighting function. Tab. 3.3 shows the number of traces needed to perform a successful CPA attack on noisy DES traces. Obviously, RM and POC have similar performances and are more efficient than AOC and R$\hat{\text{M}}$. Besides, we note that, during our experiments, when considering other sets of Side-channel traces, we noticed that the efficiency of RM and POC does not only depend on the amount of desynchronization but also on the amount of noise. Generally, we realized that POC is less robust than RM against the noise. Globally, results given in Tab. 3.2 are coherent with empirical attacks, which validates the correctness of the proposed metric (S). Besides, the metric of Gini (G metric) does not reveal any difference between compared algorithms, but allows the evaluator to distinguish whether a set of traces needs a re-synchronization. For instance, if traces are averaged then a threshold of 0.2 can be used. However, we can see that the relevance of G decreases when the noise increases. This fact regards the majority of disparity measures and is often reported in statistics books.

Table 3.2: Comparative results for the "Only time-shifting" case.

		Square metric		**G metric**	
		Avg	**Noisy**	**Avg**	**Noisy**
Misaligned	**1**	-	-	0.231	0.137
	2	-	-	0.282	0.140
	3	-	-	0.333	0.149
R$\hat{\text{M}}$	**1**	0.008	1.084	0.120	0.136
	2	0.009	1.171	0.121	0.136
	3	0.021	0.536	0.122	0.136
RM	**1**	0.003	0.295	0.123	0.136
	2	0.003	0.282	0.123	0.136
	3	0.011	0.251	0.123	0.136
AOC	**1**	0	0.405	0.124	0.136
	2	0	0.405	0.122	0.136
	3	0	0.405	0.123	0.136
POC	**1**	0	0.214	0.124	0.136
	2	0	0.214	0.122	0.136
	3	0	0.214	0.122	0.136

Time-stretching

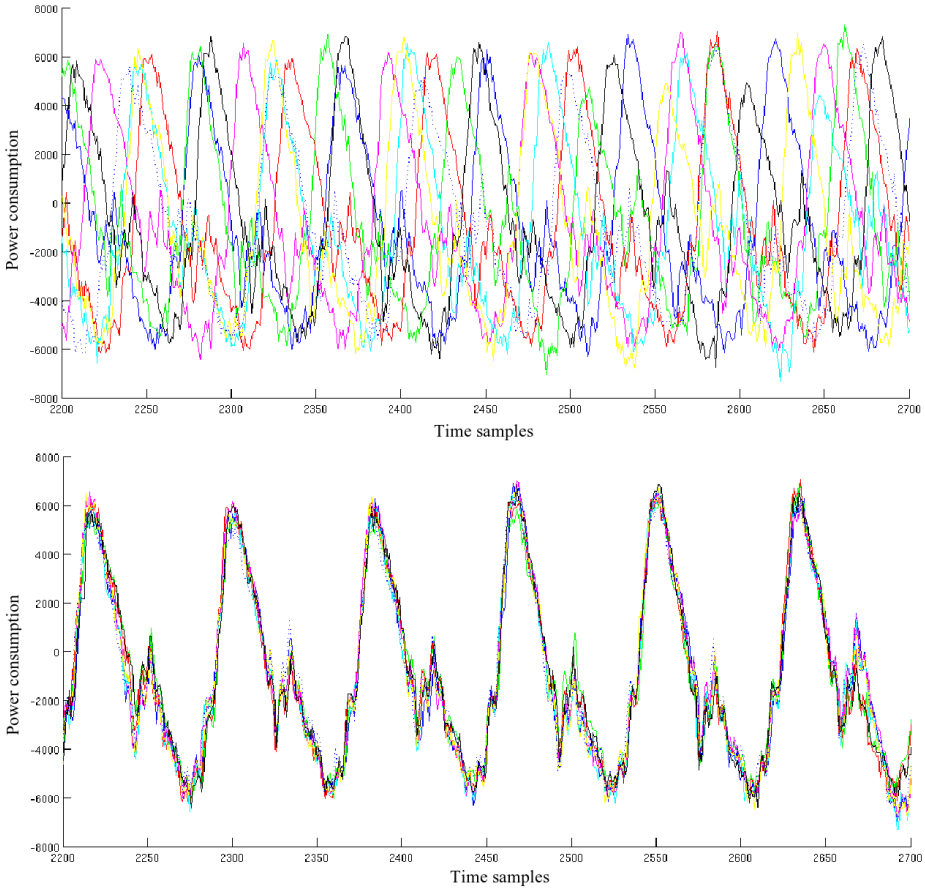


Figure 3.33: An illustration of DES traces re-synchronization using RM.

Table 3.3: Number of traces needed to succeed a CPA attack on the re-synchronized campaigns.

	Misaligned	**RM̂**	**RM**	**AOC**	**POC**
# Traces	>20000	12700	11300	12100	11200

Here, we discuss two kinds of dilation, denoted by 'fix' and 'E.E.' (Environmental Effect). In the first case, each trace is stretched with a fixed dilation coefficient b_i. It means that the period T_i is varying from one trace to another, but is not varying within one trace. In order to evaluate RM in this context, we simulate these dilations on the same traces used in Sec. 3.3.5.2. Tab. 3.4 shows the results of these experiments. The

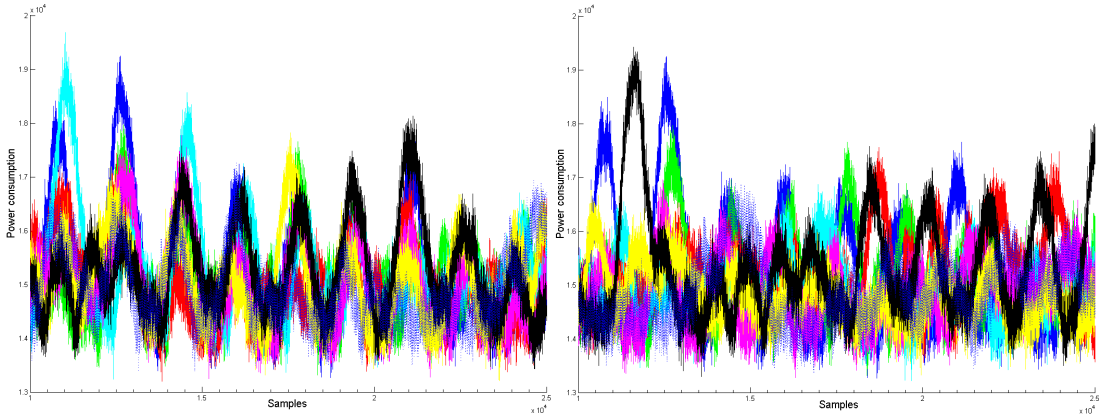


Figure 3.34: RM (left) versus POC (right) when performed on noisy AES traces.

Table 3.4: Comparative results for the "Time stretching" case.

		G **metric**	
		Avg	**Noisy**
Misaligned	**1**	0.341	0.157
	2	0.348	0.157
	3	0.354	0.157
$\hat{\textbf{RM}}$	**1**	0.127	0.136
	2	0.148	0.136
	3	0.137	0.136
RM	**1**	0.127	0.136
	2	0.137	0.136
	3	0.139	0.136

'E.E.' case aims at approaching the variation of power voltage or temperature which might occur during one acquisition campaign. These troubles do not affect the clock (so do not change T_i from a trace to another) but they speed up or slow down the computation during execution (and then affect $\mathcal{T}_i$). We propose a methodology to synchronize the misaligned traces by this kind of dilation. First, the evaluator has to profile the clock of the device. Thus, he constructs an averaged clock trace C, removes from each trace this clock component to obtain $Y_i(t) = X_i(t) - C$. Now, RM is able to align the Y_i. Indeed, although each $\mathcal{T}_i$ is different, we can use a circular standard variance to find the dilation coefficient. As $\mu^{(1)}$ can be expressed as a component of a

DFT (Eqn. 3.70), we define $\mu^{(2)}$ as follows:

$$\mu^{(2)}_{\tilde{\mathcal{T}}_i} = \sum_{t=0}^{T_i-1} \tilde{\mathcal{T}}_i(t) e^{(\frac{-2\pi}{T_i} t-\theta)^2 j} \, \mathrm{d}t \ , \tag{3.73}$$

where $\theta = \arg\left(\mu^{(1)}_{\tilde{\mathcal{T}}_i}\right)$. After RM processing, the evaluator is required to add the clock trace C to each analysed trace.

3.3.5.3 Discussion

According to the experimental results, it is clear that the evaluator is required to select the most appropriate re-synchronization algorithm for each case. Indeed, as illustrated in Fig. 3.35, he might be faced with two problems of misalignment: Time-Stretching or Only Time-shifting. In the case of Time-stretching, RM should be used to re-synchronize SCA traces. However, when traces are Only time-shifted and averaged, AOC and POC should be good choices for re-synchronization. Besides, AOC is still efficient when traces are noisy but less powerful than RM.

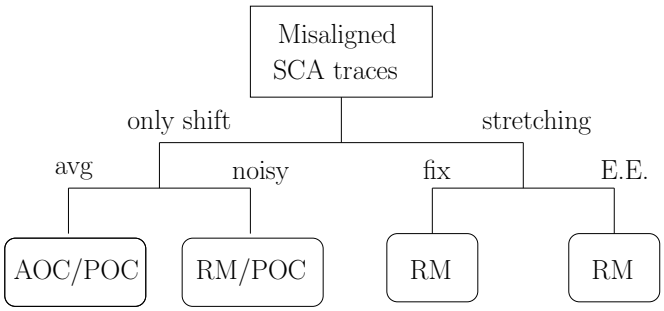


Figure 3.35: On the choice of the most appropriate re-synchronisation algorithm.

3.3.6 Conclusion

In this Chapter, we have proposed a new re-synchronization algorithm, namely RM, as a pre-processing step in Side-channel analysis. First, we have highlighted the importance of aligning SCA traces and surveyed existing techniques to get round the problem of misalignment. Second, we have put forward the theoretical principle of RM algorithm and validated its efficiency empirically with regards to the most common used techniques, Amplitude Only Correlation (AOC) and Phase Only Correlation (POC).

Chapter 4

Side-Channel Security Evaluation & Methodologies

4.1 Introduction & Contributions

Ascertaining the security of hardware circuits has always been a tough question and could pose a difficult long term problem. Actually, we are more and more surrounded by different forms of technologies such as mobile phones and smart cards that require an adequate level of security to work properly. Improper implementations of such systems could lead to the loss of sensitive and personal information. This situation is more critical when dealing with the military and defense market, which have always been ruled by high reliable devices such as ASICs and FPGAs. This thesis has focused on one of the most redoubtable attacks on embedded systems that are Side-Channel Analysis (SCA). We have seen that these attacks pose a serious practical threat to cryptographic implementations of secure devices, by exploiting unintentional physical leakage, such as the timing information, power consumption or radiated magnetic field. SCAs are passive attacks, in that the device under attack is not aware of its leaks being recorded. Thus, the risk is all the more pregnant as the device cannot protect proactively, but shall be secure even in front of an attacker that has almost unlimited time to perform his measurements. Clearly, dealing with this matter has become more important than ever. Besides, if a device, which has been evaluated as secure, could be attacked by such means, it might create a crisis of trust between vendors (manufacturers, embedded systems designers, *etc.*) and customers. In the world of Information Technology (IT), in particular embedded systems, security can be ensured by two approaches: the first one is to officially certify the embedded system product by a set of international security standards such as the **Common Criteria (CC)** [33] and the **NIST FIPS 140** [107]. The main goal behind standard certification is to obtain a valuable degree which validates the real security level implemented into the product. Obviously, standards are

implicitly ruled by needs such as marketing and business. Indeed, the certificate is no doubt an important factor in competitive market. The second approach is to asses the security robustness according to a set of analysis conducted by an evaluation lab ("non standard certification") and not necessarily documented in a formal standard: the security evaluation is often carried out through known attacks and often other security practices that are specific to the evaluation lab. These security practices are important since they reveal real vulnerabilities by performing more specific analyses, compared to official standards. Embedded systems vendors have often recourse to these evaluation labs in order to obtain a prior assessment of the security level ensured by their products, before applying for a standard certification. Generally, the lab evaluator is required to conduct its analysis following certain methodology.

Our contributions

In this Chapter, we give an example of methodology to be followed by lab evaluators, specifically when dealing with the security of embedded systems against Side-channel attacks. Our methodology can be seen as a prior step in evaluating a secure device before certifying it by FIPS or CC standards; and therefore enhance user trust. We propose an evaluation framework composed of five distinct phases which are characterization, simulation, acquisition, pre-processing and analysis. Each phase will be illustrated by practical examples and enriched by new evaluation methods and metrics. The study also highlights common errors made by evaluators and solutions to avoid them.

4.2 Certification Schemes & Standards: the Example of Common Criteria

CC is a security standard that is created in 1996 and originated from three main standards: the European standard Information Technology Security Evaluation Criteria (ITSEC), the Canadian Trusted Computer Product Evaluation Criteria and the Trusted Computer System Evaluation Criteria (TCSEC) which is created by the US Department of Defense (DoD). CC can be seen as a framework in which the relation between the three actors: **vendors - users - evaluators** is well defined and structured. Actually, CC is an international standard that first allows users to specify their security needs (or requirements). Second, it allows vendors to determine and announce the level of security involved in their products. Finally, evaluators intervene to evaluate these products by checking whether vendors are telling the true level of security about their products. Generally, users needs cover three security aspects: confidentiality, in-

tegrity and availability (explained previously in 1). In the literature, Common Criteria evaluation is based on four major definitions: Target Of Evaluation, Protection Profile, Security Target, Security Functional Requirements. Besides, CC provides seven levels of security called EALs: EAL1, the lowest level with least rigor of evaluation effort, to the EAL7, the highest level. For more details, we refer the reader to [1].

CC is very general standard that deal a wide range of IT products (hardware/software security), which **may** include cryptographic functions. Nonetheless, when comparing CC to FIPS 140, we note that FIPS 140 targets only cryptographic functions and algorithms. The CC checks whether the general security level matches the one specified by the vendor, unlike the FIPS 140 standard that requires a cryptographic system (or mechanism) to evaluate in a very specific manner. Besides, it is often reported that CC is costly, as it requires sophisticated and expansive equipments, and time consuming.

From the practical point of view, when evaluating a secure embedded system, two situations regarding the amount of knowledge of the implementation details, that are available to the evaluator, can be distinguished:

1. **Total knowledge** or **White box testing**: assumes that the evaluator knows the finer (complete) details about the cryptographic implementation (*e.g.* source design, data-sheets, *etc.*).

2. **Zero knowledge** or **Black box testing**: assumes that the evaluator has no prior knowledge of the cryptographic implementation. For instance, we can imagine that the type of the tested algorithm is unknown (non standard and non documented). In such situation, well-experienced evaluator is recommended.

In the literature, **Medium knowledge** situation or **Grey box testing** is often evoked. It assumes that the knowledge of the evaluator about the device under test is partial. For instance, the evaluator could not have access to both the plain text (the algorithm input) and the cipher text (the algorithm output); or could be limited by the number of measurement to disclosure. We can take as example the bitstream encryption mechanism (using 3-DES decryption algorithm) implemented into the Xilinx Virtex-2 FPGA. Indeed, one of the security functionalities ensured by the mechanism is that the number of measurements (using the same *secret key*) that could be acquired can not be superior than the number of encryption blocks within the design's bitstream. More precisely, the initial *secret key*, that is specified by the user, is modified by the Xilinx security mechanism to generate a second key. Then, this generated key is used to actually encrypt the bitstream. The mechanism is still efficient, in limiting the number of measurements that could be acquired, even if the algorithm used to generate the

second key from the user key is known by the evaluator; and this because the value of the generated key is dependent on both the user key and the design itself. Thus, for a given bitstream, there exists one and only one possible encryption key.

For all situations, the evaluators, often represented by one organism namely CESTI (ITSEF for the French organism), are permanently under the supervision of a what is know as **certifier** that aims principally at controlling the evaluation process. Four key concepts are strictly controlled by the certifier: the repeatability, reproducibility, the impartiality and objectivity ([1]).

4.3 Towards a Common Framework for Security Evaluation

4.3.1 Characterization Phase

This phase depends on the documentation and specifications provided by the vendor. It includes the ways to access the device, the type of the implemented countermeasures and the strategic choice in the analysis.

4.3.1.1 SCA Constraints

Generally, the evaluator is often required to conduct its analysis under two types of constraints: physical and algorithmic.

Physical Constraints

These constraints include countermeasures deployed to limit the access to the cryptographic implementation. First, the evaluator should find answer to the question of "what is the most appropriate measurement technique to spy the sensitive information leaked from the device (overwhelmingly Power or Electromagnetic Radiations, seldom Timing or Acoustic)". Moreover, he should know how to detect the start of the encryption (or decryption) process and if he can acquire as many Side-channel signals as wanted. Indeed, the most natural countermeasures consist in either increasing noise, which makes the number of acquired signals very high, or randomizing the instant where the cryptographic process is being performed. Besides, manufacturers often deploy different types of sensors and filters that mainly aim at protecting the secure implementation from improper manipulations that would threaten the cryptographic process. Those sensors and filters include the level of voltage, frequency, temperature or light. For instance, light sensors define a range of variation in which the gradient of light should be, otherwise the circuit resets. More sophisticated protections have been

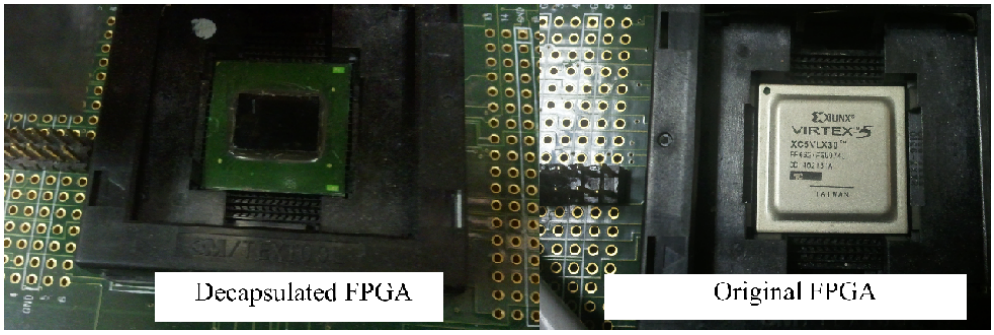


Figure 4.1: An example of decapsulated Virtex-5 FPGA (left).

proposed, such as using a robust metal enclosure that acts as a Faraday cage, invoked previously in Chapter. 3 or removing the decoupling capacitors that supply the cryptographic processor with "local" and "fast" power sources. In fact, first, shielding the circuit with metal layers reduces the electromagnetic (Em) radiations. Therefore, the Em acquisition and exploitation of Side-channel signals becomes more difficult. As a matter of fact, recent commercialised circuits (FPGAs and ASICs) are often protected by metallic surfaces (a thin layer of metal). These metallic surfaces aim, in the first place, to protect the circuit environment from the undesirable electrical effects caused by nearby components. Additionally, they are used to weaken potential attacks. As a practical example, we have verified that decapsulating the metal layer protecting the Virtex-5 FPGA results in better Side-channel analyses, specifically when the Em antennas are placed over the circuit (front side). A decapsulation of the FPGA is illustrated in Fig. **??**.

Algorithmic constraints

These constraints include provable countermeasures that depend on the basic operations used in the algorithm. A provable countermeasure satisfies two conditions. First, the countermeasure must be sound, meaning that in the framework of a given model, it can be demonstrated that its principle do indeed protect efficiently. Second, it must adhere to Kirchhoff's' principle: it shall work even if its rational is completely exposed. Two counter-examples are for instance the dummy cycles insertion, since it is not sound [32], and the code obfuscation [8], since it involves a secret method that is not expected to hold long against a determined attacker.
Basically, in this thesis, we have seen that there exist two provable countermeasures

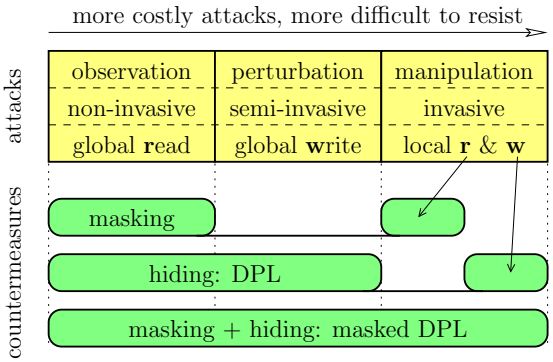


Figure 4.2: Coverage of countermeasures for all physical attacks classes.

that are often deployed: **information masking** [86, Chp. 9], which aims at randomizing the Side-channel, and **information hiding** [86, Chp. 7], which aims at balancing the Side-channel. In order to put the problematic of protecting cryptographic implementation by these countermeasures into its general context, we answer the question of what is the suitability of masking and hiding to thwart attacks. In Fig. 4.2, we give a graphical solution for such question. This figure shows that masking is also a countermeasure against probing attacks [63], since the value of the probed node becomes random. Also, hiding is a countermeasure against most fault injection attacks since the attacker erases the value stored redundantly in one pair of wires by changing only one of them. The case of symmetric faults is covered in [58] and of arbitrary faults in [21]. An interesting noting is that by associating masking and hiding, the protection extends to semi-invasive and invasive attacks. This association must be realized with care, since otherwise some attacks become possible, such as the "folding attack" [131] or the "subset attack" [98]. The synopsis of this attack merely consists in recovering the masking bit and then to defeat the hiding countermeasure.

4.3.1.2 On the Choice of the Most Appropriate Analysis (Leakage Models & Distinguishers Selection)

Through the different parts of this thesis, we have seen that Side-channels unfold according to a classical cryptanalytic scenario, that is mainly composed of:

1. A **leakage model** to manage the partitioning of the acquired Side-channel observations, which depends on the scenario (known/chosen plaintext/ciphertext),

algorithm (to explore the internal rounds by guessing manageable parts of the secret) and the implementation (software or hardware, pipelined or unrolled, protected or not, *etc.*)

2. A **distinguisher** to select the most relevant partitioning, amongst all the hypotheses on the secret. The distinguisher is basically a statistical tool, that aims at putting forward any bias. They can be for instance a difference of means [72], a covariance [59], a correlation (linear [24] or rank-based),mutual information [50] or variance [144].

In the SCA community, the question of "Which is the most appropriate strategy: looking for the optimal leakage model or the best distinguisher?" is recurrently asked informally. Nonetheless, this important question is rarely discussed in the scientific literature. Hereinafter, we give a clear understanding of the relationship between leakage models and distinguishers.

In the general picture, Side-channel analysis aim to best approximate the physical leakages (or measurements) with the corresponding intermediate values, over which a leakage model has been already applied (refer to 1.5.2.1). Formally, suppose we want to best approximate a random variable Y with another variable X based only on their joint distribution. The approximation problem, as studied previously in Sec. 2.2.4.1 is to seek for a function ϕ of X that best fits Y among all possible forms of ϕ. In our study, the variable X is deterministic since it is theoretically built from a known cryptographic process. Moreover, X represents the set of the intermediate values, over which a leakage model has been already applied. Whereas, the variable Y represents the set of the physical leakages (SCA observations or measurements). Thus, for sake of clarity, the variable X is called *the prediction* and Y *the observation*. In Fig. 4.3, we illustrate the connections between the variables X and Y, the leakage model and the distinguisher. In fact, two analysis aspects arise: first, the evaluator should know the optimal leakage model to best build the prediction X; second, he is required to find the most appropriate distinguisher to best quantify the true relationship, denoted by $\mathcal{R}_{X,Y}$, between the prediction X and the observation Y. Depending on the causal connections between X and Y, their true relationship may be linear or non linear. However, regardless the true nature of the relation, a linear model can always serve as a first approximation. Thus, the evaluator is required to think about analytical transforms (linear, normal, *etc.*) of X and Y. In Fig. 4.3, other points are highlighted. Indeed, the evaluator should analyse the leakage of all electrical nodes possible, which result in intermediate results of the cryptographic algorithm. In addition, he should distinguish two cases depending on the

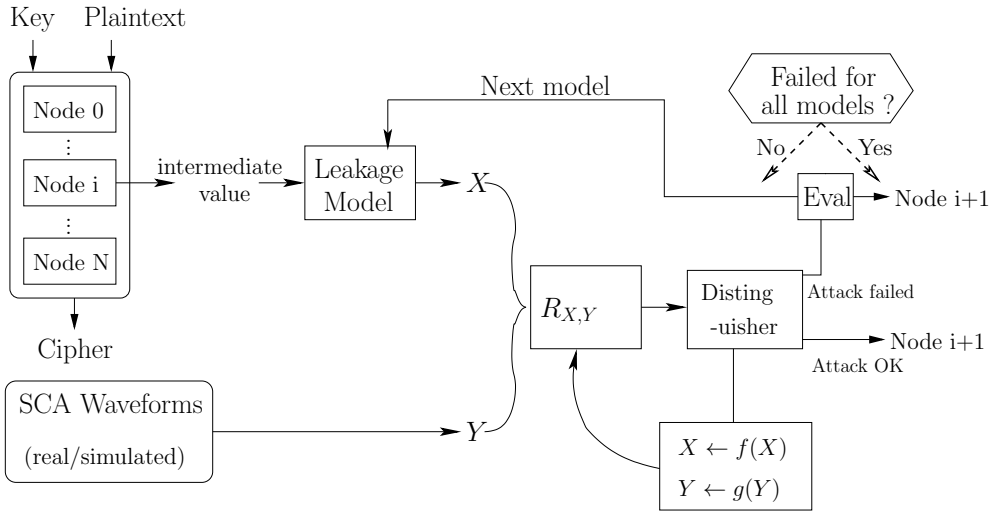


Figure 4.3: Leakage model and distinguisher connections.

success of the performed attack. In fact, when the attack fails, the evaluator is required to change either the leakage model or to find an appropriate transformation of X and Y. This task is exhaustive, in the sense that it describes all possible, rational solutions to analyse the true relationship $\mathcal{R}_{X,Y}$. Moreover, the evaluator should evaluate the degree of the statistical dependency between X and Y for each failure and by considering only the value of the secret key. In the perfect case, the dependency is null for all failures. In this case, the target node is definitely secure against the performed attacks. Note that some distinguishers could be used as tools to check the statistical dependency, such as the Pearson correlation coefficient under the Gaussian assumption [147].

4.3.2 Acquisition Phase

SCA traces are typically acquired by a digital scope. The accuracy of an oscilloscope can affect measurements greatly. Indeed, an oscilloscope with high quality display can reveal more important Side-channel signal details than one with low quality display. Thus, it is necessary to properly set up the scope before acquiring measurements. Generally, three parameters should by taken into consideration: the input bandwidth, the sampling rate and the resolution of the scope. Moreover, other equipments are needed to capture the activity of the tested device during the cryptographic process. These equipments consist of power measurement probes and antennas to measuring the electromagnetic (Em) radiations. The choice between performing power or Em acquisition is dependent on the implementation environment which includes the access

to the device and the surrounded electrical components.

4.3.2.1 A Practical Example

In order to show the effect of the sampling rate parameter, we conducted a practical application, which first consists in acquiring different sets of unprotected DES Side-channel traces with the same messages and the same secret key; but with different values of sampling rate. Second, DPA and CPA attacks, were performed on each set of traces. We placed ourselves in a situation in which the number of traces is limited to only 2000 traces. Our measurement setup consists of Xilinx FPGA soldered on SASEBO platform. One 54855 Infiniium Agilent oscilloscopes with a bandwidth of 6 GHz and a maximal sample rate of 40 GSa/s, amplifiers, antennas and probes of the HZ–15 kit from Rohde & Schwarz. We did the experiment for three different values of sampling rate (1 GSa/s, 0.5 GSa/s and 0.1 GSa/s) that can be tuned through the oscilloscope's panel. We noticed, as expected, the higher the sampling rate, the more efficient the attack. It is noteworthy that at low sampling rate (0.5 GSa/s), CPA is still successful, whereas DPA fails to recover the secret key. Moreover, for the lowest fixed sampling rate value (0.1 GSa/s), neither DPA or CPA is successfully performed.

4.3.2.2 Combination of Measurements

A common practice to carry out Electromagnetic Analysis (EMA [130]) is to acquire the strongest and most obvious leakage points on the device. However there are other points which leak information as well. We propose to acquire multiple simultaneous leakages from different leakage points. Multiple antennae can be used to acquire multiple leakages. These multiple leakages for a single activity could be combined for an efficient SCA. Multi-channel attacks have already been introduced in [7] for mono-bit DPA and Template attacks. In this study, we give a more generic outlook towards combing measurements using any distinguisher. In fact, this can be seen as a complementary work of combined Side-channel analyses, studied in the second Chapter (2.3). Besides, we also provide a metric based on information theoretic to test if the possibility of measurement combination exists for a given pair of traces.

Theoretical Background

Information gain of a single attribute X with respect to class C, also known as mutual information (defined previously) between X and C, measured in bits is:

$$Gain_c(X) = I(X;C) = \sum_x \sum_c p(x,c) \log \frac{p(x,c)}{p(x)p(c)} \ . \tag{4.1}$$

Equivalently:

$$I(X;C) = H(X) - H(X|C) \ , \tag{4.2}$$

where $H(X)$ is the entropy of X and $H(X|C)$ is the conditional entropy of X knowing C. To simplify the calculation of entropy we consider the distribution of X is Gaussian. In this case entropy can be calculated as a function of standard deviation σ_x of X as:

$$H(X) = \sum_i p(x_i) \log(p(x_i)) = \log(\sigma_x \sqrt{(2\pi e)}) \ .$$

This method might not be ideal for estimating entropy but works well in practice [113]. Nevertheless, other methods of estimating entropy can be applied. Information gain can be regarded as a measure of the strength of a 2-way interaction between an attribute X and the class C. 3-way interactions were introduced as interaction gain [64] which is equivalent to mutual information of 3-variables. Interaction gain is also measured in bits, and can be understood as the difference between the actual decrease in entropy achieved by the joint attribute XY and the expected decrease in entropy with the assumption of independence between attributes X and Y. Interaction gain can be considered equivalent to multivariate mutual information [48]. The Venn diagram representation is shown in Fig. 4.4.

$$\begin{aligned} I(X;Y;Z) &= I(X,Y;Z) - I(X;Z) - I(Y;Z) \ , \\ I(X;Y;Z) &= (D+F+G) - (F+G) - (D+G) = \textbf{-G} \ . \end{aligned} \tag{4.3}$$

As per Eqn. (4.3), interaction gain is equal to **-G**. If X and Y are independent, I(X,Y;Z) = I(X;Z)+I(Y;Z). This means the interaction gain I(X;Y;Z) is zero. Interpreting from Fig 4.4(a) and (b) combination is possible when the information equal to D is added to I(X;Z) with introduction of Y. This makes I(X,Y;Z) = D+G+F. If D is zero, then introduction of Y is not providing any extra information.

To check this condition we propose a simple test. The possibility of combination

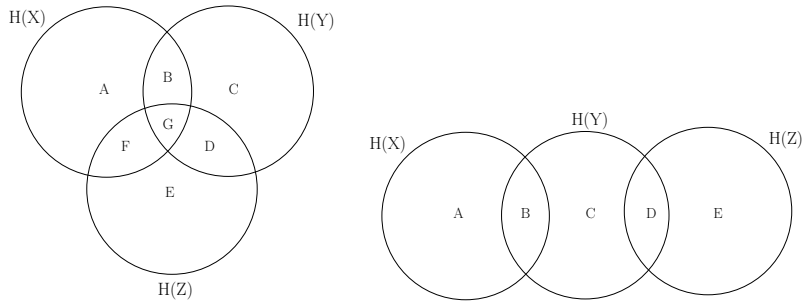


Figure 4.4: Venn diagram representation
: (a) possible combination, (b) not possible combination.

(PC) can be calculated as a ratio:

$$PC = \frac{Max(I(X;Z), I(Y;Z))}{I(X,Y;Z)}\,. \tag{4.4}$$

For a combination to exist PC should be lie between 0.5 and 1, where PC=1 will suggest no combination is possible. In context of combined attacks, interaction gain can be directly applied. This is a profiling step because knowledge of the secret key is required to calculate PC. Alternatively PC can also be used as a distinguisher. However, in this study we show how to apply combination using CPA.

Practical Results

The experimental setup used is same as in Sec. 3.3.5. We target two decoupling capacitors on the backside of the FPGA which show emanations corresponding to a DES execution. As the number of capacitor here is small, hit-and-trail method was efficient for choice of capacitance. We collect two sets of 5000 traces from two chosen capacitors for the same dataset. A crypto-processor is a bulky design and could be spread over different power banks in an FPGA which are terminated by different capacitors. Therefore different capacitor leak more information about a certain part of the circuit. Here the partition can be seen as different sboxes. We start with testing the possibility of combination. Fig 4.5 shows the ratio PC for two cases. We computed the value of PC for Sbox 0 in each case. Fig 4.5(a) shows considers traces from two capacitances which are leaking relevant information. It can be seen that the value of PC is close to 0.5 when the value of mutual information is relevant. Fig 4.5(b) considers traces from a leaking capacitance and another point which is not leaking. Here the value

of PC is close to 1. Here the value of mutual information of the two measurements is multiplied by 100 to visualize on the same scale as PC. This means that combination is possible for the traces in Fig. 4.5. Unfortunately, we did not deal with the extreme cases with our traces. In the other parts of the trace there is noise and the value of PC is randomly changing.

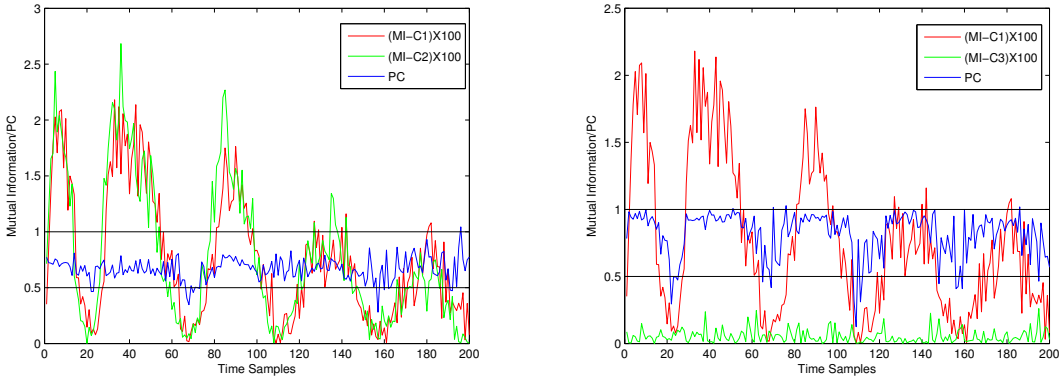


Figure 4.5: Calculation of PC for two cases when combination is (a) possible, (b) not possible.

Next step is to observe practical application of combination of measurement using a common attack like CPA. We applied CPA on the traces collected from C_1 and C_2 independently. Tab. 4.1 summarises the result of CPA on each set of traces. These results are averaged over 30 attacks. We see that C_1 is better suited for Sbox no. 0,1,3 and 7 and C_2 for the rest. Before testing the combination, we concatenate traces of C_1 and C_2 together. Traces can be normalised before concatenation specially when techniques like principle component analysis are applied but if the traces are taken with the same scale then normalisation will not help a lot. In our experiments the traces are taken with the same scale on the oscilloscope therefore normalisation is not needed. We launch an attack on the concatenated trace. The attack calculates the Pearson correlation coefficient of the key hypothesis for each trace on each of the two section of concatenated trace. To test the combination, we use an aggregate function Ψ as listed previously (Sec. 2.3.2.3). Precisely, the used aggregate function in this experiment is the Sum() on the calculated coefficient value. Spearman, Gini and other coefficients can be used but we want to demonstrate that even with Pearson coefficient the attack works. It is shown that Sum() can increase the SNR even if the two traces contain equivalent information. If the amount of information is not equivalent, Sum() will further increase the SNR hence a faster attack. Performance of Sum() as a basis for

combination has already been demonstrated in Fig. 2.9. The computation complexity is equivalent to processing a trace with twice the number of samples with minor overhead of applying the aggregate function. Two parallel attacks on non-concatenated traces will have similar computation overhead but concatenation makes it easy to manage the key hypotheses and apply aggregate functions. Tab. 4.1 shows the number of traces required to attack when combination is applied. We find that in each case the combination is better than individual attack and the gain varies from 4.16 to 44.86%. This also complies with our PC test. For each sbox we found PC to be inferior than 1 and a positive gain. The scale of PC and gain cannot be compared as each quantity is computed using a different method. As mentioned before some countermeasures change encryption key after a specific number of encryption to prevent SCA. Since the number of traces acquired is considered a scarce resource, we demonstrate that multiple measurements can be exploited for faster attack.

Table 4.1: No. of traces to attack using C_1, C_2 and combination of both.

Sbox No.	0	1	2	3	4	5	6	7
C_1	350	943	733	400	410	320	548	592
C_2	432	1073	720	980	176	281	551	192
$Comb_sum$	212	750	397	251	165	270	448	184
Percent Gain	39.42	20.46	44.86	37.25	6.25	3.96	18.24	4.16

4.3.3 Pre-processing Phase

In this thesis (3), we have seen that Signal Processing is a crucial step in the evaluation; as the main idea behind Side-channel analysis is to best detect and extract the secret information from signals. Moreover, we have shown that when assessing the robustness of a secure implementation, the evaluator might be faced with two Signal Processing problems: the noise problem affecting Side-channel traces and the de-synchronization (or misalignment) of these traces. These two cases were deeply studied in Chapter 3. Hereinafter, we pinpoint another important aspect of the analysis to consider before analysing Side-channel traces.

Side-Channel Window of Interest

In this section, we highlight the importance of selecting the right window, that we refer to as *window of interest*, when analysing Side-channel traces. The *window of interest* can be defined as the range of time samples covering only needed leakage instants, that

are sufficient to mount a successful SCA. The major advantage of the window selection is no doubt related to timing considerations. Indeed, on the one hand, it makes SCA faster as the size of processed data would be significantly reduced, instead of considering the entire cryptographic process. On the other hand, it accelerates the acquisition operation of Side-channel traces. The effect of selecting different windows sizes (Fig. 4.6) is depicted in Fig. 4.7 and Fig. 4.8. These two figures show the Success rate and Guessing entropy metrics, respectively, when a DPA is performed on an unprotected DES traces. Obviously, DPA is losing its performance when the window size is getting higher.

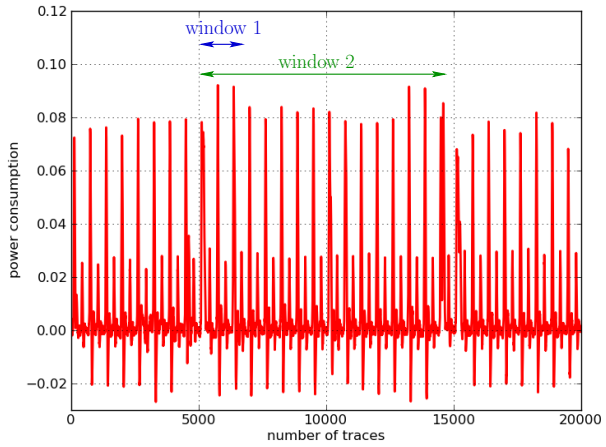


Figure 4.6: Illustration of windows selection process.

This can be explained by the increasing presence of noise when more information are taken into account rather than the real leakage covered by the smallest window. Moreover, the windowing process relaxes the memory depth parameter that is fixed through the oscilloscope and which allows more data storage memory. However, the evaluator is not always free to choose the window he wants. Thus, he must deal with the whole cryptographic process. Actually, the vendor, when having recourse to an evaluation lab, is free to keep secret some details about the cryptographic implementation. Consequently, the evaluator is bound to deal with such situation by considering the entire Side-channel trace (*i.e* all time samples) in his analysis.

4.3.4 Simulation Phase

The simulation phase is strictly dependent on the documentation provided by the vendor. This phase consists in predicting the behaviour of the cryptographic implementa-

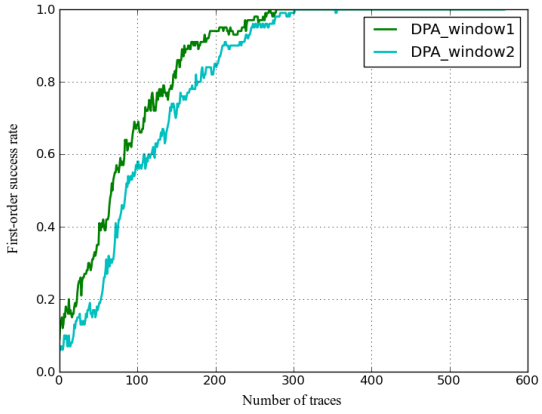


Figure 4.7: DPA First-order success rate on different windows.

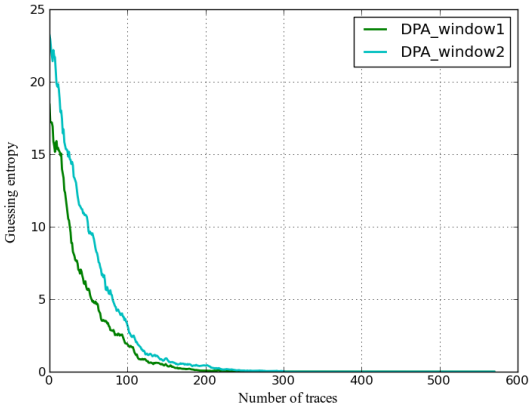


Figure 4.8: DPA guessing entropy on different windows.

tion by a software program. It replaces real components with idealized electrical models. Power simulations are crucial in that they reveal the existence of power leakages during the cryptographic process, which allows to make statements on the resistance against Side-channel Attacks. These simulations can be performed at different levels of power accuracy [86]: the analog level, the logic level and the behavioural level. However, this process is very time consuming specially when taking into account the whole implementation. Moreover, it requires attention to the smallest details about the cryptographic design such as transistor or cell netlists that are often classified as confidential; and therefore are not provided by the vendor. In case of non availability of the netlists, it is always possible for the evaluator to simulate the behaviour of the cryptographic co-processor and get a prior knowledge about its robustness against SCA. Indeed, given the cryptographic algorithm, the evaluator has some clues about the executed instructions which make him able to simulate the power consumption of these cryptographic instructions. The advantage with these simulations is that the perfect case (*i.e* without presence of noise) can be studied. Actually, if the attack fails on perfect simulated traces, it must fail on real ones. In the literature of SCA, the majority of scientific papers focus on this kind of simulations to assess the efficiency of deployed analysis. However, during our tests, we noticed that the traditional metrics [146] used to compare the efficiency of different Side-channel attacks do not necessarily give the same results for both simulated and real SCA traces. Indeed, the efficiency of SCA is very dependent on the noise variation that we can not simulate with exactitude. Therefore, we believe that such simulations are not sufficient to compare different SCA.

4.3.5 Analysis & Decision Phase

Thanks to this phase, the evaluator is able to make decisions about the robustness of the cryptographic implementation. Indeed, it can be seen as a tool box that aims at giving as much details as possible about the real security dynamics deployed in the product. Today's evaluator has access to a broad range of metrics used to assess the performance of SCA. Basically, these metrics can be sorted in three classes. The first class includes those metrics that aim at measuring the efficiency of one attack in term of number of traces required to recover the secret key, such as the Stability criteria (SC) [143], the First-order success rate (SR) [146] and the Guessing entropy (GE) [146]. The second class of metrics aims at quantifying the leaked information. Eventually, the third class, which is based on the hypothesis testing theory, provides the evaluator with an estimation of the number of SCA traces required to break the cryptographic implementation without actually performing the attack.

4.3.5.1 Cautions on the Use of SCA Metrics

It is important to know that the first class of metrics is basically dependent on the knowledge of the secret key and also on whether the number of acquired traces is limited. In fact, unlike the SC metric, the SR and GE metrics, as described in [146], both require the knowledge of the secret key and a high number of traces to be properly applied. As for the second class, it is mainly based on mutual information computations. From the statistical point of view, the evaluator should keep in mind that the accuracy of these computations is a subject of controversy specially when the number of acquired traces is limited. Concerning the third class of metrics, it is usually required to check the conditions under which the hypotheses tests must be applicable. As a matter of fact, the Gaussian assumption is strictly required before using the *Fisher transformation*, that has been extensively used in the SCA context [85; 86].

In what follows, we propose a new metric, namely KTSR, that points out that some distinguishers might be better at extracting the correct samples and that others are better suited to separate the various key hypotheses. Said differently, we address the question of the *first-order multi-variate* Side-channel analysis. Under this term, we simply revisit basic Side-channel attacks, by considering explicitly their time dimension. Therefore, we propose a framework to evaluate both aspects. Our finding is that combining a time-efficient with an hypothesis-testing-efficient distinguisher can lead to improved analyses.

4.3.5.2 Key-Time Success Rate Metric (KTSR)

Observation attacks exploit some Side-channel emanations in order to validate hypotheses on some secret being used in the circuit. The attacks make use of distinguishers to tell the correct hypothesis from the bad ones. However, in practice, the emanations are vectorial (or multi-valued): for instance, they contain many samples corresponding to different leakage instants. Therefore, the distinguisher is asked to simultaneously find the best leakage sample and the best matching key. The purpose of this study is to point out that some distinguishers might be better at extracting the correct samples and that others are better suited to separate the various key hypotheses. Said differently, we address the question of the *first-order multi-variate* Side-channel analysis. Under this term, we simply revisit the original DPA attacks, by considering explicitly their time dimension. Therefore, we propose a framework to evaluate both aspects. Our finding is that combining a time-efficient distinguisher with an hypothesis-testing-efficient distinguisher can lead to improved analysis. In this study, the compromise between finding the best sample and finding the correct key hypothesis is explored. Typical observation attacks consist in solving a double concomitant maximization problem: in the state-of-the-art, both the most leaking date and the most suitable keys are retrieved simultaneously. Many attacks do not have a priori knowledge of the leakage instants, and thus solve this double optimization problem: $\text{argmax}_{(t,k)}$ *distinguisher(t,k)*. We know that the most commonly used metrics to compute the strength of an attack on a given fixed Side-channel acquisition campaign are the *First-order success rate* and the *Guessing entropy* respectively. However, those metrics grasp only the potential of the distinguisher to detect the good or the best key hypotheses, at the expense of rewarding the discovery of the correct leaking dates. Now, it is clear that if the distinguisher selects poor dates, then without surprise the key sieving will be bad. Nonetheless, most of the theoretical efforts to understand attacks have focused on "univariate" attacks, *i.e.* attacks on the very leaking sample. Therefore, to allow for a scientific analysis of this issue, we consider the trajectory of an attack for various distinguishers, in a 2D graph: *key distinguisher* versus *time distinguisher* either for success rate or guessing entropy. The rationale is that all attacks consistently converge to the same solution $(t_{\text{good}}, k_{\text{good}})$, then this solution can also be reached by optimizing only on the time t for $k = k_{\text{good}}$ or only on the key candidates k for $t = t_{\text{good}}$. Distinguishing keys is easy to quantify, since we know the correct key. However, the definition of the correct leaking sample is less unambiguous. In the figure 4.9, we present a way to define a correct (in red) and an incorrect (in green) sample based on the arbitrary setting of a threshold.

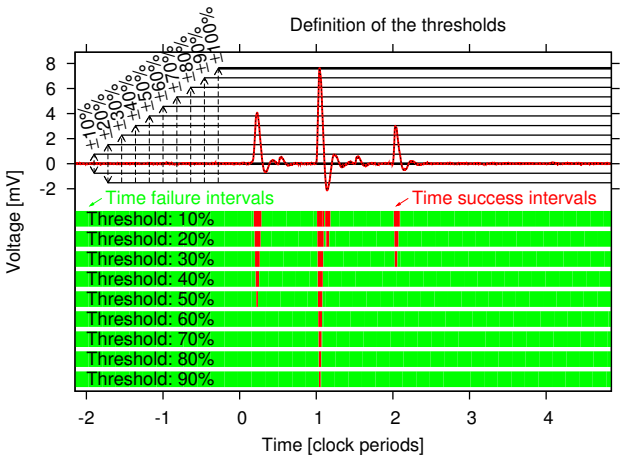


Figure 4.9: Various correct times definitions, depending on a threshold. This degree of freedom can be assimilated to the rank of a key hypothesis amongst all the candidates.

By varying the threshold, we can be more or less purist in the "relevant" *vs* "irrelevant" samples selection. Given this definition, it is possible for sound distinguishers (*i.e.* distinguishers that end up in finding the correct key using a correct sample) to draw a bi-dimensional progression graph (namely KTSR), as illustrated in Fig. 4.10. The detailed procedure to Obtain the KTSR 2D plot is given in the **Appendix**(B.4). Now, a real version (*i.e.* performed on real traces) of the KTSR bi-dimensional graph is shown in Fig. 4.11. This figure involves two attacks DPA and CPA when performed on an unprotected DES implementation.

In the situation depicted in Fig. 4.10, it is obvious that merging the quality of the distinguisher #1 for sample finding and of distinguisher #2 for key finding would be constructive. For instance, on may imagine a combination of both behaviour to develop a more efficient attack. In fact, KTSR reveals details about both leakage time and secret key behaviour. However, it does not give any additional information about the number of traces to retrieve the secret. Eventually, this metric allows to analyse the behaviour of a given distinguisher. Actually, according to Fig. 4.11, DPA is faster in finding the right time sample, while CPA is best in retrieving the correct key hypothesis (*i.e.* secret key).

4.3.6 Methodological Scheme for the Evaluation

This section is a framework in which previously proposed phases are organized in a methodological scheme, in order to streamline the task of the evaluator. As shown in Fig. 4.12, the evaluation process starts by exploring the tested device according to the

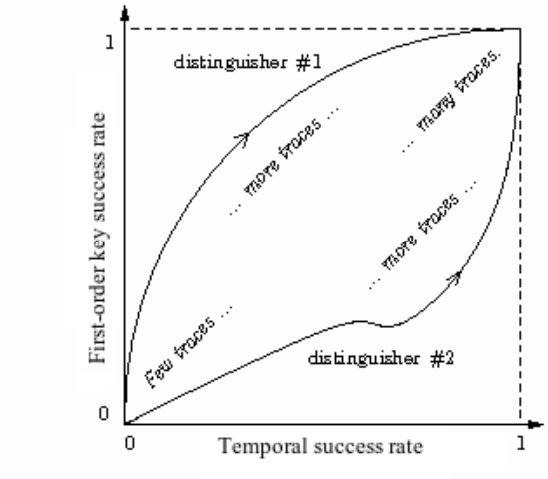


Figure 4.10: Progression of an attack metric (here: the success rate) for two different distinguishers.

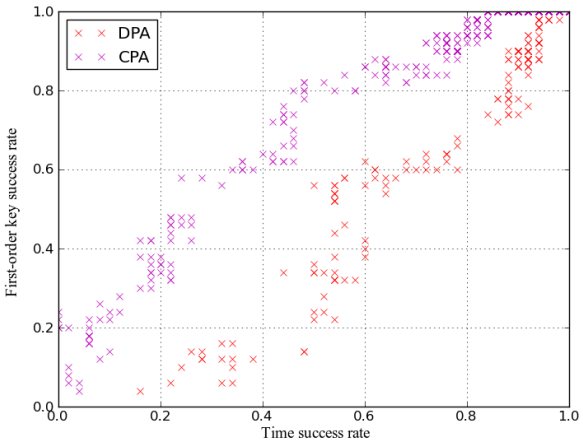


Figure 4.11: KTSR metric for DPA and CPA when performed 100 attacks on DES implementation with threshold of 90%.

documentations or specifications provided by the vendor. A common consideration in the characterization phase is to choose the most appropriate analysis. At this point, according to the provided documentations, the evaluator should be able to know which level of simulation he is allowed to perform. Actually, when transistor and cell netlists are available, the evaluator can precisely determine the different leakages instants occurring during the cryptographic process. When limited knowledge about the device is provided, the evaluator can operate in the same manner as an attacker, who often has some clues about parts of the netlists. Therefore, the evaluator can simulate the power consumption of these parts. According to the evaluation scheme, the simulation phase is then mapped to the analysis and decision phase. Arrows ① and ②, indicate the presence of a mutual relations. In fact, the evaluation is conducted to perform different simulations assessed by the analysis and decision phase until finding all leakages instants; and therefore determining the most appropriate analysis to perform on real Side-channel measurements. The next step in the evaluation process is to proceed to the acquisition phase which is mainly based on the setup of the oscilloscope. This practical phase is mapped to the preprocessing one, which aims at preparing the acquired Side-channel measurements to the analysis and decision phase. According to ⑤, the preprocessing phase receives a feedback from the decision phase. The evaluator is required to improve the analysis process by controlling the preprocessing phase. Eventually, the evaluator establishes an evaluation report in order to verify whether

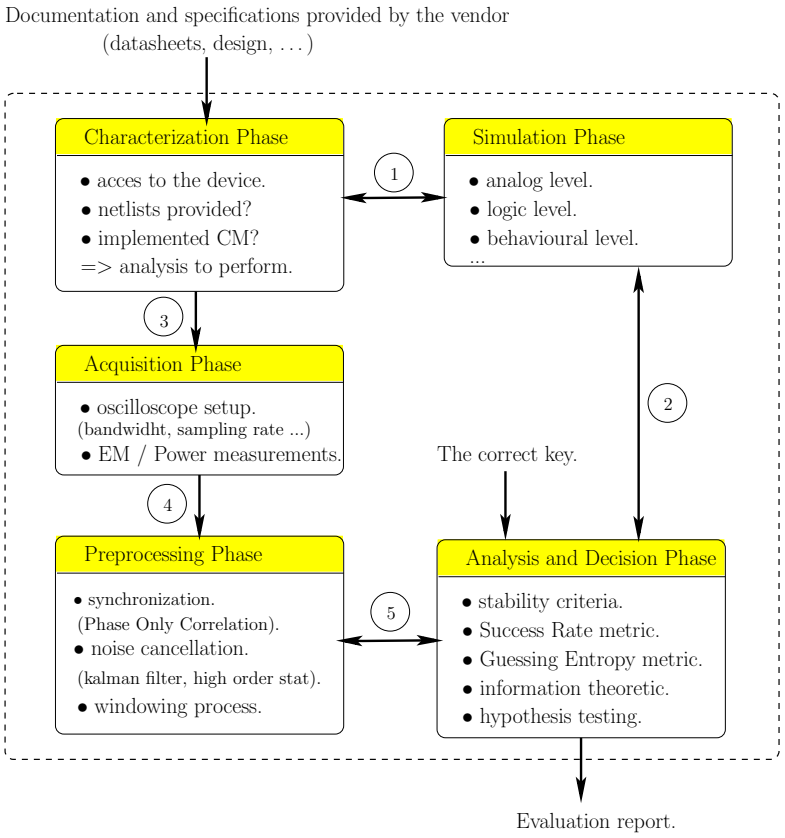


Figure 4.12: Evaluation process scheme.

his analysis meet or not the security requirements claimed by the vendor.

4.4 Conclusion

In this Chapter we have considered how evaluation targets are characterized, how their behaviour may be simulated – in order to hone targets for empirical analysis and then how data can be collected and analysed. The overall goal is the establishment of a methodological basis for this work. Besides, the lack of IT security evaluation references on the common problem of securing embedded systems against SCAs was the starting argument that pushed us to establish a generic and simple framework in which the work of the lab evaluator is organized and structured into five different phases, which are characterization, simulation, acquisition, pre-processing and analysis. We have shown that these phases are in close relationship with each other; and therefore any lack of rigor occurring through one phase could mislead the whole evaluation process.

Chapter 5

Conclusions & Perspectives

Securing modern embedded systems has been the subject of intensive research in the context of engineering systems. Recent threats called Side-Channel Analysis (SCA) have attracted much attention in embedded security areas. These attacks that unintentionally exploit physical leakage, such as the timing information and power consumption or radiated magnetic field, are passive in that the device under attack is not aware of its leaks being recorded. From the security evaluation perspective, if a device, which has been evaluated as secure, could be attacked by such means, it might create a crisis of trust between vendors (manufacturers, embedded system designers, *etc.*) and customers. Therefore, the need of securing and evaluating the robustness of embedded systems against SCAs becomes clear. This thesis investigates new techniques in the analysis of systems for Side-channel attacks. The overall goal is the establishment of a methodological basis to brighten the task of the Side-channel evaluator when assessing the robustness of secure embedded systems against SCAs.

The first part of this thesis (2) has focused on physical cryptanalysis. Several solutions and generic Side-channel attacks have been addressed. From the evaluation point of view, genericity is a crucial criterion that allows the evaluator to reduce the cost of the analysis before the multitude of existing attacks in the litterature. Actually, first we have given an in-depth study about the general performance of Correlation Power Analysis (CPA). We have shown that under the Gaussian case CPA is optimal with regards to attacks that exploit the same leakage model. This study is useful in that it allows the evaluator to assess the efficiency of CPA; and therefore to decide on the choice of an appropriate Side-channel distinguisher for his analysis. Second, we have taken advantage of CPA's performance study to put forward new methodologies, based on the combination of most commonly known SCA distinguishers, in order to accelerate the key recovery, in a **generic** manner. More precisely, the methodologies proposed combine commonly used Side-channel distinguishers, Pearson and Spearman

coefficients, both theoretically (Gini correlation) and empirically. Please note that we have intended to propose **generic** methodologies to accelerate attacks and not attacks in particular. Regarding the empirical approach, we have shown that different distinguishers can be combined using appropriate aggregate functions; and that the choice of aggregate functions depends on the practical behaviour of distinguishers. For this purpose, third we have proposed to give a special attention to such behaviour, by analysing the rank evolution of key hypotheses. The principle based on the distinguishable evolution of the ranking between the secret key and false keys has been observed on different SCAs. In this insight, we have proposed a generic algorithm, namely the Rank Corrector (RC), to correct the decision taken by Side-channel distinguishers when selecting the best key hypothesis. Besides, we assume that the efficiency of RC is mainly dependent on the amount of noise affecting Side-channel traces and the features of used distinguisher (robust/non robust coefficient, univariate/multivariate). In the same context of investigating new Side-channel distinguishers, fourth we have presented a new powerful multivariate distinguisher, called First Principal Components Analysis (FPCA). We have shown the efficiency of FPCA on unprotected as well as protected cryptographic implementations. Besides, the originality of FPCA is to use Principal Component Analysis (PCA) no more as a pre-processing tool, as deployed for Template attacks, but as a genuine distinguisher. In this insight, fifth we have proposed to use Wavelets analysis, that was initially deployed for pre-processing traces, to enhance the quality of distinguishers; and therefore improve Side-channel attacks. As perspectives, other combination of Side-channel analyses can be studied, such as the combination of multiple distinguishers or the investigation of more complex aggregate functions. Besides, in order to improve the basic algorithm of FPCA, one may look for new applications based on other multivariate data analytic tools such as the Linear Discriminant Analysis (LDA) [70], PCA based Spearman correlation [127], Kernel PCA or Independent Component Analysis (ICA), which have been proposed as new alternatives to the basic PCA, in the open litterature of multivariate analysis.

The second part of this thesis (3) has been devoted to the pre-processing of the Side-channel leaked information. The pre-processing of traces acquired is crucial as the main idea behind Side-channel attacks is to best detect, extract and analyse the secret information from digital signals. We have presented new techniques and efficient pre-processing algorithms to get rid off the issues related principally to the noise and de-synchronisation problems. First, we have proposed an algorithmic solution, based on the well known Kalman filtering theory [156]. Second, we have investigate the Expectation-Maximization algorithm (EM) to get a better estimation of the Kalman

filter (KF) parameters. Third, we have proposed the use of the Wavelet transform to best pre-process traces. Indeed, using Wavelet analysis for Side-channel noise filtering have been first proposed by Pelletier [110]. In this thesis, we have developed a new algorithm to improve the existing technique of Pelletier; and proposed other applications of Wavelet transform in pre-processing acquired signals, such as the compression of traces and the detection of cryptographic patterns. We note that, in this thesis, we have investigated in a methodological manner the use of Wavelets transform in different aspects of the Side-channel analysis. Fig. 5.1 shows the involvement of the wavelets based applications in SCA aspects. Fourth, we have given more in depth view

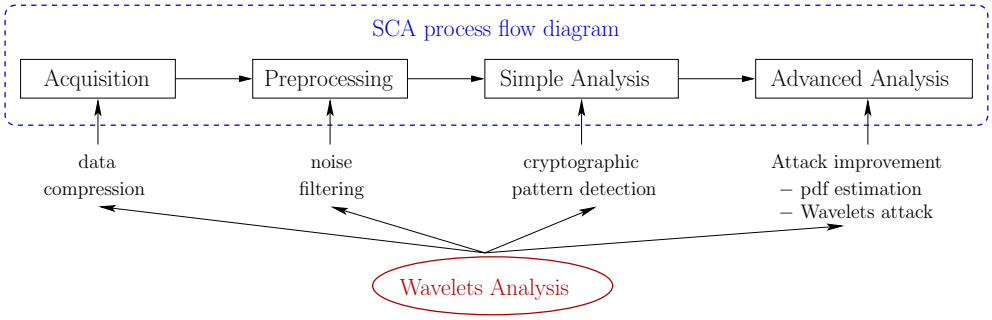


Figure 5.1: Involvement of wavelet analysis in SCA security aspects.

of adapting the basics of Electromagnetic Shielding theory, which particularly aims at decreasing the contribution of external noise sources, to the Side-channel context. Fifth, we have proposed a new re-synchronization algorithm, namely RM, as a pre-processing step in Side-channel analysis. We have highlighted the importance of aligning SCA traces and surveyed existing techniques to get round the problem of misalignment. As perspectives, one may investigate more techniques that could be more specific to the Side-channel context. For instance, our experiments using Wavelets analysis, have been essentially based on existing wavelet mother functions. Nonetheless, in the litterature of Wavelet analysis, it has been shown that it is totally possible to design new and more specific wavelet mother functions.

In the last part of this thesis (4), we have considered how evaluation targets are characterized, how their behaviour may be simulated; in order to hone targets for empirical analysis and then how data can be collected and analysed. The overall goal has been the establishment of a methodological basis for this work. We have established a generic and straightforward framework in which the work of the evaluator is organized and structured into five different phases, which are characterization, simulation,

acquisition, pre-processing and analysis. We have shown that these phases are in close relationship with each other; and therefore any lack of rigor occurring through one phase could mislead the whole evaluation process. Besides, we have also highlighted common pitfalls made by evaluators and solutions to avoid them. As perspectives, our methodology can be tested on real life embedded systems integrating cryptographic implementation. Moreover, experienced evaluators may look for new techniques for embedded systems reverse-engineering; and therefore improve the evaluation methodology proposed.

Appendix A

Publications & Activities

Title	Authors	Conference
Towards New Flavors for Combined Side-Channel Attacks	Youssef Souissi, Shivam Bhasin, Maxime Nassar, Sylvain Guilley et Jean-Luc Danger	**CT-RSA 2012**
On the Optimality of Correlation Power Analysis	Youssef Souissi, Sami Mekki, Nicolas Debande, Ali Maalaoui, Sylvain Guilley et Jean-Luc Danger	**WISTP 2012** Siwth Workshop In Information Security Theory And Practice.
RSM: a Small and Fast Countermeasure for AES, Secure against 1st and 2nd-order Zero-Offset SCAs	Maxime Nassar, Youssef Souissi, Sylvain Guilley et Jean-Luc Danger	**DATE 2012** Design, Automation, and Test in Europe.

A Multiresolution Time-Frequency Analysis Based Side Channel Attacks	Youssef Souissi, Moulay aziz Elaabid, Nicolas Debande, Sylvain Guilley et Jean-Luc Danger	**Poster Session of WIFS 2011** IEEE International Workshop on Information Forensics and Security.
"Re-Synchronization by Moments": An Efficient Solution to Align Side-Channel Traces	Nicolas Debande, Youssef Souissi, Sylvain Guilley, Jean-Luc Danger, Maxime Nassar et Thanh-Ha Le	**WIFS 2011** IEEE International Workshop on Information Forensics and Security.
Efficient Dual-Rail Implementations in FPGA using Block RAMs	Shivam Bhasin, Sylvain Guilley, Tarik Graba, Youssef Souissi et Jean-Luc Danger	**ReConFig 2011** International Conference on ReConFigurable Computing and FPGAs.
Common Framework to Evaluate Modern Embedded Systems against Side-Channel Attacks	Youssef Souissi, Shivam Bhasin, Sylvain Guilley et Jean-Luc Danger	**HST 2011** IEEE International Conference on Technologies for Homeland Security.

Embedded Systems Security: An Evaluation Methodology Against Side Channel Attacks	Youssef Souissi, Maxime Nassar, Shivam Bhasin, Sylvain Guilley et Jean-Luc Danger	**DASIP 2011** Conference on Design and Architectures for Signal and Image Processing
"Rank Correction": A New Side-Channel Approach For Secret Key Recovery	Maxime Nassar, Youssef Souissi, Sylvain Guilley et Jean-Luc Danger	**Info Sec HiComNet 2011** International Conference.
Novel Applications of Wavelet Transforms based Side-Channel Analysis	Youssef Souissi, Moulay aziz Elaabid, Nicolas Debande, Sylvain Guilley et Jean-Luc Danger	**NIAT 2011** Non-Invasive Attack Testing Workshop.
"Time-Success rate" as a new security metric for Side-Channel Analysis	Youssef Souissi, Sylvain Guilley, Maxime Nassar, Shivam Bhasin et Jean-Luc Danger	**Poster Session of CHES 2011**
DPL Implementations in FPGA using Embedded BRAM	Shivam Bhasin, Sylvain Guilley, Youssef Souissi, Tarik Graba et Jean-Luc Danger	**TrustED 2011** First International Workshop on Trustworthy Embedded Devices.
Combination of Measurements to accelerate side channel attacks	Youssef Souissi, Shivam Bhasin, Maxime Nassar, Sylvain Guilley et Jean-Luc Danger	**Poster Session of CHES 2011**

Efficient FPGA Implementations of Dual-Rail Countermeasures using Stochastic Models	Shivam Bhasin, Sylvain Guilley, Youssef Souissi et Jean-Luc Danger	**NIAT 2011** Non-Invasive Attack Testing Workshop.
Vade Mecum on Side-Channels Attacks and Countermeasures for the designer and the Evaluator	Sylvain Guilley, Guillaume Duc, Ph. Hoogvorst, Moulay aziz Elaabid, Shivam Bhasin, Youssef Souissi, Nicolas Debande, Laurent Sauvage et Jean-Luc Danger	**DTIS 2011** Design & Technology of Integrated Systems in nanoscale era.
Quantifying the Quality of Side Channel Acquisitions	Sylvain Guilley, Youssef Souissi, Houssem Maghrebi et Jean-Luc Danger	**COSADE 2011** 2nd International Workshop on Constructive Side-Channel Analysis and Secure Design.
Side Channel Analysis enhancement: A proposition for measurements resynchronisation	Nicolas Debande, Youssef Souissi, Maxime Nassar, Sylvain Guilley, Thanh-Ha Le et Jean-Luc Danger	**CryptArchi Workshop 2011**
First Principal Components Analysis: A New Side Channel Distinguisher	Youssef Souissi, Maxime Nassar, Sylvain Guilley, Jean-Luc Danger et Flament Florent	**ICISC 2010** LNCS 14th Annual International Conference on Information Security and Cryptology.
The "Rank Correction" Technique to Improve Side-Channel Attacks	Youssef Souissi, Maxime Nassar, Sylvain Guilley et Jean-Luc Danger	**CryptArchi Workshop 2010**

Techniques for electromagnetic attacks enhancement	Youssef Souissi, SamiMekki, Sylvain Guilley et Jean-Luc Danger	**DTIS 2010**
Improvement of Power Analysis Attacks Using Kalman Filter	Youssef Souissi, Sylvain Guilley, Jean-Luc Danger, Guillaume Duc et Sami Mekki	**ICASSP 2010** IEEE International Conference on Acoustics, Speech, and Signal Processing.

Appendix B

Appendix

B.1 Adding noise decreases the quality of ρ

Proof: Let χ_1 and χ_2 be two independent random variables with ϵ_1 and ϵ_2 as their associated noise. Let $X = \chi_1 + \epsilon_1$, and $Y = \chi_2 + \epsilon_2$. Because of independence, on can write:

$$\begin{aligned} Cov_{(X,Y)} &= Cov_{(\chi_1+\epsilon_1,\chi_2+\epsilon_2)} \ , \\ Cov_{(X,Y)} &= Cov_{(\chi_1,\chi_2)} + Cov_{(\chi_1,\epsilon_2)} + Cov_{(\chi_2,\epsilon_1)} + Cov_{(\epsilon_1,\epsilon_2)} \ , \\ Cov_{(X,Y)} &= Cov_{(\chi_1,\chi_2)} \ . \end{aligned} \tag{B.1}$$

Hence:

$$\begin{aligned} \rho_{(\chi_1+\epsilon_1,\chi_2+\epsilon_2)} &= \frac{Cov(\chi_1,\chi_2)}{\sqrt{(Var(\chi_1+\epsilon_1)Var(\chi_2+\epsilon_2))}} \ , \\ \rho_{(\chi_1+\epsilon_1,\chi_2+\epsilon_2)} &< \frac{Cov(\chi_1,\chi_2)}{\sqrt{(Var(\chi_1)Var(\chi_2))}=\rho_{(\chi_1,\chi_2)}} \ . \end{aligned} \tag{B.2}$$

B.2 Expectation-Maximization components calculation

B.2.1 Component α

We assume $x_0 \sim \mathcal{N}(\mu, \Sigma)$, so the probability density funtion of x_0 is given by:

$$p(x_0) = \frac{1}{\sqrt{2\pi\Sigma}} e^{\left(-\frac{(x_0-\mu)^2}{2\Sigma}\right)} \ . \tag{B.3}$$

We note that:

$$x_0 = \widehat{x}_{0|L} + (x_0 - \widehat{x}_{0|L}) \tag{B.4}$$

Applying Eqn. B.4 into the alpha part of Eqn. 3.36, we get:

$$\mathbb{E}_{X|Y}\left[\log p(x_0)|\theta^{(i)}\right] = \mathbb{E}_{X|Y}\left[-\frac{\log(2\pi)}{2} - \frac{\Sigma}{2} - \frac{(\widehat{x}_{0|L} - \mu + (x_0 - \widehat{x}_{0|L}))^2}{2\Sigma}|\theta^{(i)}\right], \tag{B.5}$$

$$= -\frac{\log(2\pi)}{2} - \frac{\Sigma}{2} - \frac{1}{2\Sigma}\mathbb{E}_{X|Y}\left[(x_0 - \widehat{x}_{0|L})^2 + (\widehat{x}_{0|L} - \mu)^2 + 2(\widehat{x}_{0|L} - \mu)(x_0 - \widehat{x}_{0|L})|\theta^{(i)}\right],$$

$$= -\frac{\log(2\pi)}{2} - \frac{\Sigma}{2} - \frac{1}{2\Sigma}\left(\mathbb{E}_{X|Y}\left[(x_0 - \widehat{x}_{0|L})^2|\theta^{(i)}\right] + (\widehat{x}_{0|L} - \mu)^2\right), \tag{B.6}$$

$$= -\frac{\log(2\pi)}{2} - \frac{\Sigma}{2} - \frac{1}{2\Sigma}\left(P_{0|L} + (\widehat{x}_{0|L} - \mu)^2\right) \ . \tag{B.7}$$

Eqn. B.6 comes from:

$$\mathbb{E}_{X|Y}\left[\underbrace{(\widehat{x}_{0|L} - \mu)}_{\text{constante}} \ (x_0 - \widehat{x}_{0|L})|\theta^{(i)}\right] = (\widehat{x}_{0|L} - \mu) \ \mathbb{E}_{X|Y}\left[x_0 - \widehat{x}_{0|L}|\theta^{(i)}\right] \tag{B.8}$$

$$= (\widehat{x}_{0|L} - \mu) \ (\underbrace{\mathbb{E}_{X|Y}\left[x_0|\theta^{(i)}\right]}_{\widehat{x}_{0|L}} - \widehat{x}_{0|L}) \tag{B.9}$$

$$= 0 \ . \tag{B.10}$$

Finally, the component α has the following expression:

$$\mathbb{E}_{X|Y}\left[\log p(x_0)|\theta^{(i)}\right] = C^{te} - \frac{\Sigma}{2} - \frac{1}{2\Sigma}\left(P_{0|L} + (\widehat{x}_{0|L} - \mu)^2\right), \tag{B.11}$$

where C^{te} is a constante.

B.2.2 Component β

Given $x_{\ell-1}$, x_ℓ follows a Gaussian distribution $\mathcal{N}(Ax_{\ell-1}, \sigma_w^2)$. Thus, its probability density function is given by:

$$p(x_\ell|x_{\ell-1}) = \frac{1}{\sqrt{2\pi\sigma_w^2}} e^{-\frac{(x_\ell - Ax_{\ell-1})^2}{2\sigma_w^2}} \ . \tag{B.12}$$

Additionally, observe that:

$$x_\ell - Ax_{\ell-1} = (x_\ell - \widehat{x}_{\ell|L}) - A(x_{\ell-1} - \widehat{x}_{\ell-1|L}) + (\widehat{x}_{\ell|L} - A\widehat{x}_{\ell-1|L}) \ . \tag{B.13}$$

Combining the above, the new expression of the component β is:

$$\begin{aligned}\sum_{\ell=1}^{L} \mathbb{E}_{X|Y}\left[\log p(x_\ell|x_{\ell-1})|\theta^{(i)}\right] &= -\frac{L}{2}\log(2\pi) - \frac{L}{2}\log\sigma_w^2 \\ &\quad - \sum_{\ell=1}^{L} \mathbb{E}_{X|Y}\left[\frac{((x_\ell - \widehat{x}_{\ell|L}) - A(x_{\ell-1} - \widehat{x}_{\ell-1|L}) + (\widehat{x}_{\ell|L} - A\widehat{x}_{\ell-1|L}))^2}{2\sigma_w^2}|\theta^{(i)}\right]\end{aligned} \tag{B.14}$$

$$\begin{aligned}&= -\frac{L}{2}\log(2\pi) - \frac{L}{2}\log\sigma_w^2 \\ &\quad - \frac{1}{2\sigma_w^2}\sum_{\ell=1}^{L} \mathbb{E}_{X|Y}\Big[(x_\ell - \widehat{x}_{\ell|L})^2 + A^2(x_{\ell-1} - \widehat{x}_{\ell-1|L})^2 + (\widehat{x}_{\ell|L} - A\widehat{x}_{\ell-1|L})^2 \\ &\quad - 2A(x_\ell - \widehat{x}_{\ell|L})(x_{\ell-1} - \widehat{x}_{\ell-1|L}) - 2A(x_{\ell-1} - \widehat{x}_{\ell-1|L})(\widehat{x}_{\ell|L} - A\widehat{x}_{\ell-1|L}) \\ &\quad + 2(x_\ell - \widehat{x}_{\ell|L})(\widehat{x}_{\ell|L} - A\widehat{x}_{\ell-1|L})|\theta^{(i)}\Big]\end{aligned} \tag{B.15}$$

$$= -\frac{L}{2}\log(2\pi) - \frac{L}{2}\log\sigma_w^2 - \frac{1}{2\sigma_w^2}\sum_{\ell=1}^{L}\left[P_{\ell|L} + A^2P_{\ell-1|L} + (\widehat{x}_{\ell|L} - A\widehat{x}_{\ell-1|L})^2 - 2AP_{\ell,\ell-1|L}\right] . \tag{B.16}$$

Eqn. B.16 results from the following equalities:

$$\mathbb{E}_{X|Y}\left[(x_\ell - \widehat{x}_{\ell|L})(x_{\ell-1} - \widehat{x}_{\ell-1|L})|\theta^{(i)}\right] = P_{\ell,\ell-1|L} \ , \tag{B.17}$$

$$\mathbb{E}_{X|Y}\left[(x_{\ell-1} - \widehat{x}_{\ell-1|L})(\widehat{x}_{\ell|L} - A\widehat{x}_{\ell-1|L})|\theta^{(i)}\right] = 0 \ , \tag{B.18}$$

$$\mathbb{E}_{X|Y}\left[(x_\ell - \widehat{x}_{\ell|L})(\widehat{x}_{\ell|L} - A\widehat{x}_{\ell-1|L})|\theta^{(i)}\right] = 0 \ . \tag{B.19}$$

Hence, the β component is given by a new explicit form:

$$\sum_{\ell=1}^{L}\mathbb{E}_{X|Y}\left[\log p(x_\ell|x_{\ell-1})|\theta^{(i)}\right] = C^{te} + -\frac{L}{2}\log\sigma_w^2 - \frac{1}{2\sigma_w^2}\sum_{\ell=1}^{L}\left[P_{\ell|L} + A^2P_{\ell-1|L} + (\widehat{x}_{\ell|L} - A\widehat{x}_{\ell-1|L})^2 - 2AP_{\ell,\ell-1|L}\right] , \tag{B.20}$$

B.2.3 Component γ

We assume that y_ℓ follows a Gaussian distribution with mean $H\,x_\ell$ and variance σ_v^2, *i.e.* $y_\ell \sim \mathcal{N}(Hx_\ell, \sigma_v^2)$. The probability density function of y_ℓ is given by:

$$p(y_\ell|x_\ell) = \frac{1}{\sqrt{2\pi\sigma_v^2}}e^{-\frac{(y_\ell - Hx_\ell)^2}{2\sigma_v^2}} \ . \tag{B.21}$$

Using the following equality, we get:

$$y_\ell - Hx_\ell = y_\ell - H\widehat{x}_{\ell|L} - H(x_\ell - \widehat{x}_{\ell|L}) \ . \tag{B.22}$$

We may then rewrite the component γ as:

$$= -\tfrac{L}{2}\log(2\pi) - \tfrac{L+1}{2}\log\sigma_v^2 - \textstyle\sum_{\ell=0}^{L}\mathbb{E}_{X|Y}\left[\frac{((y_\ell - \widehat{x}_{\ell|L}) - H(x_\ell - \widehat{x}_{\ell|L}))^2}{2\sigma_v^2}|\theta^{(i)}\right]$$

$$= C^{te} - \tfrac{L+1}{2}\log\sigma_v^2 - \tfrac{1}{2\sigma_v^2}\textstyle\sum_{\ell=0}^{L}\mathbb{E}_{X|Y}\left[(y_\ell - \widehat{x}_{\ell|L})^2 + H^2(x_\ell - \widehat{x}_{\ell|L})^2 + 2H(x_\ell - \widehat{x}_{\ell|L})(y_\ell - \widehat{x}_{\ell|L})|\theta^{(i)}\right]$$

where C^{te} refers to a constante. We have to note that:

$$\mathbb{E}_{X|Y}\left[(x_\ell - \widehat{x}_{\ell|L})(y_\ell - \widehat{x}_{\ell|L})|\theta^{(i)}\right] = (y_\ell - \widehat{x}_{\ell|L})\ \mathbb{E}_{X|Y}\left[(x_\ell - \widehat{x}_{\ell|L})|\theta^{(i)}\right] = 0\ . \quad \text{(B.25)}$$

thus the component γ is reduced to a simple form as:

$$\sum_{\ell=0}^{L}\mathbb{E}_{X|Y}\left[\log p(y_\ell|x_\ell)|\theta^{(i)}\right] = C^{te} - \frac{L+1}{2}\log\sigma_v^2 - \frac{1}{2\sigma_v^2}\sum_{\ell=0}^{L}\left[(y_\ell - \widehat{x}_{\ell|L})^2 + H^2 P_{\ell|L}\right] \quad \text{(B.26)}$$

B.3 Binomial Formulas

The development of a polynomial yields:

$$\forall x \in \mathbb{R}, (1+x)^n = \sum_{i=0}^{n}\binom{n}{i}x^i\ . \quad \text{(B.27)}$$

Hence, in $x = 1$, we get:

$$2^n = \sum_{i=0}^{n}\binom{n}{i}\ . \quad \text{(B.28)}$$

Now, from Eqn. (B.27):

$$\begin{aligned}\forall x \in \mathbb{R}, \frac{\partial}{\partial x}(1+x)^n &= n(1+x)^{n-1} \\ &= \sum_{i=0}^{n}\frac{\partial}{\partial x}\binom{n}{i}x^i = \sum_{i=1}^{n} i\binom{n}{i}x^{i-1} \\ &= \sum_{i=0}^{n} i\binom{n}{i}x^{i-1}\ . \end{aligned} \quad \text{(B.29)}$$

Hence, in $x = 1$, we get:

$$\sum_{i=0}^{n} i\binom{n}{i} = n2^{n-1}\ . \quad \text{(B.30)}$$

From Eqn. (B.29):

$$\begin{aligned}\forall x \in \mathbb{R}, \frac{\partial}{\partial x} n(1+x)^{n-1} &= n(n-1)(1+x)^{n-2} \\ &= \sum_{i=0}^{n} \frac{\partial}{\partial x} i \binom{n}{i} x^{i-1} = \sum_{i=1}^{n} i(i-1)\binom{n}{i} x^{i-2} \\ &= \sum_{i=0}^{n} i(i-1)\binom{n}{i} x^{i-2} \ .\end{aligned}$$

Hence, in $x = 1$, we get:

$$\sum_{i=0}^{n} i(i-1)\binom{n}{i} = n(n-1)2^{n-2} \ . \tag{B.31}$$

Said differently, by adding Eqn. (B.28) and Eqn. (B.30), we get:

$$\sum_{i=0}^{n} i^2 \binom{n}{i} = n(n+1)2^{n-2} \ . \tag{B.32}$$

B.4 Procedure to Obtain the KTSR 2D plot (such as the one of Fig. 4.10)

To obtain concretely the plots like the one depicted in Fig. 4.10), three steps are necessary:

1. Identify the correct and the bad samples for various thresholds;
2. Compute an attack statistics matrix for k and t;
3. Draw the graphs.

In the three next subsections B.4.1, B.4.2 and B.4.3, we detail each of these steps. For the sake of illustration, we consider that:

- we use 10 thresholds to define the good and bad time samples;
- we assume we seek 1 key amongst 256;
- we launch 100 campaigns with different keys to compute the success rates or the guessing entropies;

- we have 10,000 traces per campaign;
- each trace is made up of 20,000 samples.

B.4.1 Identification of the correct and bad samples

In this phase, we conduct an attack on one campaign amongst the 100 available, and get a curve (for the correct key only), such as the one shown in Fig. 4.9. Let's call this curve $\mathsf{DPA}(t)$ (but it works also if the distinguisher is CPA, FPCA, *etc.*). Then, we define the "good" samples T_{th}^{good} for the threshold $th \in [0\%, 100\%] = [0, 1]$ by:

$$T_{th}^{\text{good}} = \left\{ t \in [0, 20,000[, \text{ such that } |\mathsf{DPA}(t)| \geq th \times \max_{\tau} |\mathsf{DPA}(\tau)| \right\} . \tag{B.33}$$

Typically, we can compute $T_{10\%}^{\text{good}}, T_{20\%}^{\text{good}}, \cdots, T_{90\%}^{\text{good}}$. Note that those sets satisfy: $T_{10\%}^{\text{good}} \supset T_{20\%}^{\text{good}} \supset \cdots \supset T_{90\%}^{\text{good}}$.

B.4.2 Compute an attack statistics matrix for k and t

For each campaign, the attack is evaluated each time a new trace is added to the estimation. The attack's result is summarized in a couple:

- the keys ranking: for each key hypothesis, the maximum of the curve over the samples is chosen;
- the samples ranking for the correct key only: the samples are ordered according to the distinguisher's value.

Then, these results are saved in a matrix.

B.4.3 Draw the graphs

After the 100 campaigns are processed the same way, the results can be computed.

For instance, the 1st-order success rate for the key is computed as the probability that the key ranked 1 actually corresponds to the correct key k°. The oth-order success rate would be the counting of the number of times the key ranking of k° is either 1, 2, ..., or o. The guessing entropy for the key is the average ranking of k°.

Regarding the time-oriented metrics, they are defined for a given threshold th. The 1st-order success rate in time is the average number of times the best sample

corresponding to the correct key k° (first column) belongs to T_{th}^{good}. The oth-order success rate for the time is the average number of times at least one of the best oth samples is in T_{th}^{good}. The guessing entropy in time is the average ranking of the first time index that belongs to T_{th}^{good}.

Appendix C

Glossary

C.1 Acronyms

AES:	Advanced Encryption Standard
ASIC:	Application Specific Integrated Circuit
AOC:	Amplitude Only Correlation
BCDL:	Balanced Cell based Dual-rail Logic
CBC:	Cipher Block Chaining
CC:	Common Criteria
CEV:	Cumulative Explained Variance
CFB:	Cipher Feedback
CMOS:	Complementary Metal Oxide Semiconductor
CPA:	Correlation Power Analysis
CPK:	Current Predicted Key
CS:	Chosen Statistic
CWT:	Continuous Wavelet Transform
DCA:	Differential Cluster Analysis
DES:	Data Encryption Standard
DFA:	Differential Fault Attack
DFT:	Discrete Fourier Transform
DoM:	Difference of Means
DPA:	Differential Power Analysis
DPL:	Dual-rail with Precharge Logic
DSA:	Digital Signature Algorithm
DWT:	Discrete Wavelet Transform
EAL:	Evaluation Assurance Level

ECB:	Electronic Code Book
ECC:	Elliptic Curve Cryptography
Em:	Electromagnetic
EM:	Expectation Maximisation
EMA:	Electro Magnetic Analysis
EV:	Explained Variance
FFT:	Fast Fourier Transform
FIPS:	Federal Information Processing Standard
FK:	False Key
FP:	Final Permutation
FPCA:	First Principal Components Analysis
FPGA:	Field Programmable Gate Array
GE:	Guessing Entropy
HD:	Hamming Distance
HOS:	High Order Statistics
HW:	Hamming Weight
ICA:	Independent Component Analysis
IP:	Initial Permutation
ISO:	International Organization for Standardization
ITSEC:	Information Technology Security Evaluation Criteria
KF:	Kalman Filter
KS:	Kalman Smoother
KTSR:	Key Time Success Rate
LDA:	Linear Discriminant Analysis
LMMSE:	Linear Minimum Mean Square Error
MDPL:	Masked Dual-rail Precharge Logic
MI:	Mutual Information
MIA:	Mutual Information Analysis
ML:	Maximum Likelihood
MLE:	Maximum Likelihood Estimator
MMSE:	Minimum Mean Square Error
MSE:	Mean Square Error
MTD:	Minimum Traces to Disclose the key
NIST:	National Institute of Standard and Technology
NSA:	National Security Agency
RSA:	Rivest Shamir Adleman

OFB:	Output Feedback
PC:	Possibility of Combination
PCA:	Principal Component Analysis
PK:	Predicted Key
PKC:	Public Key Cryptography
PKI:	Public Key Infrastructure
POC:	Phase Only Correlation
PP:	Protection Profile
RAM:	Radiation Absorbent Material
RC:	Rank Corrector
RM:	Resynchronisation by Moments
S:	Stability
SC:	Stability Criteria
SCA:	Side Channel Attack
SCAN:	Spectral Coherence ANalysis
SFR:	Security Functional Requirements
SHA:	Secure Hash Algorithm
SK:	Secret Key
SKC:	Symmetric Key Cryptography
SNR:	Signal-to-Noise Ratio
SPA:	Simple Power Analysis
ST:	Security Target
STH:	Stability Threshold
SR:	Success Rate
STFT:	Short Fourier Transform
TCSEC:	Trusted Computer System Evaluation Criteria
TOE:	Target Of Evaluation
UART:	Universal Asynchronous Receiver Transmitter
VPA:	Variance Power Analysis
WDDL:	Wave Dynamic Differential Logic

C.2 Notations

P_{law}:	Probability law
E_m:	Electromagnetic
pdf:	Probability Density Function
sk:	Secret Key
ϵ:	Error's symbol (*i.g.* estimation error, *etc.*)
σ:	Standard deviation (σ^2_{noise} is the noise variance)
μ:	Mean operator
$\mathbb{E}$:	Expectation operator
λ_a:	Eigenvalue of Principal component a
ω:	Angular frequency
T_n:	Time period
e^x:	Exponential operator of x
ψ:	Mother wavelet function
τ:	Time shift parameter
s:	Scale shift parameter (used for wavelets)
ρ:	Pearson correlation coefficient
r:	Spearman rank correlation coefficient
ξ:	Gini correlation coefficient
$\Re_B$:	Bayes Risk
VAR:	Variance operator
Cov:	Covariance operator
Δ:	SCA distinguisher
ϕ:	Approximation function
β:	Binomial distribution
F_X:	Cumulative distribution of X
Ψ:	Aggregate function
$\hat{x}$:	An estimation of x
log:	Logarithm operator
λ_{Donoho}:	Donoho's threshold
ϕ_m:	Magnetic flux
λ_m:	Electromagnetic wavelength
G_t:	Gini dispersion index
G:	Gini metric for SCA signals resynchronization
$\Re_{X,Y}$:	True relationship between X and Y

References

[1] *http* : *//www.niap* − *ccevs.org/cc*$_d$*ocs/*.

[2] M. A. E. Aabid, S. Guilley, and P. Hoogvorst, *Template Attacks with a Power Model*. Cryptology ePrint Archive, Report 2007/443, December 2007. http://eprint.iacr.org/2007/443/.

[3] M. A. E. Aabid, O. Meynard, S. Guilley, and J.-L. Danger, *Combined Side-Channel Attacks*, in WISA, vol. 6513 of LNCS, Springer, August 24-26 2010, pp. 175–190. Jeju Island, Korea. DOI: 10.1007/978-3-642-17955-6_13.

[4] Advanced Encryption Standard (AES) home webpage, *http://csrc.nist.gov/encryption/aes, 2001*.

[5] Agilent Technologies: http://www.agilent.com/.

[6] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, *The EM Side-Channel(s)*, in CHES '02: Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems, London, UK, 2003, Springer-Verlag, pp. 29–45.

[7] D. Agrawal, J. R. Rao, and P. Rohatgi, *Multi-channel Attacks*, in CHES, vol. 2779 of LNCS, Springer, September 8-10 2003, pp. 2–16. Cologne, Germany.

[8] A. Amarilli, S. Müller, D. Naccache, D. Page, P. Rauzy, and M. Tunstall, *Can Code Polymorphism Limit Information Leakage?*, in WISTP, 2011, pp. 1–21.

[9] J. A. Ambrose, R. G. Ragel, and S. Parameswaran, *A smart random code injection to mask power analysis based side channel attacks*, in CODES+ISSS

'07: Proceedings of the 5th IEEE/ACM international conference on Hardware/software codesign and system synthesis, New York, NY, USA, 2007, ACM, pp. 51–56.

[10] M. Aminghafari, N. Cheze, and J.-M. Poggi, *Multivariate denoising using wavelets and principal component analysis*, Comput. Stat. Data Anal., 50 (2006), pp. 2381–2398.

[11] S. Anderson, *Statistical methods for comparative studies: techniques for bias reduction*, Wiley series in probability and mathematical statistics: Applied probability and statistics, Wiley, 1980.

[12] ANTENNESSA (http://www.satimo.com/), *EME SPY 120: Personal Exposure Meter*, tech. rep.

[13] D. Arakaki, *CalPoly Electrical Engeneering (Antenna Anechoic Chamber); http://www.ee.calpoly.edu/projects/anechoic_chamber/*.

[14] C. Archambeau, É. Peeters, F.-X. Standaert, and J.-J. Quisquater, *Template Attacks in Principal Subspaces*, in CHES, vol. 4249 of LNCS, Springer, October 10-13 2006, pp. 1–14. Yokohama, Japan.

[15] B. Arnold, E. Castillo, and J. Sarabia, *Conditional specification of statistical models*, Springer series in statistics, Springer, 1999.

[16] L. Avendano, *Improvement of an extended kalman filter power line interference suppressor for ecg signals.*

[17] B. R. Bakshi, *Multiscale pca with application to multivariate statistical process monitoring*, AIChE Journal, 44 (1998), pp. 1596–1610.

[18] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*, A Wiley-Interscience publication, Wiley, 2001.

[19] L. Batina, B. Gierlichs, and K. Lemke-Rust, *Comparative Evaluation of Rank Correlation Based DPA on an AES Prototype Chip*, in ISC, vol. 5222 of Lecture Notes in Computer Science, Springer, September 15-18 2008, pp. 341–354. Taipei, Taiwan.

[20] R. Bevan and E. Knudsen, *Ways to Enhance Differential Power Analysis*, in ICISC, vol. 2587 of Lecture Notes in Computer Science, Springer, November 28-29 2002, pp. 327–342. Seoul, Korea.

[21] S. Bhasin, J.-L. Danger, F. Flament, T. Graba, S. Guilley, Y. Mathieu, M. Nassar, L. Sauvage, and N. Selmane, *Combined SCA and DFA Countermeasures Integrable in a FPGA Design Flow*, in ReConFig, IEEE Computer Society, December 9–11 2009, pp. 213–218. Cancún, Quintana Roo, México, DOI: 10.1109/ReConFig.2009.50, http://hal.archives-ouvertes.fr/hal-00411843/en/.

[22] A. Bos, *Parameter estimation for scientists and engineers*, Wiley-Interscience, 2007.

[23] P. Brasseur, *Ocean Weather Forecasting. An Integrated View of Oceanography*, Springer, 2006.

[24] E. Brier, C. Clavier, and F. Olivier, *Correlation Power Analysis with a Leakage Model*, in CHES, vol. 3156 of LNCS, Springer, August 11–13 2004, pp. 16–29. Cambridge, MA, USA.

[25] ——, *Correlation power analysis with a leakage model*, in CHES, vol. 3156 of LNCS, Springer, August 11–13 2004, pp. 16–29. Cambridge, MA, USA.

[26] E. Brookner, *Tracking and Kalman Filtering Made Easy*, Wiley-Interscience, April 2008.

[27] S. Broughton and K. Bryan, *Discrete Fourier Analysis and Wavelets: Applications to Signal and Image Processing*, John Wiley & Sons, 2011.

[28] J. Candy, *Bayesian Signal Processing: Classical, Modern and Particle Filtering Methods*, Adaptive and Learning Systems for Signal Processing, Communications and Control Series, John Wiley & Sons, 2011.

[29] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, *Towards Sound Approaches to Counteract Power-Analysis Attacks*, in CRYPTO, vol. 1666 of LNCS, Springer, August 15-19 1999. Santa Barbara, CA, USA. ISBN: 3-540-66347-9.

[30] S. Chari, J. R. Rao, and P. Rohatgi, *Template Attacks*, in CHES, vol. 2523 of LNCS, Springer, August 2002, pp. 13–28. San Francisco Bay (Redwood City), USA.

[31] A. C.Harvey, *Forecasting,structural time series models and the Kalman filter*, Cambridge University Press, 2008.

[32] C. Clavier, J.-S. Coron, and N. Dabbous, *Differential Power Analysis in the Presence of Hardware Countermeasures*, in CHES, LNCS, London, UK, August 2000, Springer-Verlag, pp. 252–263.

[33] C. C. consortium, *Application of attack potential to smartcards v2-5*, April 2008. http://www.commoncriteriaportal.org/files/supdocs/CCDB-2008-04-001.pdf.

[34] J.-S. Coron and I. Kizhvatov, *Analysis and Improvement of the Random Delay Countermeasure of CHES 2009*, in CHES, vol. 6225 of Lecture Notes in Computer Science, Springer, August 17-20 2010, pp. 95–109. Santa Barbara, CA, USA.

[35] J.-S. Coron, P. C. Kocher, and D. Naccache, *Statistics and Secret Leakage*, in Financial Cryptography, vol. 1962 of Lecture Notes in Computer Science, Springer, February 20-24 2000, pp. 157–173. Anguilla, British West Indies.

[36] P. Dagnelie, *Statistique thorique et applique. Tome 2, Infrence statistique une et deux dimensions*, De Boeck, 2006.

[37] A. Das and C. Madhavan, *Public-Key Cryptography: Theory and Practice*, Pearson Education, 2009.

[38] A. Dehbaoui, S. Tiran, P. Maurine, F.-X. Standaert, and N. Veyrat-Charvillon, *Spectral Coherence Analysis - First Experimental Results* - . Cryptology ePrint Archive, Report 2011/056, 2011. http://eprint.iacr.org/.

[39] Dempster, Laird, and Rudin, *Maximum likelihood from incomplete data via the em algorithm*, Journal of the Royal Statistical Society, B39 (1977), pp. 1–38.

[40] J. Doget, E. Prouff, M. Rivain, and F.-X. Standaert, *Univariate side channel attacks and leakage modeling*, Journal of Cryptographic Engineering, 1 (2011), pp. 123–144.

[41] H. F. Durrant-Whyte and T. C. Henderson, *Multisensor data fusion*, in pringer Handbook of Robotics, B. Siciliano and O. Khatib, eds., 2008, pp. 585–610.

[42] W. V. Eck, *Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?*, in Computers Security, 1985.

[43] J. S. Edward W. Kamen, *Introduction to optimal estimation Advanced textbooks in control and signal processing Control and Signal Processing Series*, Springer, 1999.

[44] D. Gabor, *Theory of communication*, J. Inst. Elect. Eng., 93 (1946), pp. 429–457.

[45] F. Galton, *Natural inheritance.*, London :Macmillan,. http://www.biodiversitylibrary.org/bibliography/32181.

[46] Gamry Instruments, *Technical Note: The Faraday Cage: What Is It? How Does It Work?*, 2010.

[47] M. K. Georg T. Becker and C. Paar, *Side-Channel based Watermarks for IP Protection*, in COSADE, February 4-5 2010, pp. 47–50. Darmstadt, Germany. http://cosade2010.cased.de/files/proceedings/cosade2010_paper_9.pdf.

[48] B. Gierlichs, L. Batina, B. Preneel, and I. Verbauwhede, *Revisiting Higher-Order DPA Attacks: Multivariate Mutual Information Analysis*, in CT-RSA, vol. 5985 of LNCS, Springer, March 1-5 2010, pp. 221–234. San Francisco, CA, USA.

[49] B. Gierlichs, L. Batina, and P. Tuyls, *Mutual Information Analysis – A Universal Differential Side-Channel Attack*. Cryptology ePrint Archive, Report 2007/198, 2007.

[50] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, *Mutual information analysis*, in CHES, 10th International Workshop, vol. 5154 of Lecture Notes in Computer Science, Springer, August 10-13 2008, pp. 426–442. Washington, D.C., USA.

[51] B. Gierlichs, E. De Mulder, B. Preneel, and I. Verbauwhede, *Empirical comparison of side channel analysis distinguishers on DES in hardware*, in ECCTD. European Conference on Circuit Theory and Design, IEEE, ed., August 23-27 2009, pp. 391–394. Antalya, Turkey.

[52] C. W. Gini, *Variability and mutability, contribution to the study of statistical distributions and relations*, Studi Economico-Giuridici della R. Universita de Cagliari, (1912). Reviewed in: Light, R.J., Margolin, B.H.: An Analysis of

Variance for Categorical Data. *J. American Statistical Association*, Vol. 66 pp. 534-544 (1971).

[53] F. Gravetter and L. Wallnau, *Essentials of statistics for the behavioral sciences*, Thomson/Wadsworth, 2008.

[54] S. Guilley, *Geometrical Counter-Measures against Side-Channel Attacks*, PhD thesis, ENST / CNRS LTCI, January 2007. 219 pages; Id: `2007 E 003`, `http://pastel.paristech.org/2562/`.

[55] S. Guilley, S. Chaudhuri, L. Sauvage, P. Hoogvorst, R. Pacalet, and G. M. Bertoni, *Security Evaluation of WDDL and SecLib Countermeasures against Power Attacks*, IEEE Transactions on Computers, 57 (2008), pp. 1482–1497.

[56] S. Guilley, P. Hoogvorst, and R. Pacalet, *A Fast Pipelined Multi-Mode DES Architecture Operating in IP Representation*, Integration, The VLSI Journal, 40 (2007), pp. 479–489. DOI: 10.1016/j.vlsi.2006.06.004.

[57] S. Guilley, K. Khalfallah, V. Lomne, and J.-L. Danger, *Formal Framework for the Evaluation of Waveform Resynchronization Algorithms*, in WISTP, June 1 2011.

[58] S. Guilley, L. Sauvage, J.-L. Danger, and N. Selmane, *Fault Injection Resilience*, in FDTC, IEEE Computer Society, August 21 2010, pp. 51–65. Santa Barbara, CA, USA. DOI: 10.1109/FDTC.2010.15.

[59] S. Guilley, L. Sauvage, J.-L. Danger, N. Selmane, and R. Pacalet, *Silicon-level solutions to counteract passive and active attacks*, in FDTC, 5th Workshop on Fault Detection and Tolerance in Cryptography, IEEE-CS, Washington DC, USA, aug 2008, pp. 3–17. (Up-to-date version on HAL: `http://hal.archives-ouvertes.fr/hal-00311431/en/`).

[60] L. Hemming, I. E. C. Society, I. Antennas, and P. Society, *Electromagnetic anechoic chambers: a fundamental design and specification guide*, IEEE Press, 2002.

[61] N. Homma, S. Nagashima, Y. Imai, T. Aoki, and A. Satoh, *High-Resolution Side-Channel Attack Using Phase-Based Waveform Matching*, in CHES, vol. 4249 of LNCS, Springer, October 10-13 2006, pp. 187–200. Yokohama, Japan.

[62] P. Hoogvorst, *The Variance Power Attack*, in COSADE, February 4-5 2010, pp. 4–9. Darmstadt, Germany. http://cosade2010.cased.de/files/proceedings/cosade2010_paper_2.pdf.

[63] Y. Ishai, A. Sahai, and D. Wagner, *Private Circuits: Securing Hardware against Probing Attacks*, in CRYPTO, vol. 2729 of LNCS, Springer, August 17–21 2003, pp. 463–481. Santa Barbara, California, USA.

[64] A. Jakulin and I. Bratko, *Analyzing attribute dependencies*, in PKDD 2003, volume 2838 of LNAI, Springer-Verlag, 2003, pp. 229–240.

[65] G. M. James, *Curve Alignment by Moments*, Annals of Applied Statistics, 1 (2007), pp. 480–501.

[66] J.J.Boutros, *A tutorial on iterative probabilistic decoding and channel estimation: Graph representation, information flow and probabilistic algorithm in decoding and communication*, (2005).

[67] A. Juels, D. Molnar, and D. Wagner, *Security and privacy issues in e-passports*, in Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks, Washington, DC, USA, 2005, IEEE Computer Society, pp. 74–88.

[68] E. Kamen and J. Su, *Introduction to optimal estimation*, Advanced textbooks in control and signal processing, Springer, 1999.

[69] R. Khattree and D. N.Naik, *Multivariate data reduction and descrimination*, 2000.

[70] W. Klecka, *Discriminant analysis*, Quantitative applications in the social sciences, Sage Publications, 1980.

[71] P. C. Kocher, J. Jaffe, and B. Jun, *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*, in Proceedings of CRYPTO'96, vol. 1109 of LNCS, Springer-Verlag, 1996, pp. 104–113. (PDF).

[72] ——, *Differential Power Analysis*, in Proceedings of CRYPTO'99, vol. 1666 of LNCS, Springer-Verlag, 1999, pp. 388–397.

[73] I. Koichi, T. Masahiko, and T. Naoya, *Encryption secured against DPA*, June 10 2008. Fujitsu US Patent 7386130, http://www.patentstorm.us/patents/7386130/fulltext.html.

[74] L.Batina and B.Gierlichs, *Differential Cluster Analysis*, in Cryptographic Hardware and Embedded Systems – CHES 2009, C. Clavier and K. Gaj, eds., vol. 5747 of Lecture Notes in Computer Science, Lausanne, Switzerland, 2009, Springer-Verlag, pp. 112–127.

[75] T.-H. Le, J. Cledire, C. Servire, and J.-L. Lacoume, *Higher order statistics for side channel analysis enhancement*, in e-Smart, September 2006. Sophia-Antipolis, France.

[76] ——, *Noise Reduction in Side Channel Attack using Fourth-order Cumulant*, IEEE Transaction on Information Forensics and Security, 2 (2007), pp. 710–720. DOI: 10.1109/TIFS.2007.910252.

[77] D. S. Lee, J. M. Park, and P. A. Vanrolleghem, *Adaptive multiscale principal component analysis for on-line monitoring of a sequencing batch reactor.*, J Biotechnol, 116 (2005), pp. 195–210.

[78] Y. Li, K. Sakiyama, L. Batina, D. Nakatsu, and K. Ohta, *Power Variance Analysis Breaks a Masked ASIC Implementation of AES*, in DATE'10, IEEE Computer Society, March 8-12 2010. Dresden, Germany.

[79] Y. Li, K. Sakiyama, L. Batina, D. Nakatsu, and K. Ohta, *Power Variance Analysis breaks a masked ASIC implementation of AES*, in DATE, IEEE, March 8-12 2010, pp. 1059–1064. Dresden, Germany.

[80] S. Lin and X. Huang, *Advanced Research on Computer Education, Simulation and Modeling: International Conference, CESM 2011, Wuhan, China, June 18-19, 2011. Proceedings*, no. ptie. 2 in Communications in Computer and Information Science Series, Springer, 2011.

[81] V. Lomn, A. Dehbaoui, P. Maurine, L. Torres, and M. Robert, *Differential Power Analysis enhancement with statistical preprocessing*, in DATE, IEEE, ed., March 8-12 2010.

[82] H. Maghrebi, J.-L. Danger, F. Flament, and S. Guilley, *Evaluation of Countermeasures Implementation Based on Boolean Masking to Thwart First and Second Order Side-Channel Attacks*, in SCS, IEEE, November 6–8 2009, pp. 1–6. Jerba, Tunisia. Complete version online: http://hal.archives-ouvertes.fr/hal-00425523/en/. DOI: 10.1109/ICSCS.2009.5412597.

[83] H. Maghrebi, J. L. Danger, F. Flament, and S. Guilley, *Evaluation of Countermeasures Implementation Based on Boolean Masking to Thwart First and Second Order Side-Channel Attacks*, in SCS, IEEE, November 6–8 2009, pp. 1–6. Jerba, Tunisia. Complete version online: http://hal.archives-ouvertes.fr/hal-00425523/en/. DOI: 10.1109/ICSCS.2009.5412597.

[84] H. Maghrebi, S. Guilley, J.-L. Danger, and F. Flament, *Entropy-based Power Attack*, in HOST, IEEE Computer Society, June 13-14 2010, pp. 1–6. Anaheim Convention Center, Anaheim, CA, USA. DOI: 10.1109/HST.2010.5513124.

[85] S. Mangard, *Hardware Countermeasures against DPA – A Statistical Analysis of Their Effectiveness*, in CT-RSA, vol. 2964 of Lecture Notes in Computer Science, Springer, 2004, pp. 222–235. San Francisco, CA, USA.

[86] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, Springer, December 2006. ISBN 0-387-30857-1, http://www.dpabook.org/.

[87] S. Mangard, E. Oswald, and F.-X. Standaert, *One for All - All for One: Unifying Standard DPA Attacks*. Cryptology ePrint Archive, Report 2009/449, 2009. http://eprint.iacr.org/2009/449. To appear in "IET Information Security".

[88] S. Mangard and K. Schramm, *Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations*, in CHES, vol. 4249 of LNCS, Springer, October 10-13 2006, pp. 76–90. Yokohama, Japan.

[89] E. Mateos and C. Geboty, *Side channel analysis using giant magneto-resistive (gmr) sensors*, in Cosade, 2011. Darmstadt, GERMANY.

[90] E. Mateos and C. H. Gebotys, *A new correlation frequency analysis of the side channel*, in Proceedings of the 5th Workshop on Embedded Systems Security, WESS '10, New York, NY, USA, 2010, ACM, pp. 4:1–4:8.

[91] E. Mateos and C. H. Gebotys, *Side channel analysis using giant magneto-resistive (gmr) sensors*, Design, (2011).

[92] J. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control*, Pearson Education, 1995.

[93] T. Messerges, E. Dabbish, and R. Sloan, *Examining Smart-Card Security under the Threat of Power Analysis Attacks*, IEEE Trans. Computers, 51 (2002), pp. 541–552.

[94] T. S. Messerges, *Using second-order power analysis to attack dpa resistant software*, in Proceedings of the Second International Workshop on Cryptographic Hardware and Embedded Systems, CHES '00, London, UK, 2000, Springer-Verlag, pp. 238–251.

[95] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, *Investigations of Power Analysis Attacks on Smartcards*, in USENIX — Smartcard'99, May 10–11 1999, pp. 151–162. Chicago, Illinois, USA (Online PDF).

[96] T. Moon and W. Stirling, *Mathematical methods and algorithms for signal processing*, Prentice Hall, 2000.

[97] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice Hall, August 1999.

[98] E. D. Mulder, B. Gierlichs, B. Preneel, and I. Verbauwhede, *Practical DPA Attacks on MDPL*, in First International Workshop on Information Forensics and Security (WIFS), IEEE Signal Processing Society, December 6-9 2009. London, United Kingdom. Also http://eprint.iacr.org/2009/231.

[99] J. Myers and A. Well, *Research design and statistical analysis*, L. Erlbaum Associates, 1995.

[100] D. Naccache, *Why CIM-PACA?; www.arcsis.org/uploads/media/1_PresDNaccache300908.pdf*.

[101] H. N. Nagaraja, *Functions of concomitants of order statistics*, Journal of the Indian Society for Probability and Statistics, 7 (2003), pp. 15–32.

[102] M. Nassar, S. Bhasin, J.-L. Danger, G. Duc, and S. Guilley, *BCDL: A high performance balanced DPL with global precharge and without early-evaluation*, in DATE'10, IEEE Computer Society, March 8-12 2010, pp. 849–854. Dresden, Germany.

[103] Y. Nievergelt, *Wavelets made easy*, Birkhäuser, 1999.

[104] NIST/ITL/CSD, *Data Encryption Standard. FIPS PUB 46-3*, Oct 1999. http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf.

[105] ——, *Advanced Encryption Standard (AES). FIPS PUB 197*, Nov 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[106] ——, *Recommendation for Block Cipher Modes of Operation.* Methods and Techniques, December 2001. (Online reference).

[107] ——, *Security Requirements for Cryptographic Modules. FIPS PUB 140-2*, Dec. 2002. http://csrc.nist.gov/cryptval/140-2.htm.

[108] K. Pearson, *Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity, and Panmixia*, Royal Society of London Philosophical Transactions Series A, 187 (1896), pp. 253–318.

[109] . Peeters, F.-X. Standaert, and J.-J. Quisquater, *Power and electromagnetic analysis: Improved model, consequences and comparisons*, Integration, The VLSI Journal, special issue on "*Embedded Cryptographic Hardware*", 40 (2007), pp. 52–60. DOI: 10.1016/j.vlsi.2005.12.013.

[110] H. Pelletier and X. Charvet, *Improving the DPA attack using Wavelet transform*, September 2005. NIST's Physical Security Testing Workshop. Website: http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-3/physec/papers/physecpaper14.pdf.

[111] T. Popp and S. Mangard, *Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints*, in Proceedings of CHES'05, vol. 3659 of LNCS, Springer, August 29 – September 1 2005, pp. 172–186. Edinburgh, Scotland, UK.

[112] J. Proakis and M. Salehi, *Digital communications*, McGraw-Hill higher education, McGraw-Hill, 2008.

[113] E. Prouff and M. Rivain, *Theoretical and Practical Aspects of Mutual Information Based Side Channel Analysis*, in ACNS, Springer, ed., vol. 5536 of LNCS, June 2-5 2009, pp. 499–518. Paris-Rocquencourt, France.

[114] F. Rachidi, *Compatibilit lectromagntique*, Course note of École Polytechnique Fdrale de Lausanne, 2006.

[115] M. Rahman, *Applications of Fourier Transforms to Generalized Functions*, WIT Press, 2011.

[116] C. Rechberger and E. Oswald, *Practical Template Attacks*, in WISA, vol. 3325 of LNCS, Springer, august 2004, pp. 443–457.

[117] K. H. Rhee and D. Nyang, eds., *Information Security and Cryptology - ICISC 2010 - 13th International Conference, Seoul, Korea, December 1-3, 2010, Revised Selected Papers*, vol. 6829 of Lecture Notes in Computer Science, Springer, 2011.

[118] M. Rivain, E. Prouff, and J. Doget, *Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers*, in CHES, vol. 5747 of Lecture Notes in Computer Science, Springer, September 6-9 2009, pp. 171–188. Lausanne, Switzerland.

[119] A. Roche, G. Malandain, X. Pennec, and N. Ayache, *Multimodal image registration by maximization of the correlation ratio*, 1998.

[120] J. L. Rodgers and A. W. Nicewander, *Thirteen Ways to Look at the Correlation Coefficient*, The American Statistician, 42 (1988), pp. 59–66.

[121] J. L. Rodgers and W. A. Nicewander, *Thirteen ways to look at the correlation coefficient*, The American Statistician, 42 (1988), pp. 59–66.

[122] B. Rosner, *Fundamentals of biostatistics*, Brooks/Cole Cengage Learning, 2010.

[123] R. Russo, *Statistics for the behavioural sciences: an introduction*, Psychology Press, 2003.

[124] S. E. Y. S., *A measure of association base on gini's mean difference*, Communications in statistics. Theory and methods, 16 (1987), pp. 207 – 231.

[125] A. Salomaa, *Public-key cryptography*, Texts in theoretical computer science, Springer, 1996.

[126] G. SAPORTA, *Probabilits analyse des donnes et statistiques*, 2008.

[127] G. Saporta, *Data mining et statistique dcisionnelle. L'intelligence des donnes*, Technip, 2010.

[128] SASEBO board from the Japanese RCIS-AIST: http://www.rcis.aist.go.jp/special/SASEBO/index-en.html.

[129] A. Satoh, *Side-channel Attack Standard Evaluation Board, SASEBO (Project of the AIST); SASEBO-G experimental board – http://www.rcis.aist.go.jp/special/SASEBO/SASEBO-G-en.html*.

[130] L. Sauvage, S. Guilley, and Y. Mathieu, *ElectroMagnetic Radiations of FPGAs: High Spatial Resolution Cartography and Attack of a Cryptographic Module*, ACM Trans. Reconfigurable Technol. Syst., 2 (2009), pp. 1–24. Full text in http://hal.archives-ouvertes.fr/hal-00319164/en/.

[131] P. Schaumont and K. Tiri, *Masking and Dual Rail Logic Don't Add Up*, in CHES, vol. 4727 of LNCS, Springer, September 10-13 2007, pp. 95–106. Vienna, Austria.

[132] W. Schindler, *A Combined Timing and Power Attack*, in Public Key Cryptography, 2002, pp. 263–279.

[133] W. Schindler, K. Lemke, and C. Paar, *A Stochastic Model for Differential Side Channel Cryptanalysis*, in CHES, LNCS, ed., vol. 3659 of LNCS, Springer, Sept 2005, pp. 30–46. Edinburgh, Scotland, UK.

[134] L. Shang, J. Jaeger, and R. Krebs, *Efficiency analysis of data compression of power system transients using wavelet transform*, in Power Tech Conference Proceedings, 2003 IEEE Bologna, vol. 4, june 2003, p. 6 pp. Vol.4.

[135] A. Sharma and M. Prakash, *Linear Transformation*, Discovery Publishing House, 2007.

[136] J. Shlens, *A tutorial in Principal Component Analysis*, (2005).

[137] D. Simon, *Kalman filtering*, Embedded Sytems Programming, pages 72-79, (June 2001).

[138] S.Kolenikov and G.Angeles, *The use of discrete data in PCA for socio-economic status evaluation*, (2005).

[139] P. S.Maybeck, *Stochastic models, estimation, and control*, ACADEMIC PRESS, 1979.

[140] L. I. Smith, *A tutorial in Principal Component Analysis*, (2002).

[141] D. Sorensen and D. Gianola, *Likelihood, Bayesian and MCMC methods in quantitative genetics*, Statistics for biology and health, Springer-Verlag, 2002.

[142] F.-X. Standaert and C. Archambeau, *Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages*, in CHES, vol. 5154 of Lecture Notes in Computer Science, Springer, August 10–13 2008, pp. 411–425. Washington, D.C., USA.

[143] F.-X. Standaert, P. Bulens, G. de Meulenaer, and N. Veyrat-Charvillon, *Improving the Rules of the DPA Contest*. Cryptology ePrint Archive, Report 2008/517, 2008. http://eprint.iacr.org/2008/517.

[144] F.-X. Standaert, B. Gierlichs, and I. 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, vol. 5461 of LNCS, Springer, December 3-5 2008, pp. 253–267. Seoul, Korea.

[145] F.-X. Standaert, F. Koeune, and W. Schindler, *How to Compare Profiled Side-Channel Attacks?*, in ACNS, Springer, ed., vol. 5536 of LNCS, June 2-5 2009, pp. 485–498. Paris-Rocquencourt, France.

[146] F.-X. Standaert, T. Malkin, and M. Yung, *A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks*, in EUROCRYPT, vol. 5479 of LNCS, Springer, April 26-30 2009, pp. 443–461. Cologne, Germany.

[147] G. S. Stphane Tuffry, *Data mining et statistique dcisionnelle. L'intelligence des donnes*, Technip, 2010.

[148] TELECOM ParisTech SEN research group, *DPA Contest (1^{st} edition)*, 2008–2009. http://www.DPAcontest.org/.

[149] ——, *DPA Contest (2^{nd} edition)*, 2009–2010. http://www.DPAcontest.org/v2/.

[150] C. Thompson and L. Shure, *Image Processing Toolbox: For Use with MATLAB;[user's Guide]*, MathWorks, 1995.

[151] C. C. Tiu, *A new frequency-based side channel attack for embedded systems. master degree thesis, deparment of electrical and computer engineering,university of waterloo,waterloo*, tech. rep., 2005.

[152] P. Topiwala, *Wavelet image and video compression*, Kluwer international series in engineering and computer science, Kluwer Academic, 1998.

[153] J. G. J. van Woudenberg, M. F. Witteman, and B. Bakker, *Improving Differential Power Analysis by Elastic Alignment*, in CT-RSA, 2011, pp. 104–119.

[154] G. Vasilescu, *Electronic Noise and Interfering Signals*, Springer.

[155] S. Yitzhaki, *Gini's mean difference: a superior measure of variability for non-normal distributions*, International Journal of Statistics, 2 (2003), pp. 285 – 316.

[156] P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach*, American Institute of Aeronautics and Astronautics, December 2006.

[157] W. Zeng, H. Yu, and C. Lin, *Multimedia security technologies for digital rights management*, Electronics & Electrical, Academic Press, 2006.

[158] Zeng Guang Hou, *PCA for data fusion and navigation of mobile robots*, vol. 3495 of LNCS, Springer, 2005, pp. 610–611.