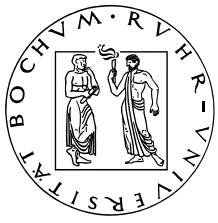
Models and Algorithms for Physical Cryptanalysis



Dissertation
zur
Erlangung des Grades eines
Doktor-Ingenieurs
der
Fakultät für Elektrotechnik und Informationstechnik
an der Ruhr-Universität Bochum

von

Kerstin Lemke-Rust

Bochum, Januar 2007

Thesis Advisor: Prof. Dr.-Ing. Christof Paar, Ruhr University Bochum, Germany
External Referee: Prof. Dr. David Naccache, École Normale Supérieure, Paris, France

Author contact information: Kerstin.Lemke-Rust@gmx.de

Abstract

This thesis is dedicated to models and algorithms for the use in physical cryptanalysis which is a new evolving discipline in implementation security of information systems. It is based on physically observable and manipulable properties of a cryptographic implementation.

Physical observables, such as the power consumption or electromagnetic emanation of a cryptographic device are so-called 'side channels'. They contain exploitable information about internal states of an implementation at runtime. Physical effects can also be used for the injection of faults. Fault injection is successful if it recovers internal states by examining the effects of an erroneous state propagating through the computation.

This thesis provides a unified framework for side channel and fault cryptanalysis. Its objective is to improve the understanding of physically enabled cryptanalysis and to provide new models and algorithms. A major motivation for this work is that methodical improvements for physical cryptanalysis can also help in developing efficient countermeasures for securing cryptographic implementations.

This work examines differential side channel analysis of boolean and arithmetic operations which are typical primitives in cryptographic algorithms. Different characteristics of these operations can support a side channel analysis, even of unknown ciphers. It also provides evidence that existing simple leakage models are suboptimal in practice and that there is a need for improvements.

A main research contribution of this thesis is a new stochastic model for multivariate side channel analysis, allowing for an approximation of the real side channel leakage for any given internal state of an implementation. The proposed stochastic methods can capture both different time instants and different internal states as part of a multivariate side channel cryptanalysis. Furthermore, methods are made available in case the implementation applies masking techniques to hide internal states. Experimental results are included confirming the efficiency of these stochastic algorithms. Particularly, it is proved that the new algorithms are clearly superior to univariate differential side channel analysis. A performance analysis for templates and stochastic methods has been added. This led to further optimizations so that the final algorithms can be seen as the most efficient ones for side channel cryptanalysis.

The last chapter addresses modelling of fault channel cryptanalysis, with a focus on fault injection techniques based on radiation and particle impact into integrated circuits. Strategies and countermeasures are evaluated as result of this new physical model.

Kurzdarstellung der Dissertation

Diese Dissertation ist der Modellierung und Entwicklung von Algorithmen für die physikalische Kryptoanalyse gewidmet. Die physikalische Kryptoanalyse ist ein neues aufstrebendes Forschungsgebiet innerhalb der Implementierungssicherheit von Informationssystemen. Sie basiert auf der Nutzung von physikalisch messbaren oder beeinflussbaren Eigenschaften einer kryptographischen Implementierung.

Physikalische Messgrößen wie die Leistungsaufnahme und elektromagnetische Abstrahlung der kryptographischen Implementierung stellen sogenannte Seitenkanäle dar, die verwertbare Informationen über interne Zustände zur Laufzeit der Implementierung beinhalten. Physikalische Effekte können darüber hinaus auch zur Injektion von Fehlern eingesetzt werden. Eine Fehlerinjektion ist erfolgreich, wenn ein Fehler in der Implementierung fortschreitet und die Ausgabe von einem fehlerhaften Kryptogramm erfolgt.

Diese Dissertation bietet einen gemeinsamen Rahmen für seitenkanal- und fehlerbasierte Kryptoanalyse. Das Ziel ist es, das Verständnis von physikalischer Kryptoanalyse zu fördern und neue Modelle und Algorithmen zu entwickeln. Eine wesentliche Motivation für die Arbeit liegt darin, dass eine Verbesserung in der Methodik der physikalischen Kryptoanalyse auch bei der Entwicklung von effizienten Gegenmaßnahmen hilft.

Diese Arbeit behandelt die differentielle Seitenkanalanalyse bei logischen und arithmetischen Operationen, die typische atomare Bausteine in kryptographischen Algorithmen darstellen. Die unterschiedlichen Charakteristiken dieser Operationen können eine Seitenkanalanalyse von unbekannten Kryptoalgorithmen unterstützen. Insbesondere wird nachgewiesen, dass einfache Modelle für Seitenkanäle in der Praxis nicht optimal sind und dass es einen Bedarf an Verbesserungen gibt.

Ein Hauptbeitrag dieser Dissertation ist ein neues stochastisches Modell für multivariate Seitenkanalanalysen, das eine Approximation des Seitenkanals für jeden internen Zustand einer Implementierung ermöglicht. Hierdurch werden Methoden bereitgestellt, die sowohl unterschiedliche Zeitpunkte als auch unterschiedliche interne Zustände in einer multivariaten Seitenkanalanalyse erfassen können. Die Algorithmen sind auch für den Fall einsetzbar, dass die Implementierung interne Zustände in einer maskierten Darstellung bearbeitet. Die experimentellen Ergeb-

nisse bestätigen die Effizienz der stochastischen Algorithmen. Insbesondere wird nachgewiesen, dass die neuen Algorithmen der univariaten differentiellen Seitenkanalanalyse deutlich überlegen sind. Es ist ferner ein Effizienzvergleich zwischen stochastischen Methoden und sogenannten Templates durchgeführt worden. Dies hat zu weiteren Optimierungen geführt, so dass die entwickelten Algorithmen als die effizientesten in der Seitenkanalanalyse gesehen werden können.

Das abschließende Kapitel dieser Arbeit dient der Modellierung von fehlerbasierter Kryptoanalyse. Der Schwerpunkt liegt auf Fehlerinjektionsstechniken mit elektromagnetischer Strahlung und Teilchenstrahlung in integrierten Schaltkreisen. Strategien für Gegenmaßnahmen werden als Ergebnis dieses neuen physikalischen Modells bewertet.

To my parents and to Christian.

Acknowledgements

I am very grateful to Prof. Christof Paar, Chair for Communication Security at the Ruhr University Bochum for various kinds of support that I received. First of all, I deeply thank Christof for accepting me as a Ph.D. student. Working in his group at the university was a great pleasure and motivation for me after eight years in the industry. I have enjoyed many fruitful discussions and I thank for all the helpful suggestions while working on this thesis. Last but not least I am thankful for the wonderful open-minded atmosphere. My deep gratitudes also go to Prof. David Naccache for reviewing this thesis and attending my Ph.D. defense on June 1, 2007.

I would like to express my gratitudes to all present and former members of Christof's research group in Bochum. Many of them have become good friends of mine over the last years. Especially, I am grateful to Irmgard Kühn for the warm atmosphere, for running everything smoothly and all her help with the organization. I thank Kai Schramm for the joint work on analyzing side channels and for sharing ideas as well as tools and equipment. Greatly mentionable is also the work with Benedikt Gierlichs on comparing stochastic methods and templates for differential side channel analysis that has led to a CHES 2006 paper and to valuable results for this thesis. It has been a big pleasure and fun to do research with Prof. Ahmad-Reza Sadeghi. In many refreshing discussions, Ahmad opened my eyes to new views and I appreciate all the lessons in learning of how to write a paper. Further acknowledgements go to Sandeep Kumar for deeper insights to VHDL programming and the joint work for the eSTREAM project. Editing of our book on *Embedded Security in Cars* was another big experience in the previous years and I especially enjoyed the continuous work with Marko Wolf on this. My sincere thanks also to Timo Kasper, Thomas Eisenbarth, Dario Carluccio, Christian Stüble, Markus Kasper, Gordon Meiser, and Yifei Liu for the joint research on various topics. Special thanks to Marcel Selhorst, Hans Christian Röpke, Karsten Tellmann, and Horst Edelmann for helping me with the computer and web administration. And to Benedikt and Timo for proofreading a preliminary version of this manuscript during Christmas time 2006! All the best to all of you!

Prof. Werner Schindler deserves my acknowledgements for the joint development of stochastic methods for differential side channel analysis.

The mathematical model would not have been possible without him. I sincerely thank for the various discussions we had in preparing the paper for CHES 2005.

The work described in this thesis was supported through the IST Contract IST-2002-507932 ECRYPT, the European Network of Excellence in Cryptology. Thanks to all the researchers in the ECRYPT project for sharing their ideas. Especially to Lejla Batina, Benedikt Gierlichs, Tanja Lange, Stefan Mangard, Nele Mentens, Elke de Mulder, Elisabeth Oswald, Kazuo Sakiyama, and François-Xavier Standaert. Moreover, to Tanja for patiently answering all my boring questions on ECRYPT reporting and formalisms. Thanks to many others of the crypto community for the good times at the various workshops and conferences.

Finally, I would like to express my kind gratitudes to my parents and my husband Christian who encouraged me all the time I spent working on this thesis. My deepest love is with them. In times like these and in times like those. What will be will be and so it goes. And it always goes on and on.

Table of Contents

1 Introduction **1**
- 1.1 Motivation . . . 2
- 1.2 Outline and Research Contributions . . . 3

2 Background **7**
- 2.1 Mathematical Statistics . . . 8
  - 2.1.1 Introduction . . . 8
  - 2.1.2 Hypothesis Tests . . . 13
  - 2.1.3 Vectorial Statistics . . . 19
- 2.2 Entropy and Information . . . 25
- 2.3 CMOS VLSI Technology . . . 27

3 Related Work **29**
- 3.1 Physical Security . . . 30
  - 3.1.1 Attack Scenarios . . . 31
  - 3.1.2 Security Objectives . . . 33
  - 3.1.3 Security Requirements . . . 33
  - 3.1.4 Tamper Evidence . . . 35
  - 3.1.5 Tamper Response . . . 36
  - 3.1.6 Tamper Resistance . . . 38
- 3.2 Physical Cryptanalysis . . . 40
  - 3.2.1 Adversary Model . . . 41
- 3.3 Side Channel Cryptanalysis . . . 45
  - 3.3.1 Refinements of Adversary Model . . . 46
  - 3.3.2 Timing Analysis . . . 47
  - 3.3.3 Simple Side Channel Analysis . . . 51

3.3.4 Differential Side Channel Analysis 56
3.3.5 Differential Collision Analysis 65
3.3.6 Countermeasures 66
3.3.7 Second Order Differential Side Channel Analysis . 67
3.3.8 Multivariate Analysis 69
3.4 Fault Channel Cryptanalysis 72
3.4.1 Refinements of Adversary Model 72
3.4.2 Techniques for Fault Induction 74
3.4.3 Modelling of Faults 77
3.4.4 Simple Fault Analysis 77
3.4.5 Differential Fault Analysis 80
3.4.6 Countermeasures and Further Directions 81

4 DSCA on Boolean and Arithmetic Operations 83
4.1 Contribution 84
4.2 Previous Work 85
4.3 DSCA using n-bit sized Basic Operations 87
4.3.1 Boolean Operation XOR 89
4.3.2 Addition modulo 2^n 91
4.3.3 Modular Multiplication 92
4.4 Application to Cryptographic Algorithms 96
4.4.1 IDEA 96
4.4.2 RC6 97
4.4.3 HMAC-Construction 99
4.5 Experimental Results of an IDEA Implementation 104
4.5.1 8051 Microcontroller 105
4.5.2 AVR Microcontroller 105

5 Stochastic Methods for Differential Side Channel Analysis 111
5.1 Contribution 112
5.2 Previous Work 113
5.3 The Stochastic Model and Algorithms 115
5.3.1 The Principle Idea 115
5.3.2 Distance between the true function h_t and functions $h' \in \mathcal{F}_{u;t}$ 117
5.3.3 The EIS Property 119
5.3.4 Profiling Phase 121

5.3.5 Key Recovery Phase 127
5.3.6 Generalization to Masked Implementations 133
5.3.7 Generalization to Multi Channels 137
5.4 Experimental Analysis of an AES Implementation 140
5.4.1 Profiling Phase 140
5.4.2 Selection of Instants 144
5.4.3 Key Recovery Phase 150
5.5 Experimental Analysis of a Masked Implementation . . . 155
5.5.1 Profiling Phase 156
5.5.2 Key Recovery Phase 158

6 Templates vs. Stochastic Methods 165
6.1 Contribution 166
6.2 Previous Work 167
6.3 Application of Templates and Stochastic Methods 168
6.3.1 Template Attack 168
6.3.2 Stochastic Methods 171
6.3.3 Compendium of Differences 172
6.4 Performance Evaluation 173
6.4.1 Metrics, Parameters, and Factors to Study 173
6.4.2 Experimental Design 175
6.5 Experimental Results for Original Attacks 175
6.5.1 Comparison of Profiling Efficiency 175
6.5.2 Comparison of Classification Efficiency 177
6.5.3 Weaknesses and Strengths 178
6.6 Experimental Results for Optimized Attacks 178
6.6.1 Templates vs. T-Test based Templates 181
6.6.2 First-order Stochastic Method vs. T-Test based High-order Stochastic Method 182
6.6.3 Overall Comparison 184

7 A Model on Physical Security Bounds against Tampering 187
7.1 Contribution 188
7.2 Previous Work 189
7.3 Adversary Model 190
7.3.1 Objectives of the Adversary 192
7.3.2 Physical Means of the Adversary 193

7.4 Physical Security Bounds 198
7.4.1 Evaluation of Countermeasure Strategies 199

8 Conclusion and Open Problems **203**
8.1 Conclusion 204
8.2 Open Problems 205

Bibliography **207**

List of Figures **219**

List of Tables **221**

List of Algorithms **223**

List of Abbreviations **225**

Curriculum Vitae **227**

Publications **229**

Chapter 1

Introduction

1.1 Motivation

The science of *cryptology* is driven by two opposed, but complementary disciplines, *cryptography* and *cryptanalysis*. While cryptography is the science of developing mathematical algorithms and protocols for protecting information, cryptanalysis is the science of breaking cryptographic schemes. Cryptanalysis thereby helps to improve the assurance in cryptographic tools.

In computer science, cryptology aims at providing *security*. Citing from [128], security is about "preventing adverse consequences from the intentional and unwarranted actions of others". While originating from the military, security has moved into commercial and governmental applications and cryptography is nowadays implemented in mobile phone smart cards, digital signature cards, electronic payment schemes, Internet services, electronic tickets, and – most recently – in identification cards. Those "intentional and unwarranted actions" are attacks. Security is implemented in order to defend assets such as secrets and cryptographic keys against attacks.

Different ways can lead to success in cryptanalysis. Among them are (i) mathematical analysis of the cryptographic scheme by using algebraic or stochastic methods, (ii) exhaustive search for all possible keys (brute force), (iii) social engineering that manipulates people to recover a secret, and (iv) implementation analysis that aims to extract secrets from the implementation of a cryptographic scheme.

Traditionally, cryptanalysis employs mathematical tools to evaluate the security claims by cryptographers. In mathematical cryptanalysis, physically enabled information flow is implicitly assumed to be nonexisting. However, once cryptographic schemes are implemented in integrated circuits, the resulting cryptographic implementation can no longer be solely seen as a mathematical object. Moreover and much more impressively, recent research results in the last decade have demonstrated that implementation attacks are a serious threat to cryptographic modules.

Physical information flow originates from measurable physical leakage of the cryptographic device, i.e., by observing the timing, the power consumption, and the electro-magnetic emanation. Such kind of attacks are referred to as *side channel cryptanalysis*. The observed physical leakage is the relevant input to cryptanalysis. Roughly speaking, side

channel cryptanalysis uses *physical observables* resulting from internal states of a cryptographic implementation as an information source for cryptanalysis.

Fault channel cryptanalysis uses *physical means* in order to modify internal states of an implementation aiming at additional information for cryptanalysis. Here, physical information flow may occur if a modified internal state is entered and propagates through the implementation. Many scenarios of fault channel cryptanalysis exploit erroneous cryptograms by means of mathematical cryptanalysis.

In response to such findings, cryptanalysis has been adapted to additionally include engineering tools to check whether the security claims for a cryptographic implementation are still valid when physical attacks are taken into account. This new class is said to be *implementation cryptanalysis*. If one just observes a cryptographic implementation one calls this a *passive implementation attack*. On the other hand, if an adversary actively stimulates an implementation by physical means the attack is named *active implementation attack*. Meanwhile, a cat-and-mouse game between implementation based cryptography and implementation based cryptanalysis can be observed. Some proposals for implementation countermeasures have in turn been defeated.

This thesis deals with modelling and provides algorithms for *physical cryptanalysis*. Physical cryptanalysis allows a passive or active physical interaction at any internal state of an implementation and covers both side channel cryptanalysis and fault channel cryptanalysis. Attacks have turned out to be astonishingly easy in previous years, especially on small integrated circuits such as smart cards, and may have been given an impression of being magic. Physical cryptanalysis is still somehow at the beginning of being a well-established and well-understood discipline and remains an area of active research. Many proposals for theoretical attacks have been emerged, not all of them have been – or could at all be – demonstrated in practice, yet. This is especially true for fault channel cryptanalysis.

1.2 Outline and Research Contributions

This thesis is dedicated to modelling of physical cryptanalysis. It aims (i) to improve the understanding, (ii) to improve the leakage models,

and (iii) to develop new algorithms.

Adversary success at physical cryptanalysis may crucially depend on the correctness of the leakage or fault models of a concrete implementation. Widely used models for side channel cryptanalysis either target single bits of an intermediate result or assume that each bit of a processor's word size contributes an identical portion to the overall leakage. This can be suboptimal if the real leakage of an implementation is more complicated. Here, this thesis builds up on the need for improvement of methodologies for describing hypothetical models and for assessing their quality as outlined by Oswald [107]. Similar considerations apply to fault channel cryptanalysis where most attacks assume a specific fault model. Even more demanding, many attacks are still of theoretical nature and may raise the question whether a certain attack is a real threat for a given cryptographic implementation.

Another objective of this thesis is to provide a state-of-the-art survey in physical cryptanalysis from the methodical point of view. Implementation security differs from algorithmic security, as for an assessment properties of the concrete circuit layout are decisive.

Parameters with an impact on efficiency in physical cryptanalysis include (i) the quantity of inherent leakage (chip dependent), (ii) the quality of the laboratory equipment (lab dependent), and (iii) the algorithms' ability to extract information (method dependent). Among them, this thesis deals with the development of algorithms. As part of this thesis, experiments have been carried out with standard microcontrollers that are commercially available and do not incorporate any physical countermeasures. These experiments are used for a proof-of-concept of proposed algorithms. Further improvements of the laboratory equipment used may be conceivable and the analysis of physically secured integrated circuits may require significantly enhanced efforts.

This thesis is organized as follows. Chapter 2 provides the background and consists of an introductory part to mathematical statistics in Section 2.1, an introduction to information theory in Section 2.2, and to the principles of CMOS VLSI design in Section 2.3.

Chapter 3 introduces related work regarding physical cryptanalysis. Therefore, first physical security for cryptographic devices is revisited in Section 3.1. Then a framework for physical cryptanalysis is given in Section 3.2. In Section 3.3 related work in the context of side channel cryptanalysis is presented, while Section 3.4 concentrates on fault

channel cryptanalysis.

Chapter 4 to Chapter 7 provide the new research contributions of this thesis. Each chapter is headed by a more detailed explanation of the research contribution in Section 4.1 to Section 7.1.

Chapter 4 deals with the question of how differential side channel cryptanalysis (DSCA) can be applied to boolean and arithmetic operations which are atomic operations of product ciphers such as IDEA and RC6 as well as for hash based message authentication codes. This chapter provides an analysis of multi-bit DSCA signals that are revealed in n-bit sized primitive operations such as exclusive-or, addition modulo 2^n, and modular multiplication assuming a Hamming weight model. The characteristics of DSCA results differ for these basic operations and can support side channel analysis of an unknown implementation and even for an unknown cipher. Experimentally, both an IDEA implementation on an 8051 microcontroller and on an AVR microcontroller are evaluated. Whereas the physical leakage of the 8051 microcontroller can be well approximated in this model, one observes more difficulties in case of an AVR microcontroller indicating that the Hamming weight model is not appropriate for the real physical leakage.

Chapter 5 presents a new stochastic model for optimizing the efficiency of differential side channel cryptanalysis by means of multivariate stochastic methods. The new idea is to profile the real physical leakage by approximation within a suitable chosen vector subspace that is spanned by basis functions of the targeted data space. This chapter includes the theoretical framework and introduces algorithms for a 'minimum principle' and a 'maximum likelihood principle'. The methods are also adapted to second-order side channel cryptanalysis in the presence of masking countermeasures. Experimental applications of the stochastic algorithms are given for two implementations on AVR microcontrollers. Section 5.4 contains an analysis of an (unmasked) AES implementation in different parameter settings such as the selection of instants and the choice of vector subspaces, whereas Section 5.5 shows how the stochastic methods can be applied to boolean masking that is the most general case for higher order analysis. It is demonstrated that the adaptation of probability densities provided by stochastic methods is clearly advantageous regarding to common methods for first-order and second-order DSCA. In summary, the new stochastic methods improve the understanding of the source of an attack and its true risk potential.

Chapter 6 deals with the performance of the new stochastic methods developed in Chapter 5 if compared to template attacks. More precisely, it considers the efficiency of templates and the stochastic maximum likelihood principle under identical physical conditions. Using the originally proposed attacks, it was found that for a low number of profiling measurements stochastic methods are more efficient, whereas in the case of high numbers of measurements templates achieve superior performance results. Additionally, Chapter 6 includes proposals for both methods under consideration. The most crucial improvement deals with the selection of time instants for the multivariate density. As the main result of optimizations, T-Test based templates are the method of choice if a high number of measurements is available for profiling. However, in case of a low number of measurements, stochastic methods are an alternative and can reach superior efficiency both in terms of profiling and classification. Moreover, stochastic methods are even applicable if the number of measurements at profiling is less than the number of subkeys. This is of high importance at block cipher designs with an enlarged bit size of subkeys.

While Chapter 4 to Chapter 6 deal with side channel cryptanalysis, Chapter 7 is dedicated to the modelling of fault channel cryptanalysis. Its main contribution is to present an adversary model with a strong focus on fault injection techniques based on radiation and particle impact and to define physical security parameters against tampering adversaries. Strategies for countermeasures are evaluated out of the new physical model. It is hoped that this framework is useful for mapping concrete impact probabilities of a given circuit as well as to improve the circuits' layouts.

The conclusion of this thesis is given in Chapter 8. Finally, Chapter 8 points to open problems for further research.

Chapter 2

Background

This chapter aims to provide a background on mathematical statistics, information theory, and CMOS VLSI design that is required for the understanding of this thesis. Mathematical statistics is important as both

- physical side channel observables and
- physical fault injections

are statistical quantities. Information theory is used to supply a measure of success for implementation attacks. CMOS VLSI is the leading technology for building cryptographic modules and deserves an introduction of its basics. For further reading on cryptographic algorithms and applications see [92, 127, 8].

2.1 Mathematical Statistics

This section provides the necessary background for mathematical statistics with a focus on relevant aspects for physical cryptanalysis. The presentation and notation mainly follows the lines of Pestman [113]. The principal objective of this section is to revisit and provide mathematical facts that are needed for a good understanding of mathematical statistics used in this thesis. The results are stated without proofs.

2.1.1 Introduction

In probability experiments, randomness plays a central rule. By repeating the probability experiment under exactly the same conditions, one generally does not observe identical outcomes of the experiment. A brief introduction to probability theory and mathematical statistics is provided below.

The set of all possible outcomes is denoted as *sample space* Ω. An *event* is understood to be a subset of Ω. Let A be an arbitrary subset of Ω. The complement of A in Ω is then defined as A^c. Probability theory introduces a probability space of admissible events by a collection of subsets of Ω to be a σ-algebra.

Definition 2.1. A collection $\mathfrak{A}$ of subsets of Ω is said to be a *σ-algebra* if it satisfies the following three properties:

(i) $\Omega \in \mathfrak{A}$,

(ii) If A $\in \mathfrak{A}$ then also $A^c \in \mathfrak{A}$,

(iii) If A_1, A_2, ... $\in \mathfrak{A}$ then also $\bigcup_{i=1}^{\infty} A_i \in \mathfrak{A}$.

Two simple examples for $\mathfrak{A}$ are the empty set $\varnothing$ and the sample space Ω.

The probability measure $\mathbb{P}$ is the chance that event A is going to occur, i.e., a real number in the interval [0,1]. More generally, $\mathbb{P}$ is a special case of a *measure*, i.e., a mapping $\mu : \mathfrak{A} \longrightarrow [0, \infty]$ fulfilling the two conditions

(i) $\mu(\varnothing) = 0$,

(ii) For every mutually disjoint collection A_1, A_2, ... $\in \mathfrak{A}$ one has

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i).$$

For a probability measure $\mathbb{P}$, additionally $\mu(\Omega) = 1$ holds.

A *probability space* associated with a probability experiment is a triplet $(\Omega, \mathfrak{A}, \mathbb{P})$ of sample space Ω, σ-algebra $\mathfrak{A}$, and probability measure $\mathbb{P}$ on $\mathfrak{A}$.

Before stochastic variables can be introduced, one has to provide a definition of the term $\mathfrak{A}$-*measurable*. Let $\vec{X} : \Omega \to \mathbb{R}^n$ be a function. For every subset $A \subset \mathbb{R}^n$, $\vec{X}^{-1}(A) := \{\omega \in \Omega : \vec{X}(\omega) \in A\}$. The function $\vec{X}$ is said to be $\mathfrak{A}$-*measurable* if $\vec{X}^{-1}(A) \in \mathfrak{A}$.

Precisely, a *stochastic variable* X is defined as an $\mathfrak{A}$-measurable function $X : \Omega \to \mathbb{R}$. Accordingly, a *stochastic vector* $\vec{X}$ is an $\mathfrak{A}$-measurable function $\vec{X} : \Omega \to \mathbb{R}^n$.

Throughout this thesis, stochastic variables are denoted with capital letters, while their instantiations are written in lower case letters, e.g., x. If the number of outcomes of the stochastic variable is finite, it is said to have a *discrete probability distribution* $\mathbb{P}_X$. If the number of outcomes is not finite, the stochastic variable has an *absolutely continuous probability distribution* $\mathbb{P}_X$.

For applications in mathematics and natural sciences, the most important continuous probability distribution is the *normal distribution*

that is given by the probability density $f(\bullet)$ as

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{2.1}$$

The normal distribution is also referred to as $N(\mu, \sigma^2)$. It is symmetric about its mean $x = \mu$. The value of σ is a measure of the width of the distribution. Small values of σ yield a sharp and narrow distribution about $x = \mu$, whereas large values of σ give a flat and wide distribution.

One discrete probability distribution is the *binomial distribution* with parameters $n \in \{1, 2, \dots\}$ and $\theta \in [0, 1]$ which is defined as

$$\mathbb{P}(X = k) = \binom{n}{k} \theta^k (1-\theta)^{n-k} \tag{2.2}$$

for all $k \in \{0, 1, 2, \dots, n\}$.

For continuous probability distributions, the expectation of a stochastic variable X is revealed by

$$\mathbb{E}(X) = \int x \, d\mathbb{P}_X(x) \tag{2.3}$$

provided that $\int |x| \, d\mathbb{P}_X(x) < \infty$. If an experiment is repeated very often, the empirical mean value becomes 'close' to the expectation value. Accordingly, for discrete probability distributions the expectation value is defined by replacing integration with summation over all discrete events in Eq.(2.3).

The variance of X, denoted as $\mathrm{Var}(X)$, is obtained as

$$\mathrm{Var}(X) := \mathbb{E}(X^2) - (\mathbb{E}(X))^2, \tag{2.4}$$

provided that $\mathbb{E}(X^2)$ exists. The variance provides a measure for the spread of the outcome x around the expectation value $\mathbb{E}(X)$, i.e., the standard deviation $\sigma(X) = \sqrt{\mathrm{Var}\,(X)}$.

If one considers two different stochastic variables, one may ask for an association between them. The covariance and the correlation coefficient give such a degree of association.

Definition 2.2. If the variance of the stochastic variables X and Y exists, then the covariance of X and Y is defined by

$$\mathrm{cov}(X, Y) := \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y).$$

Definition 2.3. If the variance of the stochastic variables X and Y exists, then the correlation coefficient of X and Y is defined by

$$\rho(X,Y) := \frac{\mathrm{cov}(X,Y)}{\sigma(X)\sigma(Y)}.$$

It can be shown that: $-1 \leq \rho(X,Y) \leq +1$. Absolute correlation coefficients near 1 indicate a strong linear relationship. Two stochastic variables are said to be *uncorrelated* if the correlation coefficient is close to zero.

A joint probability distribution of two stochastic variables X and Y is denoted by $\mathbb{P}_{X,Y}$.

Definition 2.4. The stochastic variables X and Y are said to be *statistically independent* if

$$\mathbb{P}_{X,Y} = \mathbb{P}_X \cdot \mathbb{P}_Y.$$

By introducing a *tensor product* of a system of functions $f_1, \ldots, f_n{:}\mathbb{R} \to \mathbb{R}$ as function $f{:}\mathbb{R}^n \to \mathbb{R}$, the following theorem for statistically independent stochastic variables is given.

Theorem 2.1. *Suppose $X_1, \ldots, X_n$ are stochastic variables with probability densities $f_{X_1}, \ldots, f_{X_n}$. The system $X_1, \ldots, X_n$ is statistically independent if and only if the stochastic n-vector $(X_1, \ldots, X_n)$ has a probability density $f_{X_1} \cdots f_{X_n}$.*

For statistically independent stochastic variables a variety of properties apply, e.g., Proposition 2.2 and Proposition 2.3.

Proposition 2.2. *If X and Y are statistically independent stochastic variables with existing variances then $cov(X,Y) = 0$.*

The opposite of Proposition 2.2 is generally not true. However, under certain assumptions (see [113], Theorem I.6.7) for normal distributed populations one can show that $\mathrm{cov}(X,Y) = 0$ can imply that X and Y are statistically independent.

Proposition 2.3. *Let X_1 and X_2 be statistically independent stochastic variables. If variable X_1 is $N(\mu_1, \sigma_1)$-distributed and variable X_2 is $N(\mu_2, \sigma_2)$-distributed, then the variable $S := X_1 + X_2$ is $N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$-distributed.*

Definition 2.5. Suppose X and Y enjoy a discrete or an absolutely continuous probability distribution. Then the function $f_X(\bullet|Y = y)$, defined by

$$f_X(x|Y = y) = \begin{cases} \frac{f_{X,Y}(x,y)}{f_Y(y)}, & \text{if } f_Y(y) > 0, \\ 0 & \text{if } f_Y(y) = 0, \end{cases}$$

is said to be the conditional probability density of X, given $Y = y$.

For conditional probability densities, the Bayesian theorem is fundamental. Here, $f_{X,Y}(x, y)$ is the joint distribution of X and Y and $f_X(x)$ and $f_Y(y)$ are the marginal probabilities.

Proposition 2.4. *If X and Y enjoy discrete or absolutely continuous probability distributions, then $f_{X,Y}(x, y) = f_X(x|Y = y)f_Y(y) = f_Y(y|X = x)f_X(x)$*

In the context of empirical experiments a *sample of size n* is introduced as a sequence $x_1, x_2, \ldots, x_n$ of instantiations of a stochastic variable X. The *sample mean* is defined as

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{2.5}$$

The *sample variance* is

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2. \tag{2.6}$$

Proposition 2.5 (for a general case) and Proposition 2.6 (for normally distributed populations) yield that the empirical variance of $\overline{x}$ is asymptotically reduced with $\frac{1}{n}$. Note that this is a widely used result for side channel cryptanalysis, as the precision of the empirical mean value increases with $\sqrt{n}$.

Proposition 2.5. *If $x_1, x_2, \ldots, x_n$ is a sample from a population with mean μ and variance σ^2, then $\mathbb{E}(\overline{x}) = \mu$ and $var(\overline{x}) = \frac{\sigma^2}{n}$.*

Proposition 2.6. *If $x_1, x_2, \ldots, x_n$ is a sample from a $N(\mu, \sigma^2)$-distributed population, then $\overline{x}$ is a $N(\mu, \frac{\sigma^2}{n})$-distributed variable.*

The importance of the normal distribution in probability and statistics becomes comprehensible if one considers the central limit theorem (Theorem 2.7). Especially, the central limit theorem proves that the mean value of any stochastic variable that is not normally distributed converges in distribution to the normal distribution.

Theorem 2.7. *Let $X_1, X_2, \ldots$ be a statistically independent sequence of stochastic variables, all of them with expectation μ and variance σ^2. We assume that $\sigma \neq 0$. Under these conditions the variable*

$$\frac{1}{n}S_n := \frac{X_1 + X_2 + \cdots + X_n}{n}$$

is approximately $N(\mu, \sigma^2/n)$-distributed.

2.1.2 Hypothesis Tests

Side channel analysis applies hypothesis testing as a decision strategy for key guessing. Because of this, the background in statistical hypothesis testing is provided. The presentation follows [113, 119]. A book dedicated to hypothesis testing is [78].

A *statistical hypothesis* H is a conjecture about the probability distribution of a population. A hypothesis H is said to be *simple* if the distribution of the population is completely specified by H. Otherwise, H is called a *composite* hypothesis. In a simple setting, one deals with the null hypothesis H_0 and the alternative hypothesis H_1.

Definition 2.6. A hypothesis test is understood to be an ordered sequence $(x_1, \ldots, x_n; H_0, H_1; G)$ where $x_1, \ldots, x_n$ is a sample, H_0 and H_1 are hypotheses concerning the probability distribution of the population, and $G \subset \mathbb{R}^n$ a *critical region* in the hypothesis test.

If $(x_1, \ldots, x_n)$ is an element of G, H_1 is accepted, otherwise the decision is made in favor of H_0. Note that this decision procedure may lead to wrong results. If one decides to accept H_1, whereas in reality H_0 is true, one has committed a *type I error*. The probability of committing a type I error will be denoted by α. The probability α is also known as the *level of significance* of the hypothesis test. Acceptance of H_0 whereas H_1 is true is called a *type II error*. The probability of committing a type II error will be denoted by β.

Generally, for composite hypotheses, one cannot provide the numerical value of β. However, by considering a family of probability densities f(•,θ) with $\theta \in \Theta$, β is seen as a function of a parameter θ. For example, in case of a normal distribution, one may define $\theta = (\mu, \sigma)$. From now on it is assumed that

$$H_0 : \theta \in \Theta_0 \text{ and } H_1 : \theta \in \Theta_1$$

with $\Theta_0 \cap \Theta_1 = \varnothing$ and $\Theta_0 \cup \Theta_1 = \Theta$.

The common approach is to define the level of significance first, e.g., $\alpha = 0.05$. For a given value of α, critical regions G of different efficiency can be chosen. One says that $1 - \beta(\theta)$ is the power function on Θ_1. Of course, one is interested in choosing a critical region G generating maximal power. The Neyman-Pearson lemma answer the question how to find an optimal critical region.

Theorem 2.8. *Suppose the hypothesis $H_0 : \theta = \theta_0$ is tested against $H_1 : \theta = \theta_1$. Let G be the critical region defined by*

$$G = G(\delta) := \left\{ \vec{x} \in \mathbb{R}^n : \frac{L_0(\vec{x})}{L_1(\vec{x})} \leq \delta \right\} (\delta > 0)$$

and let α be the size of it. Among all critical regions of size α, this critical region G is for the test the one generating maximal power.

In Theorem 2.8, L_0 is the likelihood function for $\theta = \theta_0$ and L_1 is the likelihood function for $\theta = \theta_1$. Outcomes $\vec{x}$ yielding a high value of L_θ are said to be *likely*. If $x_1, x_2, \ldots, x_n$ is a sample from a population with probability density $f(\bullet, \theta)$ then

$$L_\theta(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} f(x_i; \theta)$$

The value $\hat{\theta}$ that leads to the maximum of L_θ is said to be the *maximum likelihood estimation of θ*. Informally speaking, Theorem 2.8 says that the critical region G where L_0/L_1, also referred to as *likelihood ratio*, produces small values is the one generating maximum power.

In practical tests one distinguishes between

- *one-sided alternatives*, e.g., one tests $H_0 : \mu = \mu_0$ against $H_1 : \mu > \mu_0$ (or alternatively $H_1 : \mu < \mu_0$), and

- *two-sided alternatives*, e.g., one tests $H_0 : \mu = \mu_0$ against $H_1 : \mu \neq \mu_0$.

For two-sided alternatives, there are two critical regions that sum up to size α.

Hypothesis tests can be classified in tests concerning normally distributed populations and so-called 'non-parametric methods'. For side channel cryptanalysis one often assumes that observables stem from a normal distribution so that tests for normal distributions are by far the most common ones.

T-Test for Significantly Different Means

The T-Test assumes $N(\mu, \sigma^2)$-distributed populations. It deals with hypotheses concerning the mean value μ. The parameter σ^2 is either supposed to be known beforehand or – and this is the more important case for practical purposes – is approximated by computing the empirical variance.

For side channel analysis, Theorem 2.9 considers the most relevant case. Here, one asks whether the mean values of two normally distributed populations are different or not.

Theorem 2.9. *Two statistically independent samples $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$ are drawn from a $N(\mu_X, \sigma^2)$-distributed and a $N(\mu_Y, \sigma^2)$-distributed population, respectively, where μ_X, μ_Y and σ are unknown. If, under a prescribed value of Δ, we wish to test the hypothesis*

$$H_0 : \mu_Y - \mu_X = \Delta \;\; \textit{against} \;\; H_1 : \mu_Y - \mu_X \neq \Delta$$

then the likelihood ratio takes the form

$$\Lambda(x_1, \ldots, x_m, y_1, \ldots, y_n) = \left[1 + \frac{1}{m+n+2} \left(\frac{\overline{y} - \overline{x} - \Delta}{S_p \sqrt{\frac{1}{m} + \frac{1}{n}}} \right)^2 \right]^{-\frac{m+n}{2}}$$

where

$$S_p^2 = \frac{(m-1)\, S_x^2 + (n-1)\, S_y^2}{m+n-2}$$

is the pooled empirical variance. Critical regions based on this ratio are given by

$$G = \left\{ (x_1, \ldots, x_m, y_1, \ldots, y_n) \in \mathbb{R}^{m+n} : \frac{|\overline{y} - \overline{x} - \Delta|}{S_p \sqrt{\frac{1}{m} + \frac{1}{n}}} \geq c \right\}$$

By setting $\Delta = 0$, one obtains the special case for deciding whether two samples from normal populations have the same mean or not. The test statistic

$$T = \left(\frac{\overline{y} - \overline{x} - \Delta}{S_p \sqrt{\frac{1}{m} + \frac{1}{n}}} \right) \tag{2.7}$$

is t-distributed with $m+n-2$ degrees of freedom. The parameter c is adjusted so that the critical region G is of size α. Table 2.1 summarizes the characteristics of one-sided and two-sided alternatives. The decision is made based on the t-distribution that depends on the number of degrees of freedom and the value of α for one-sided alternatives, respectively the value of $\frac{\alpha}{2}$ for two-sided alternatives. The t-distribution can be found in many standard statistic books (e.g., [119, 32, 113]).

Table 2.1: T-Test of two populations $N(\mu_X, \sigma^2)$ and $N(\mu_Y, \sigma^2)$ with unknown σ. The abbreviation $\Delta_\mu := \mu_Y - \mu_X$ is used below.

H_0	H_1	Statistics	Acceptance of H_1
$\Delta_\mu \leq \Delta$	$\Delta_\mu > \Delta$	$\left(\frac{\overline{y}-\overline{x}-\Delta}{S_p\sqrt{\frac{1}{m}+\frac{1}{n}}} \right)$	$(t_{n+m-2;1-\alpha}; \infty)$
$\Delta_\mu \geq \Delta$	$\Delta_\mu < \Delta$	$\left(\frac{\overline{y}-\overline{x}-\Delta}{S_p\sqrt{\frac{1}{m}+\frac{1}{n}}} \right)$	$(-\infty; -t_{n+m-2;1-\alpha})$
$\Delta_\mu = \Delta$	$\Delta_\mu \neq \Delta$	$\left\vert \left(\frac{\overline{y}-\overline{x}-\Delta}{S_p\sqrt{\frac{1}{m}+\frac{1}{n}}} \right) \right\vert$	$(t_{n+m-2;1-\frac{\alpha}{2}}; \infty)$

For side channel analysis, one typically encounters sample sizes of a few tens up to a few hundred thousands in each population. For a degree of freedom of more than 200, the t-distribution can be approximately substituted by the normal distribution.

In literature, other variants of the T-Test are described [113, 119]. Especially, one distinguishes the variants

- one population $N(\mu, \sigma_0^2)$ where σ_0 is known,
- one population $N(\mu, \sigma^2)$ where σ is unknown,
- two populations $N(\mu_X, \sigma_X^2)$ and $N(\mu_Y, \sigma_Y^2)$ where σ_X and σ_Y are known,
- two populations $N(\mu_X, \sigma^2)$ and $N(\mu_Y, \sigma^2)$ where σ is not known, and
- two populations $N(\mu_X, \sigma_X^2)$ and $N(\mu_Y, \sigma_Y^2)$ where $\sigma_X \neq \sigma_Y$.

These T-Test variants can be found, e.g., in [113, 119] and are not further discussed here.

F-Test for Significantly Different Variances

The F-Test also assumes $N(\mu, \sigma^2)$-distributed populations. It investigates significant differences of the parameter σ. Though less common in side channel analysis, a test for significantly different variances might be useful if differences in means cannot be proven. Again, the most relevant case for side channel analysis is considered in Theorem 2.10.

Theorem 2.10. *We draw two statistically independent samples $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$ from a $N(\mu_X, \sigma_X^2)$-distributed and a $N(\mu_Y, \sigma_Y^2)$-distributed population, respectively. If we wish to test the hypothesis*

$$H_0 : \sigma_X = \sigma_Y \quad \textit{against} \quad H_1 : \sigma_X \neq \sigma_Y$$

then the likelihood ratio is given by

$$\Lambda(x_1, \ldots, x_m, y_1, \ldots, y_n) = \frac{\left(\frac{m-1}{m}\right)^{\frac{m}{2}} \left(\frac{n-1}{n}\right)^{\frac{n}{2}} \left(\frac{S_y^2}{S_x^2}\right)^{\frac{n}{2}}}{\left(\frac{m-1}{m+n} + \frac{n-1}{m+n}\frac{S_y^2}{S_x^2}\right)^{\frac{m+n}{2}}}$$

and the critical region based on this ratio is of the form: $G = G_1 \bigcup G_2$, where

$$G_1 = \left\{ (x_1, \ldots, x_m, y_1, \ldots, y_n) \in \mathbb{R}^{m+n} : \frac{S_y^2}{S_x^2} \leq c_1 \right\}$$

Table 2.2: F-Test of two populations $N(\mu_x, \sigma_x^2)$ and $N(\mu_y, \sigma_y^2)$.

H_0	H_1	Statistics	Acceptance of H_1
$\sigma_x^2 \leq \sigma_y^2$	$\sigma_x^2 > \sigma_y^2$	$\frac{S_y^2}{S_x^2}$	$(F_{n-1;m-1;1-\alpha}; \infty)$
$\sigma_x^2 \geq \sigma_y^2$	$\sigma_x^2 < \sigma_y^2$	$\frac{S_y^2}{S_x^2}$	$[0; F_{n-1;m-1;\alpha})$
$\sigma_x^2 = \sigma_y^2$	$\sigma_x^2 \neq \sigma_y^2$	$\lvert\frac{S_y^2}{S_x^2}\rvert$	$[0; F_{n-1;m-1;\frac{\alpha}{2}}) \cup$ $(F_{n-1;m-1;1-\frac{\alpha}{2}}; \infty)$

and

$$G_2 = \left\{ (x_1, \ldots, x_m, y_1, \ldots, y_n) \in \mathbb{R}^{m+n} : \frac{S_y^2}{S_x^2} \geq c_2 \right\} .$$

The outcome of the test statistic

$$T = \frac{S_Y^2}{S_X^2}$$

is decisive. Under H_0 this statistic is F-distributed with $n-1$ degrees of freedom in the numerator and $m-1$ degrees in the denominator. The F-distribution can be found in many standard statistic books (e.g., [119, 32, 113]). The constants c_1 and c_2 of Theorem 2.10 are adjusted so that both G_1 and G_2 are of size $\frac{\alpha}{2}$.

Non-parametric Tests

Non-parametric gain importance if the populations are not necessarily normally distributed. One example is Wilcoxon's rank-sum test that builds rank numbers in the joint set of $\{x_1, \ldots, x_m, y_1, \ldots, y_n\}$. Let $r(x_i)$ and $r(y_i)$ denote the rank number of x_i and y_i in the set above, respectively. The test statistics uses the rank sums

$$T_X := \sum_{i=1}^{m} r(x_i) \text{ and } T_Y := \sum_{i=1}^{n} r(y_i) .$$

If T_X (or T_Y) has a very small or large outcome then one observes a tendency that smaller or larger outcomes emanate from one population and may reject H_0. For further reading it is referred to [113].

2.1.3 Vectorial Statistics

The introductory part in Section 2.1.1 provided statistics for scalar measurements, i.e., the outcomes of a stochastic variable are elements of $\mathbb{R}$. In side channel analysis one is, however, able to observe multiple instants, even by using multiple measurement set-ups and in fault channel analysis one may introduce multiple fault injections. In vectorial statistics, the outcomes of a stochastic variable are vectors, i.e., outcomes are elements of $\mathbb{R}^p$.

In this section, vectorial statistics is reflected, as far as it is needed for a thorough understanding of this thesis. In detail, this section consists of

- Introduction to Linear Algebra,
- Multivariate Gaussian Distribution,
- Normal Correlation Analysis, and
- Multiple Linear Regression.

Introduction to Linear Algebra

An element $\vec{x} \in \mathbb{R}^p$ is said to be a *vector*. Its p elements $x_1, \ldots, x_p \in \mathbb{R}$ are called *scalars*. The *vectorial addition*

$$\vec{x} + \vec{y} := (x_1 + y_1, \ldots, x_p + y_p)$$

of two vectors $\vec{x}$ and $\vec{y}$ in combination with a *scalar multiplication*

$$\alpha\vec{x} := (\alpha x_1, \ldots, \alpha x_p)$$

with $\alpha \in \mathbb{R}$ provides the so-called *linear structure on* $\mathbb{R}^p$. The zero vector $\vec{0}$ is $\vec{0} := (0, \ldots, 0)$. It can be shown that vectors $v_1, \ldots, v_p$ form a *basis* in $\mathbb{R}^p$ if and only if every vector $\vec{x}$ can unambiguously be decomposed as

$$\vec{x} = \alpha_1\vec{v_1} + \cdots + \alpha_p\vec{v_p}.$$

A *linear subspace* of $\mathbb{R}^p$ is a subset $\mathbb{M} \subset \mathbb{R}^p$ meeting the following three properties:

1. $\vec{0} \in \mathbb{M}$,

2. If $\vec{x}, \vec{y} \in \mathbb{M}$, then $\vec{x} + \vec{y} \in \mathbb{M}$,

3. If $\alpha \in \mathbb{R}$ and $\vec{x} \in \mathbb{M}$ then $\alpha\vec{x} \in \mathbb{M}$.

One defines the *inner product* (or *scalar product*) of two vectors $\vec{x}$ and $\vec{y}$ as

$$\langle \vec{x}, \vec{y} \rangle := x_1 y_1 + \cdots + x_p y_p \ .$$

The length of a vector $\vec{x}$ is defined to be $\|\vec{x}\| := \sqrt{\langle \vec{x}, \vec{x} \rangle}$. Accordingly, the distance between two vectors $\vec{x}$ and $\vec{y}$ is $\|\vec{x} - \vec{y}\|$.

A $q \times p$ *matrix* $\mathbf{M}$ with scalars $M_{11}, \ldots, M_{qp}$ is defined by:

$$\mathbf{M} = \begin{pmatrix} M_{11} & M_{12} & . & . & . & . & M_{1p} \\ M_{21} & M_{22} & . & . & . & . & M_{2p} \\ . & . & & & & & . \\ . & . & & & & & . \\ . & . & & & & & . \\ M_{q1} & M_{q2} & . & . & . & . & M_{qp} \end{pmatrix}$$

The *transposed* of a $q \times p$ matrix $\mathbf{M}$ is given by $(\mathbf{M}^T)_{ij} = \mathbf{M}_{ji}$, i.e., by:

$$\mathbf{M}^T = \begin{pmatrix} M_{11} & M_{21} & . & . & . & . & M_{q1} \\ M_{12} & M_{22} & . & . & . & . & M_{q2} \\ . & . & & & & & . \\ . & . & & & & & . \\ . & . & & & & & . \\ M_{1p} & M_{2p} & . & . & . & . & M_{qp} \end{pmatrix}$$

A $p \times p$ matrix $\mathbf{M}$ is said to be *invertible* (or *regular*) if there exists a $p \times p$ matrix $\mathbf{B}$ such that

$$\mathbf{MB} = \mathbf{BM} = \mathbf{1}$$

with $\mathbf{1}$ being the identity map on $\mathbb{R}^p$. If so, $\mathbf{B}$ is said to be the *inverse* of $\mathbf{M}$, referred to as $\mathbf{M}^{-1}$.

One denotes $\det(\mathbf{M})$ as the determinant of a $p \times p$ matrix $\mathbf{M}$:

$$\det(\mathbf{M}) = \begin{vmatrix} M_{11} & M_{12} & . & . & . & . & M_{1p} \\ M_{21} & M_{22} & . & . & . & . & M_{2p} \\ . & . & & & & & . \\ . & . & & & & & . \\ . & . & & & & & . \\ M_{p1} & M_{p2} & . & . & . & . & M_{pp} \end{vmatrix}$$

i.e., defined by

$$\det(\mathbf{M}) := \sum_{\pi} (-1)^{j(\pi)} M_{1i_1} M_{2i_2} \dots M_{pi_p}$$

where the sum runs over all permutations π of the numbers $1, 2, \dots, p$, in such a way that each product includes exactly one element of each column and row of the matrix $\mathbf{M}$. The term $j(\pi)$ denotes the number of inversions of the permutation

$$\pi = \begin{pmatrix} 1 & 2 & . & . & . & p \\ i_1 & i_2 & . & . & . & i_p \end{pmatrix}.$$

Vectorial Normal Distribution

In vectorial statistics, the expectation value $\mathbb{E}(\vec{X})$ is understood to be the vector $(\mathbb{E}(X_1), \dots, \mathbb{E}(X_p))$. It is assumed that all components of the stochastic vector $\vec{X} = (X_1, \dots, X_p)$ are of finite variance. The covariance matrix $\mathbf{C}(\vec{X})$ is then defined as

$$\mathbf{C}(\vec{X}) = \begin{pmatrix} \operatorname{cov}(X_1, X_1) & \operatorname{cov}(X_1, X_2) & . & . & . & . & \operatorname{cov}(X_1, X_p) \\ \operatorname{cov}(X_2, X_1) & \operatorname{cov}(X_2, X_2) & . & . & . & . & \operatorname{cov}(X_2, X_p) \\ . & . & & & & & . \\ . & . & & & & & . \\ . & . & & & & & . \\ \operatorname{cov}(X_p, X_1) & \operatorname{cov}(X_p, X_2) & . & . & . & . & \operatorname{cov}(X_p, X_p) \end{pmatrix} \tag{2.8}$$

where the matrix elements contain the covariance as given in Definition 2.2.

In Section 2.1.1 *scalar* or *univariate* samples are introduced. The generalization for vectorial samples is as follows: A *vectorial sample* (or *multivariate sample*) of size n is understood to be a statistically independent set of stochastic vectors $\vec{x_1}, \dots, \vec{x_n}$ all enjoying a common probability distribution.

It can be shown that a normally distributed stochastic p-dimensional vector with expectation vector $\vec{\mu}$ and covariance matrix $\mathbf{C}$ has a probability density given by

$$f(\vec{x}) = \frac{1}{(2\pi)^{\frac{p}{2}} \sqrt{\det(\mathbf{C})}} \exp\left[-\frac{1}{2} \langle \mathbf{C}^{-1} (\vec{x} - \vec{\mu}), (\vec{x} - \vec{\mu}) \rangle \right]. \tag{2.9}$$

Normal Correlation Analysis

If one knows that the 2-dimensional vector (X, Y) enjoys a normal distribution, normal correlation analysis saves unnecessary loss of efficiency if compared to non-parametric tests [113, 71].

One can prove that X and Y are statistically independent if and only if they are uncorrelated, which constitutes one of the special cases discussed in Section 2.1.1. Statistical independence of X and Y implies that the correlation coefficient ρ is zero. It can be shown that the maximum likelihood estimator of ρ is indeed Pearson's product-moment correlation coefficient

$$R_p = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n} (x_i - \overline{x})^2 \sum_{i=1}^{n} (y_i - \overline{y})^2} \tag{2.10}$$

of a sample $(x_1, y_1), \ldots, (x_n, y_n)$.

In side channel analysis one aims to give an answer whether or not an observed value for R_p is significantly different from zero. For this, one has to check the hypothesis $H_0 : R_p = 0$ versus $H_1 : R_p \neq 0$. For small values of n one can use the results of Proposition 2.11 to carry out an hypothesis test based on the t-distribution.

Proposition 2.11. *Suppose $(x_1, y_1), \ldots, (x_n, y_n)$ is a sample of size n from a 2-dimensional normally distributed population with $\rho = 0$. Then the variable*

$$\frac{\sqrt{n-2}R_p}{\sqrt{1 - R_p^2}}$$

is t-distributed with n-2 degrees of freedom.

For high values of n, one can use the fact that the statistic Z of Proposition 2.12 is asymptotically normally distributed. This statistic Z is often referred to as Fisher's Z.

Proposition 2.12. *Suppose $(x_1, y_1), \ldots, (x_n, y_n)$ is a sample of size n from a 2-dimensional normally distributed population with correlation coefficient ρ. Then for large n the statistic*

$$Z := \frac{1}{2} \ln \frac{1 + R_p}{1 - R_p}$$

is approximately $N(\mu, \sigma^2)$-distributed with

$$\mu = \frac{1}{2}\ln\frac{1+\rho}{1-\rho} \text{ and } \sigma^2 = \frac{1}{n-3}$$

Multiple Regression Analysis

If one assumes a linear relation between two variables x and y, i.e.,

$$y = a + bx$$

and wants to make an estimation on the constants a and b this can be solved using the method of least squares, recalled in Proposition 2.13.

Proposition 2.13. *Suppose that, concerning the quantities x and y, there is a theoretical relationship $y = \alpha + \beta x$. Then the least squares estimations of α and β, based on n measurements $(x_1,y_1),\ldots,(x_n,y_n)$, are given by*

$$\hat{\beta} = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})(x_i - \overline{x})} \text{ and } \hat{\alpha} = \overline{y} - \hat{\beta}\overline{x}.$$

In multiple regression analysis one deals with the vectorial generalization of this problem.

$$y = a + \langle \vec{b}, \vec{x} \rangle$$

with $a \in \mathbb{R}$, $\vec{b} \in \mathbb{R}^m$, and $\vec{x} = (x_1, \ldots, x_m) \in \mathbb{R}^m$. The x_i are also seen as the *controlled variables* whereas y is the outcome of the stochastic variable Y as response to the given stimuli $(x_1, \ldots, x_m)$. For analysis, one has n pairs of measurements $(x_1, \ldots, x_m, y)$. One now wants to find the least square estimates $\alpha \in \mathbb{R}$ and $\vec{\beta} \in \mathbb{R}^m$ in a given linear subspace $\mathbb{M} \subset \mathbb{R}^n$. From now on, the vectors $\vec{x_1} = (x_{11}, x_{21}, \ldots, x_{n1})$ up to $\vec{x_m} = (x_{1m}, x_{2m}, \ldots, x_{nm})$ are introduced for each controlled variable x_i $(1 \leq i \leq m)$ given n measurements. $\mathbb{M}$ is spanned by the $m+1$ base vectors $\{\vec{e}, \vec{x_1}, \ldots, \vec{x_m}\}$ whereby $\vec{x_i} \in \mathbb{R}^n$ for all $i \in \{1, \ldots, m\}$ and $\vec{e} = (1, \ldots, 1) \in \mathbb{R}^n$.

The $n \times (m+1)$ matrix

$$\mathbf{M} = \begin{pmatrix} 1 & x_{11} & x_{12} & . & . & . & x_{1m} \\ 1 & x_{21} & x_{22} & . & . & . & x_{2m} \\ . & . & & & & & . \\ . & . & & & & & . \\ . & . & & & & & . \\ 1 & x_{n1} & x_{n2} & . & . & . & x_{nm} \end{pmatrix}$$

is said to be the *design matrix* of the fitting problem. Note that it is $n > m$. Correspondingly, one defines a vector $\vec{y} := (y_1, y_2, \ldots, y_n)$. In matrix notation one obtains

$$\begin{pmatrix} \hat{\alpha} \\ \hat{\beta}_1 \\ . \\ . \\ . \\ \hat{\beta_m} \end{pmatrix} = \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T \begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{pmatrix}$$

Proposition 2.14 summarizes the solution for general linear least squares.

Proposition 2.14. *The least square estimators* $\hat{\alpha}$, $\hat{\vec{\beta}}$ *of* α, $\vec{\beta}$ *are given by*

$$\left(\hat{\alpha}, \hat{\vec{\beta}}\right) = \mathbf{TY} = \mathbf{T}\left(Y_1, \ldots, Y_n\right)$$

where $\mathbf{T} : \mathbb{R}^n \to \mathbb{R}^{m+1}$ *is the linear operator belonging to the matrix*

$$[\mathbf{T}] = \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T.$$

In practice, one has to solve a matrix inversion which can be done using standard algorithms, e.g., the Gauss-Jordan elimination [115].

Note that this method of general linear least squares can be applied to any $m+1$ basis functions $g_i : \mathbb{R}^u \to \mathbb{R}$ with $0 \leq i \leq m$. For $\vec{x} \in \mathbb{R}^u$ the model is then

$$y(\vec{x}) = \sum_{k=0}^{m} b_k g_k(\vec{x}) \ .$$

2.2 Entropy and Information

The foundations of information theory in communication systems were laid down by Shannon in [135]. In the context of physical cryptanalysis one is mainly interested in a measure how much 'information' is generated by a cryptographic device or – stated differently – how much 'uncertainty' remains in a cryptographic device. Information leakage, e.g., by physical attacks, may lead to a loss of 'information', respectively 'uncertainty' in a cryptographic device. In cryptography, information is usually stored in a digitized form, e.g., an n-bit secret key value. Because of this, one focuses on discrete probability distributions and follows the lines of [135, 92].

Let X be a stochastic variable which takes on a finite set of values $x_1, x_2, \ldots, x_n$ with probability $\mathbb{P}(X = x_i) = p_i$.

Definition 2.7. The entropy of X is defined to be

$$H(X) = -\sum_{i=1}^{n} p_i \lg p_i.$$

By convention, $p_i \lg p_i = 0$ if $p_i = 0$.
It can be seen that

(i) $0 \leq H \leq \lg n$,

(ii) $H = 0$ if and only if there is no uncertainty about the outcomes, i.e., $p_i = 1$ for one $i \in \{1, \ldots, n\}$, and

(iii) $H = \lg n$ if and only if $p_i = \frac{1}{n}$ for each $i \in \{1, \ldots, n\}$, i.e., all outcomes are equally likely.

Let $H(X)$ and $H(Y)$ denote the entropy of the stochastic variable X and Y, respectively.

Definition 2.8. The joint entropy of X and Y is defined as

$$H(X,Y) = -\sum_{x,y} \mathbb{P}\left(X = x, Y = y\right) \lg \mathbb{P}\left(X = x, Y = y\right).$$

If follows that $H(X,Y) \leq H(X) + H(Y)$. Equality holds if and only if X and Y are statistically independent.

Definition 2.9. The conditional entropy of X given Y is

$$H(X|Y) = -\sum_{y} \mathbb{P}(Y = y)\ H(X|Y = y)$$

with

$$H(X|Y = y) = -\sum_{x} \mathbb{P}(X = x|Y = y)\ \lg \mathbb{P}(X = x|Y = y)$$

$H(X|Y)$ provides a measure about the uncertainty of X remaining after Y has been observed. It is used to measure the entropy loss $H(X) - H(X|Y)$ as result of applying physical cryptanalysis. $H(X) - H(X|Y)$ is also known as the *mutual information* of X and Y that is the amount of information that Y reveals about X.

2.3 CMOS VLSI Technology

This section provides a background on CMOS VLSI design as far as it is relevant for the understanding of this technology in this thesis. The presentation follows the lines of Weste and Eshraghian [145] and uses some entries of [2].

Complementary Metal Oxide Silicon (CMOS) technology has become the leading technology for manufacturing integrated circuits (also referred to as ICs or chips) in the last decades. CMOS design allows Very Large Scale Integration (VLSI), i.e., chips which consist of more than hundreds of thousands transistors. Continuous minimization has yielded submicron structures and upcoming CMOS technology will be based on gate widths even smaller than 90 nanometers.

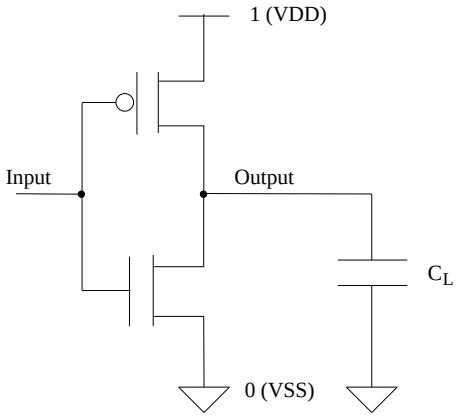


Figure 2.1: CMOS inverter

A CMOS gate consists of a pull-down network to connect the output to logical '0' (V_{SS}) and a pull-up network to connect the output to logical '1' (V_{DD}). The CMOS inverter depicted in Figure 2.1 is built of a p-channel transistor that is almost perfect in passing a logical '1' if the gate input is zero and an n-channel transistor that gives a nearly perfect logical '0' if the gate input is logical '1'. During a transition of the input signal, e.g., from '0' to '1' or vice versa, both n- and p-transistors are conductive for a short period of time. This causes a

dynamic power dissipation. However, in steady-state, it is characteristic for CMOS technology that static power dissipation is very low.

The total power dissipation P of a circuit is the sum of the static power dissipation P_s, the dynamic power dissipation due to the charging of output capacitances P_d, and the dynamic short-circuit power dissipation P_{sc}.

$$P = P_s + P_d + P_{sc}.$$

Among them, P_d is usually the dominant term. It is given as

$$P_d = C_L V_{DD}^2 f_p$$

wherein C_L is the load capacitance, V_{DD} the supply voltage, and f_p the frequency.

CMOS VLSI circuits are manufactured as a wafer or disk out of silicon that is less than 1 mm thick. In IC production, a typical CMOS process involves deposition, etching, patterning, and ion implantation. Deposition is any process that transfers material onto the wafer, while etching is any process that removes material from the wafer. Patterning allows to shape or alter the existing shape of the surface and is also known as lithography. Typical mask materials are photoresist, polysilicon, silicon dioxide, and silicon nitride. These materials can be selectively removed while preserving a barrier for a process at other locations. Ion implantation leads to local modifications of electrical properties and forms regions of varying doping concentrations.

Modern ICs have three or more metal layers. The third dimension is both used for routing purposes as well as for the construction of gates. Metal layers are connected by vias. On top of the last metal layer a passivation layer is built that covers the chip except for openings at test pads needed for device testing. Before delivery the chip is bonded and finally encapsulated in package material.

Chapter 3

Related Work

This chapter aims to provide an introduction and summary of related research. It is organized in the four main areas

- Physical Security (Section 3.1),
- Physical Cryptanalysis (Section 3.2),
- Side Channel Cryptanalysis (Section 3.3), and
- Fault Channel Cryptanalysis (Section 3.4).

While the section on physical security gives an introduction to the area of implementation based security and includes the common terminology on physical security for cryptographic modules from the ISO and FIPS standards, the further sections on physical cryptanalysis, side channel cryptanalysis, and fault channel cryptanalysis provide research results related to this thesis.

3.1 Physical Security

Cryptographic modules are designed to provide security services in computer science, for instance, integrity and confidentiality of application data. For the corresponding security mechanisms cryptographic keys are needed which have to be protected themselves against unauthorized disclosure and modification.

The concept of the *cryptographic boundary* is a term used in the FIPS-140 security requirements [105] to define a clear boundary between the internals of the cryptographic module and its environment. The relevant security parts to be protected are all encapsulated by the cryptographic boundary which includes, e.g., the processing hardware, data and program memories as well as other critical components such as a physical random number generator (RNG) or a real time clock (RTC) (see Fig. 3.1). Typically, a cryptographic module possesses external interfaces to the cryptographic boundary, e.g, for data communication and power supply. If existing these lines are untrusted as they can be accessed externally.

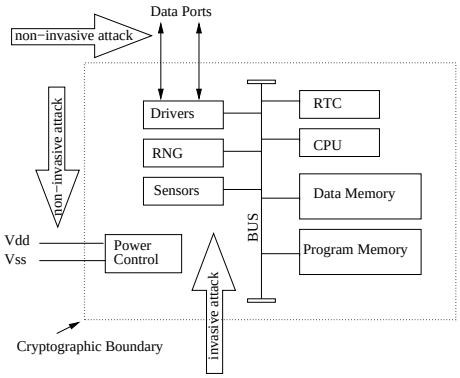


Figure 3.1: Components of a computer system that are enclosed in a cryptographic boundary.

Operational Environment

The operational environment of a cryptographic module may range from the security server located at the headquarter of a banking association to security tokens which are handed over to the end user. While it can be assumed that central security servers are protected by effective *environmental security measures* (e.g., guards, alarm systems, and special organizational measures), these measures cannot be enforced in the case of devices in the possession of untrusted end users. If security tokens cannot be protected by the environment, they have to be constructed to protect themselves.

Between the two polarized cases of *protected* and *non-protected environments* there is a wide range of *partially protected environments*. An example for a partially protected environment are random checks of the cryptographic module whether there have been tampering attempts or not.

3.1.1 Attack Scenarios

An implementation attack is said to be *invasive* if it breaches the cryptographic boundary. Otherwise, it is *non-invasive*. An implementation attack is called *active* if the adversary actively interferes with the cryptographic implementation, e.g., operates the implementation under ex-

treme physical conditions. If active interactions with the cryptographic implementation are completely absent and the cryptographic module is just operated in its intended environment, the attack is said to be *passive*.

In ISO 13491-1 [65] five attack scenarios are defined which indicate the main areas of concern for security modules.

- *Penetration* is an active, invasive attack on the cryptographic module, i.e. it includes breaking into the cryptographic boundary of the module. The aim is to intercept data at the internal communication lines or to read out the memory in order to determine the secret keys stored inside the security module.

- *Monitoring* is a passive, non-invasive attack which leaves the cryptographic boundary intact. This class of attack makes use of the inherent leakage of the cryptographic module, e.g., by measuring the electromagnetic emanation. Capturing electromagnetic emanation of electronic devices such as video displays, microchips or printers (the US military calls this TEMPEST [91, 75, 8]) and side channel analysis on cryptographic modules are prominent monitoring attacks. For a detailed description on side channel cryptanalysis it is referred to Section 3.3.

- *Manipulation* is an active, non-invasive attack which leaves the cryptographic boundary intact. The attack aims to obtain a service in an unintended manner at the logical interface [65]. In addition to [65], manipulating attacks may also include anomalous environmental conditions. For instance, the cryptographic module may be operated under extreme operating conditions, e.g., with short-time glitches in the power supply or at an extreme temperature. Many early fault channel based attacks belong to this attack scenario (cf. Section 3.4).

- *Modification* is an active, invasive attack. This usually includes penetrating the cryptographic boundary. Unlike penetration attacks, the aim is to modify internal components or the internal memories used. Advanced fault channel based attacks aim at an internal modification of the cryptographic device (cf. Section 3.4).

- *Substitution* includes the removal of the cryptographic module, which is then substituted by an emulating device with a modified implementation of security functions. The cryptographic boundary is not of primary interest in this attack. Note that the removed module can be used for a comprehensive analysis of the internal construction.

3.1.2 Security Objectives

There are two different objectives in physical security.

- The first security objective aims to prevent the disclosure and modification of internal data (e.g., cryptographic keys and application data). For its realization, either *tamper-resistant* or *tamper-responsive* measures are implemented. According to [65] the terms are defined as follows. A *tamper-responsive characteristic* provides an active response to the detection of an attack, thereby preventing its success. Tamper-responsive measures may lead to an immediate zeroization of secret cryptographic keys once an attack is detected. A *tamper-resistant characteristic* provides passive physical protection against an attack. Tamper resistance implies that the cryptographic module is able to avert all attacks even without any active reaction.

- The second security objective aims to provide assurance whether or not a cryptographic module has been tampered with. For this, *tamper-evident* characteristics are needed. According to [65] a *tamper-evident characteristic* provides evidence that an attack has been attempted. Note that tamper evidence cannot prevent breaking into the cryptographic boundary nor the disclosure of internal data of the cryptographic module. Moreover, the use of a tamper-evident scheme requires a control authority that regularly and carefully inspects the cryptographic module.

3.1.3 Security Requirements

Among other sets of requirements, such as, e.g., smart card IC protection profiles used by Common Criteria evaluations [63] and [133], the FIPS 140 security requirements give insights into the implementation of

secure computer systems for the use in unprotected areas. FIPS (Federal Information Processing Standards) are developed under the National Institute of Standards and Technology (NIST), for use by US federal government departments.

For the evaluation of cryptographic modules the NIST published the standard FIPS 140 [104] in 1994. It contains security requirements for cryptographic modules. In 2001, FIPS 140-2 [105] superseded the previous standard FIPS 140-1.

The FIPS 140-2 standard defines four security levels ranging from a low security level to the definition of highly resistant cryptographic modules. The evaluation aspects cover eleven requirement areas: (i) Cryptographic Module Specification, (ii) Cryptographic Module Ports and Interfaces, (iii) Roles, Services, and Authentication, (iv) Finite State Model, (v) Physical Security, (vi) Operational Environment, (vii) Cryptographic Key Management, (vii) Electromagnetic Interference / Electromagnetic Compatibility, (ix) Self-Tests, (x) Design Assurance, and (xi) Mitigation of Other Attacks.

The requirements in each area are detailed and even go down to the implementation level. Physical security is one aspect among them. The majority of areas, however, deals with logical functions to be implemented. Other aspects cover the quality of the design documentation. 'Mitigation of Other Attacks' is an optional area without concrete test procedures: This area includes recent implementation attacks such as 'Power Analysis', 'Timing Analysis', 'Fault Induction' and 'TEMPEST'.

Regarding 'Physical Security' FIPS 140-2 distinguishes the embodiments

- Single-chip cryptographic module,
- Multiple-chip embedded cryptographic module, and
- Multiple-chip standalone cryptographic module.

The requirements on physical security increase from level 1 (no special protections) to level 4 (control of environmental temperature and voltage, single chip: 'hard opaque removal-resistant coating', multiple-chip: 'tamper detection envelope with tamper response and zeroization circuitry'). Level 2 and level 3 provide tamper-evident measures. Level 3 includes an automatic zeroization when the privileged maintenance access interface is entered.

Sections 3.1.4 to 3.1.6 detail the requirements for tamper evidence, tamper response, and tamper resistance. Further, a short summary on relevant issues and possible efficient technical solutions is added for each of them. These summaries should not be seen as a 'final word' on technical realizations, but instead as some promising directions for building physically secured cryptographic modules.

3.1.4 Tamper Evidence

Tamper evidence requires a trusted overseer (or control personnel) who carefully inspects the cryptographic device whether tampering attempts have been occurred. One may require that inspections are conducted in a random, non-predictable way. Between inspections, the cryptographic module may reside in a public area under surveillance or in a non-protected environment. In the first case it may be assumed that a potential attacker can be identified afterwards using, e.g., video streaming data. In the second scenario the operator is responsible for compliance with the legal regulations. Randomness of inspections is important to avoid a regular replacement with an emulating device between the controls. Additionally, one requires that the overseer correctly identifies indications towards tamper attempts and carries out inspections with care in detail, according to the rules.

Tamper evidence cannot prevent the disclosure of internal confidential information as well as cryptographic keys. Applications that rely on the secrecy of cryptographic keys should not rely on tamper-evident measures if the cryptographic module is not permanently protected by the operating environment.

Technical Solutions

Technical realizations of tamper-evident characteristics include security seals (e.g., with special inscriptions and holograms), special covers and enclosures. Hereby, it is crucial that (i) the removal of these items should be sufficiently difficult and leave remaining traces that can be recognized by trained inspection personnel, (ii) the items should include special characteristics which are not commercially available, (iii) the faking of these items is sufficiently difficult and can be recognized by trained control personnel, and (iv) the items are controlled during manufacture and

delivery.

Some solutions may include 'obscure', i.e., uncommon approaches that are not obvious for an attacker. The combination of such efforts as well as the non disclosure of detailed information about its mechanical and chemical construction may achieve an acceptable security level. Possible approaches can be found in [144], as there are brittle packages and specially prepared surfaces such as 'Crazed Aluminium', 'Polished Packages' and 'Bleeding Paint'.

3.1.5 Tamper Response

Tamper response requires that the cryptographic module detects any intrusion attempts at the cryptographic boundary. As a consequence, the cryptographic boundary has to be permanently supervised.

Besides invasive attacks there might be some critical operational conditions which can lead to unforeseen events, such as tampering with the power lines and the environmental temperature. For their detection, special sensors are integrated inside the cryptographic module which permanently monitor the operating conditions.

One general requirement for tamper response is that an internal power supply must be available to detect and react to tamper attempts. Note that smart cards and similar tokens are not equipped with an internal power supply and therefore tamper response measures cannot always be guaranteed.

Another general requirement is that security sensitive information has to be zeroized as fast as possible, so that the zeroization cannot be stopped by the attacker. Therefore, critical data to be zeroized has to be stored in a RAM-based memory.

Technical Solutions

The most adequate technical solution to provide tamper responsiveness is an active shield at the cryptographic boundary, e.g., implemented by flexible printed circuit sensors [144]. The flexible printed circuit sensors include a mesh of conducting wires which are printed as close as possible to each other so that the connection of two near by wires causes short-circuits which can be detected. The wires may be made of silk-screened conductive paste which provide a high resistance and are difficult to

attach to. The mesh is embedded into a potting material. Attempts to remove the potting material impacts the overall electrical resistance of the wires, which is permanently measured and observed. Especially, chemical attacks result in both dissolving the potting and the insulation between the wires. Any alarm at the active shield has to result in ignition of the zeroization circuitry. The speed of the zeroization is crucial as an interruption would leave the cryptographic module in an intermediate state which might still contain sensitive data. Because of that, the preferred solution is a hardware-based implementation. A critical condition occurs if the internal power supply is no longer capable to activate the zeroization circuitry. To avoid this, the zeroization circuitry has to be triggered before the critical state of the internal power supply is reached.

An aspect is that a simple power-down of the SRAM data storage might be not sufficient due to data remanence caused by 'burn-in' of the data over a long time period (see [61]). One practical solution to minimize these effects is to periodically update the representation of the data stored in SRAM, e.g., by using an encryption key which is periodically changed internally. In [61] and [144] an alternative destructive approach for very sensitive applications is proposed, i.e., the exposure of the cryptographic device to high temperatures.

Monitoring attacks at the external interfaces cannot be detected by the cryptographic module. It is therefore necessary that these kinds of attacks based on electromagnetic emanation are made as difficult as possible. If compared to tamper-resistant measures, tamper responsive devices can benefit from the active shield of the device. One possible approach is that of [134], where it is proposed to switch capacitances between the internal power supply and the external power lines so that data-dependent signals at the external lines are minimized. To prevent electromagnetic emanation, a metal shielding case inside the cryptographic boundary may be an appropriate solution.

For the control of environmental conditions, the cryptographic module should be equipped with temperature, voltage, and frequency sensors. Whereas temperature changes expand slowly, the reaction time is not as critical as in the case of tampering attempts on the external voltage line and which demand an instant reaction. Another requirement might be that the cryptographic module detects any removal attempts. For these kinds of attacks, movement sensors are reasonable solutions.

3.1.6 Tamper Resistance

Typically, cryptographic modules claiming tamper resistance cannot actively detect tamper attempts at all circumstances. Prominent examples are smart cards and RFID tags that are supplied with voltage and clock externally. If the module is not powered on, penetration and modification attempts cannot be detected. Because of that, the internal construction of tamper-resistant modules has to withstand physical attacks. Moreover, a removal of a tamper-resistant module can usually not be prevented by the module itself.

The requirements to be fulfilled for tamper-resistant ICs are, e.g., set down in the protection profiles [133] and [63]. Appropriate technical solutions can be derived from these requirements.

Technical Solutions

Tamper resistance also implies to counteract reverse-engineering as much as possible. Some typical measures in the internal construction include the encryption of internal bus lines and memories which contain critical persistent data. Moreover, the layout should contain special characteristics, such as the scrambling of bus lines and memories as well as special logic styles.

Another important fact is the shrinking of structure widths in the semiconductor industry. Upcoming transistor technology is based on 90 nm gate widths. This technology demands specialized equipment, such as FIB (focused ion beam) workstations for penetration and modification. In the recent work [142] it was suggested to use a protective coating realizing physically unclonable functions (PUFs) on top of the IC so that FIB attacks are claimed to be prevented from retrieving key data. It is also important to note that test features used during manufacturing will no longer be available in the operating environment. Such requirements have to be specified as part of the life cycle of the product.

During start-up of the tamper-resistant module it is recommended to invoke self tests which verify the integrity of critical internal data and the correctness of the functionality. A special focus might be on an internal hardware-based random generator, which might be also needed for special countermeasures. A destruction of the random number generator has to be detected during start-up and operation.

At runtime the cryptographic module might be exposed to environmental conditions that are outside of the secure operation range. Critical operating conditions have to be recognized immediately. Therefore, the tamper-resistant module should be equipped with sensors for temperature as well as for the voltage and clock supply. Operation outside the secure range of parameters has to be prevented, and a secure state has to be entered, e.g., by enforcing a reset condition.

Actually, state-of-the-art fault analysis techniques make use of light injection and cause photoelectric effects in de-packaged integrated circuits. Optical fault induction allows for a great spatial precision and it is possible to target one SRAM cell. A survey on methods of fault induction, their effects, and possible countermeasures can be found in [14] and in Section 3.4.

Monitoring attacks at the external galvanic interfaces as well as at internal lines (after some successful internal modification during power-down) cannot be detected by the cryptographic module. This applies also to electromagnetic emanation. The construction of effective countermeasures is still an area of active research. A summary on implementation based countermeasure strategies is given in Section 3.3.6. Additional countermeasures may be foreseen in the application, e.g., usage counters to cryptographic operations and key management functions aiming at short key lifetimes. However, it should be noted that countermeasures usually implicate a degradation in performance, either in terms of chip size in the case of hardware countermeasures or in terms of efficiency and code size in the case of software countermeasures, both of which result in higher costs.

3.2 Physical Cryptanalysis

Physical Cryptanalysis covers

- Side Channel Cryptanalysis (Section 3.3) and
- Fault Channel Cryptanalysis (Section 3.4).

Roughly speaking, side channel cryptanalysis uses *physical observables* resulting from internal states of a cryptographic computation as input for cryptanalysis, while fault channel cryptanalysis uses *physical means* in order to modify internal states of an implementation. One may also consider side channel cryptanalysis as passive physical attacks on cryptographic implementations while fault channel cryptanalysis are active physical attacks.

The objective of this section is to provide a framework providing common properties of these two classes of physical implementation attacks. The arrangement in the attack scenarios of Section 3.1.1 is provided in Figure 3.2 and Figure 3.3.

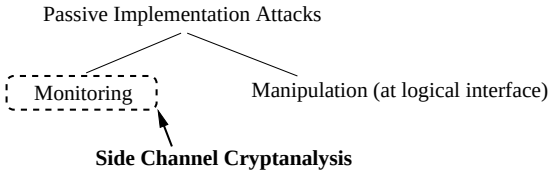


Figure 3.2: Passive implementation attacks and side channel cryptanalysis.

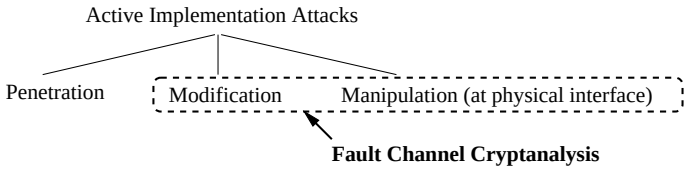


Figure 3.3: Active implementation attacks and fault channel cryptanalysis.

3.2.1 Adversary Model

At physical attacks, one assumes that the adversary is able to physically interact with the targeted cryptographic implementation. In the context of side channel cryptanalysis, interacting means to perform measurements of physical observables. Interaction in the case of fault channel cryptanalysis is active and aims at causing faults at the target implementation by applying physical means.

For simplicity, it is assumed that the cryptographic implementation includes one cryptographic algorithm operating on one secret (or private) cryptographic key. The cryptographic algorithm is assumed to be known, the secret (or private) key is unknown and constitutes the target of the attack. The computation of the cryptographic algorithm is started if the cryptographic module receives a specific query and it stops by sending a response. Both query and response may contain further data. In the case of physical cryptanalysis, the adversary objective is usually key recovery.

Following the threat model of a block cipher [69] one can build up a classification of attacks ranging from low assumptions on the adversary's capabilities to very powerful adversaries. However, physical cryptanalysis includes even attacks that do not require any part of ciphertext or plaintext or assumptions on key relations, and gain information only from physical leakage. As this class requires no assumptions, it has to be added on top of the list.

- *null attack*: The adversary does not obtain any part of ciphertext, nor any part of plaintext, nor any key relations.
- *ciphertext-only attack*: The adversary obtains the ciphertext or part of the ciphertext. Usually, one assumes, that the plaintext is based on special character sets, e.g., ASCII.
- *known plaintext attack*: The adversary obtains some pairs of corresponding plaintext and ciphertext, i.e., plaintext-ciphertext pairs.
- *non-adaptive chosen plaintext attack*: The adversary is able to choose the plaintext and obtains the corresponding ciphertext. The choice of the plaintext does not depend on previously revealed plaintext-ciphertext pairs.
- *adaptive chosen plaintext attack*: The adversary is able to choose the plaintext and obtains the corresponding plaintext-ciphertext

pair. The choice of the plaintext is adaptive on previously revealed plaintext-ciphertext pairs.

- *non-adaptive chosen ciphertext attack*: The adversary is able to decrypt arbitrary ciphertexts and obtains the corresponding plaintext. The choice of ciphertext does not depend on previously revealed plaintext-ciphertext pairs.
- *adaptive chosen ciphertext attack*: The adversary is able to decrypt arbitrary ciphertexts and obtains the corresponding plaintext. The choice of the ciphertext is adaptive on previously revealed plaintext-ciphertext pairs.
- *adaptive combined chosen plaintext and chosen ciphertext attack*: The adversary is able to decrypt and encrypt arbitrary texts. The choice of the plaintext and ciphertext is adaptives on previously revealed plaintext-ciphertext pairs.
- *related key attack*: The adversary knows mathematical relations between different key values used, but not their actual values.

Mathematical cryptanalysis considers four crucial parameters for attack efficiency.

- *Time complexity*, i.e., the overall computational time needed for an attack.
- *Data complexity*, i.e., the overall data such as plaintext-ciphertext pairs needed for an attack.
- *Memory complexity*, i.e., the memory needed for the computation, e.g., to store plaintext-ciphertext pairs during computation.
- *Success probability*, i.e., the average success probability when repeated several times.

For implementation attacks, additional parameters have an impact on attack efficiency.

- *Knowledge of implementation details:* This item concerns the 'a priori' knowledge of the adversary on the implementation. In implementation cryptanalysis, one usually encounters the fact that

the implementation is not fully known or understood. The attack may be carried out as a *white-box attack* at which all construction details of the implementation are known to the adversary or in form of a *black-box attack* at which implementation details are not known. In between these two extremes are *grey-box attacks*, in which case some implementation details are known. White-box attacks are the ones achieving maximum power.

- *Physical access:* The adversary may have *direct physical access* to the cryptographic module which enables the use of instantaneous observables and short-range physical means or only *indirect physical access* to the cryptographic module, e.g., via a network. In the latter case, an adversary cannot use observables or apply physical means in the near vicinity of the module.

- *Laboratory equipment:* The laboratory equipment can be decisive in terms of the quality of the measurement process or fault injection process and therefore for the overall efficiency of an attack.

- *Number of stages:* The adversary may have physical access to an identical cryptographic implementation or a simulator of the cryptographic implementation that can be used for profiling of the implementation. If a profiling stage exists, the attack is said to consist of two stages. The first stage is called *profiling stage* and the second stage is the *key recovery stage*. The profiling stage aims at gaining 'a-priori' knowledge on the physical characterization that in turn is applied at the key recovery stage. The device used for the profiling stage is called the *profiling device*. The device targeted in the attack is said to be the *target device*. If a profiling stage does not exists, the attack is a *one-stage attack* and can not benefit from an 'a-priori' characterization at the target device, i.e., this is the setting with the lowest adversary capabilities.

- *Profiling stage:* If a profiling stage exists attacks are further classified according to adversary capabilities at profiling. The adversary may know the key or may be able to choose any key for profiling. If the implementation uses masked internal data representations this leads to the refinements of (i) attacks without knowing masks, (ii) attacks with known masks, and (iii) attacks with chosen masks.

The most powerful adversary can choose any internal data representation, i.e., it applies profiling with chosen keys and chosen masks.

- *Models and algorithms:* Models on the physical properties of a cryptographic implementation and algorithms to turn these physical properties into a cryptanalytic attack have an impact on the efficiency of an attack, and thereby on the time complexity, data complexity, memory complexity, and success probability. The use of algorithms for physical cryptanalysis usually depends on further adversary parameters, i.e., the knowledge of implementation details, the kind of physical access, and the assumptions for the stages of an attack.
- *Number of cryptographic operations:* The number of cryptographic operations is usually a measure for the time and data complexity of an attack.
- *Number of cryptographic devices:* The number of cryptographic devices is of interest if the adversary does not succeed with one cryptographic device, e.g., because of a usage counter or a permanent failure of the device. It is usually a measure for the time and data complexity of an attack.

Perfect security of an implementation implies that the entropy loss of an internal state, e.g., the secret (or private) cryptographic key of a block cipher, is zero after applying physical cryptanalysis. As consequence, an adversary does not learn anything about the internal state of an implementation. Correspondingly, a cryptographic implementation is considered to be totally broken if the remaining entropy of the internal state approaches zero. If physical cryptanalysis achieves a partial entropy loss one faces the crucial question whether the remaining entropy still resists other types of cryptanalytic attacks, e.g., a brute-force attack.

In physical cryptanalysis one often deals with so-called *subkeys* of the overall cryptographic key. A subkey may consist of one or some bits of the overall cryptographic key. If one is able to compromise subkey by subkey, one successively reduces the key space. An adversary has success if either the cryptographic key can be directly disclosed or the efforts to run a brute-force attack are 'sufficiently' reduced.

3.3 Side Channel Cryptanalysis

Side channel cryptanalysis uses physical observables resulting from internal states of a cryptographic computation as additional information source for cryptanalysis. The term *physical observable* is understood to be any property of a physical system state determined by a physical operation. The outcomes of the measurement of a physical observable are real-valued, i.e., elements of $\mathbb{R}$. In the context of this thesis, measurements concern the execution time, power, and EM emanation of a device. Accordingly, the physical observables are scalars (see Section 3.3.2 on Timing Analysis), vectors for instantaneous leakage (see Section 3.3.3, 3.3.4, and 3.3.8 on Simple Analysis, Differential Analysis, and Multivariate Analysis) and tensors if multiple instantaneous leakage channels are used in parallel.

The underlying working hypothesis for side channel cryptanalysis is that the internal state S of a cryptographic device while it computes a cryptographic function has an impact on a physical observable Y. The process of measuring Y can be seen as observing the outcome of a stochastic variable. The mapping of S upon Y is implementation dependent and constitutes the *side channel*. The task for side channel cryptanalysis is to recover the internal state S by measuring the outcomes of Y.

Due to the complexity of the overall computational process, the adversary generally has to simplify leakage models for S. Common models that have turned out to be surprisingly successful are summarized in Section 3.3.3. By assuming a leakage model for S side channel cryptanalysis can be able to reveal the internal state of S.

In side channel cryptanalysis, the adversary objective is key recovery. Informally speaking, an adversary is successful if side channel enhanced cryptanalysis leads to a critical entropy loss of a secret cryptographic key. The internal state S of a cryptographic device can be interpreted as a cryptographic key, e.g., for an 128-bit secret key, it follows $S \in \{0,1\}^{128}$, or a combination of key space and message space, e.g., $S \in \{0,1\}^{128} \times \{0,1\}^{128}$. This is to be made more precise in Section 3.3.1.

3.3.1 Refinements of Adversary Model

For side channel attacks, one assumes that the adversary is able to perform measurements of physical observables while using the targeted cryptographic implementation. Here, implementation-based attack parameters introduced in Section 3.2.1 are refined for side channel cryptanalysis if applicable.

- *Physical access:* In the context of side channel cryptanalysis indirect physical access means that an adversary cannot use observables in the near vicinity of the module. For example, if access to the target device is only feasible via a network this usually implies that only cumulative observables can be used, e.g., the overall execution time of a cryptographic operation.

- *Laboratory equipment:* In the context of side channel cryptanalysis laboratory equipment means measurement equipment that may be decisive in terms of the quality of the measurement process and for the efficiency by optimizing the signal-to-noise ratio in the measurement process.

- *Models and algorithms:* The task of side channel cryptanalysis is a signal detection problem on the basis of measurement data. Assumptions on the leakage model may be needed for the algorithms. The choice of statistical tests and decision strategies may be crucial for the algorithms' efficiency in cryptanalysis.

- *Number of cryptographic operations* In the context of side channel cryptanalysis the number of cryptographic operations is a synonym for the number of measurements. By increasing the number of measurements N, the impact of noise in the statistic is suppressed with $\sqrt{N}$ (see Section 2.1.1), thereby enabling the detection of minor side channels.

Perfect security of an implementation on observable Y implies that the entropy loss of an internal state S, i.e., the cryptographic key of a block cipher, after observable Y is measured is zero, i.e., $H(S) = H(S|Y)$. For the definition of adversary success, it is useful to introduce a threshold H_T on remaining entropy that is considered to be an appropriate security bound for block ciphers against brute-force attacks,

e.g., an implementation of a block cipher is considered to be *threshold H_T-secure on observable Y* if a lower bound H_T on the remaining entropy $H(S|Y)$ exists so that $H(S|Y) \geq H_T$. Currently, one considers that $H_T = 80$ is an adequate boundary to prevent brute-force attacks on symmetric ciphers, e.g., [13] estimates that current capabilities of intelligence agencies allow to recover a 75-bit key after 73 days. Correspondingly, an implementation of a block cipher is considered to be totally broken if H_T approaches zero.

3.3.2 Timing Analysis

In 1996 [72], timing analysis was the first side channel based attack invented in the public domain. This work provided the methodology to compromise keys of RSA, DSS and other cryptosystems by measuring the execution time of the overall cryptographic operation.

As in [72], the following description focuses on RSA implemented by modular exponentiation. To perform a modular exponentiation $c = a^b \bmod m$ in $\mathbb{Z}_m$, the bitwise representation $b = (b_{n-1}b_{n-2}\cdots b_1 b_0)_2$ is used. The *square and multiply* algorithm evaluates this representation, e.g., by starting from the most significant bit b_{n-1} , see Algorithm 3.1.

Algorithm 3.1 Square and multiply algorithm

Input: $a \in \mathbb{Z}_m$, $b = (b_{n-1}, b_{n-2}, \ldots, b_0)_2$
Output: $a^b \in \mathbb{Z}_m$
1: $c = 1$; .
2: **for** j from $n-1$ downto 0 **do**
3: $\quad c \leftarrow c \cdot c \bmod m$; {Squaring}
4: $\quad$ **if** $b_j = 1$ **then**
5: $\quad\quad c \leftarrow c \cdot a \bmod m$; {Multiplication}
6: $\quad$ **end if**
7: **end for**
8: Return (c);

Obviously, the time needed to process one exponent bit is increased if the bit is set to '1' because of the additional multiplication. Note that the original publication by Kocher does not exploit the observation of internal characteristics during the processing of the algorithm, e.g.,

the execution times of single multiplications. A straightforward implementation of the modular exponentiation as shown in Algorithm 3.1 is extremely vulnerable on Simple Side Channel Analysis of instantaneous observables (see Section 3.3.3) under the assumption that the operation used for squaring $c \leftarrow c \cdot c \bmod m$ and multiplication $c \leftarrow c \cdot a \bmod m$ can be distinguished (e.g., by their timing or leakage pattern). However, the timing attack presented here can be also applied by an adversary that has indirect physical access to the cryptographic device, e.g., by eavesdropping a network communication line (see Figure 3.4).

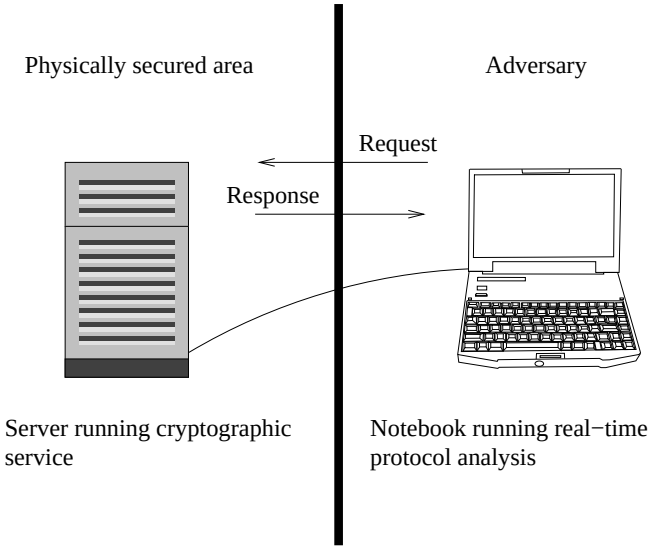


Figure 3.4: Timing measurement at a network connection to a server implementing a cryptographic operation.

For a successful timing attack it is required that the execution time of the elementary operations squaring and multiplication in Algorithm 3.1 is data dependent, i.e., it depends on its operands c and a. The attack requires a high number of computations using varying input data a_i and a fixed secret key b. The overall execution time measured is denoted by $T(a_i)$ wherein $i \in \{1, ..., N\}$ runs through all N invocations of the algorithm. It is assumed that the input data a_i as well as the modulus

m are known to the adversary. The adversary objective is to disclose the secret exponent b of the target device.

Further, this attack requires that the adversary is able to simulate or predict the timing behavior of the attacked device rather accurately, i.e., the execution time has to be either simulated if the adversary owns or is able to build a simulator of the device, or directly measured by using a profiling device that can be loaded with chosen values for a_i, m, and b. Profiling and key recovery builds a recurring cycle in this attack, i.e., the profiling device has to be available in the key recovery stage.

In the following, the bit-by-bit approach of [72] is explained. One assumes that the attacker has already compromised the l most significant bits of b, i.e., bits $b_{n-1}, \ldots, b_{n-l}$. The statistical decision problem is whether bit b_{n-l-1} equals 0 (hypothesis H_0) or b_{n-l-1} equals 1 (hypothesis H_1).

The execution time for the processing is determined for both hypotheses H_0 and H_1, respectively, by loading the profiling device or simulator with $(b_{n-1}, \ldots, b_{n-l}, 0)_2$ for H_0 and $(b_{n-1}, \ldots, b_{n-l}, 1)_2$ for H_1. Let $T^{H_0}(a_i)$ be the execution time for H_0 and $T^{H_1}(a_i)$ the time for H_1 of the $l+1$-bit sized exponent at the profiling device.

For the hypothesis test, data sets are then built according to

$$H_0: \ \{T(a_i) - T^{H_0}(a_i) | 1 \leq i \leq N\}$$

and

$$H_1: \ \{T(a_i) - T^{H_1}(a_i) | 1 \leq i \leq N\}.$$

The underlying idea of the attack follows Proposition 2.3 which says that for two normally distributed stochastic variables X_1 and X_2 with variance σ_1 and σ_2, their sum $X_1 + X_2$ is normally distributed with variance $\sigma_1^2 + \sigma_2^2$. In a simple explanation one may consider the time t_j needed for bit j and assume that the execution times for exponent bits are effectively independent from each other. Further, let assume that S_j^2 denotes the variance of bit j and, for simplicity reasons, that $S_j^2 = S^2$ for all j. The overall variance of the entire n-bit modular exponentiation is then nS^2. For the correct hypothesis, one expects a slightly reduced variance of the entire data set, as the execution time of the correct hypothesis equals the observed computation for the $l+1$-bit sized exponent part, i.e., one expects a variance of $(n-l-1)S^2$. For the false hypothesis, however, one has included the time for a false

operation for the predicted exponent bit. As this false operation is seen to be independent on the observed correct operation, one expects an increase variance of $(n - l + 1)S^2$. Significantly different variances can be detected, e.g., with the F-Test of Section 2.1.2.

Besides measurement uncertainties, i.e., additional delay times of the network, noise consists of the remaining number of non-predicted bits b_{n-l-2} to b_0 and their contribution to the overall execution time. Because of that, one encounters a high noise level for a small value of l which complicates the signal detection problem at the starting point of this attack and a highly reduced noise floor towards high values of l. As false hypotheses lead to an increased variance of the data sets above, this attack has an inherent error detection and correction property.

Refinements of the leakage models are possible, e.g., one may replace the variance for one bit by the variance for the multiplication and the variance for the squaring operation. Practical tests of the timing attack were done in [47] and [125]. Both publications use a Montgomery multiplication with a constant execution time except for the additional subtraction that is done if the intermediary result of the multiplication is greater than the modulus. In summary, there are only two possible execution times for each multiplication which simplifies the leakage model. For a 512-bit exponent [47] reported that approximately 350,000 measurements are needed, whereas the improved statistics of [125] reduced the number of measurements by a factor of up to 50. Another variant of the timing attack can be applied to a CRT implementation of RSA if implemented using Montgomery multiplication (see [123]). The application to an OpenSSL-based web server was demonstrated in [33].

As already pointed out by [72], RAM cache hits of general purpose computers can produce timing characteristics in implementations of block ciphers, that make use of table based substitutions as, e.g., DES. These cache attacks were elaborated for DES by [111] and for AES by [21].

Very recently, a new class of timing attacks to general purpose high-performance CPUs has been emerged: *Branch Prediction Analysis* [3]. The core idea is to exploit the deterministic processing of branch prediction units of modern CPUs that predict the expected instruction sequence before obtaining the actual result of the branch directive. For this kind of attack, the adversary is assumed to be able to run an unprivileged process in parallel to a target process computing a cryptographic

algorithm on the same processor.

3.3.3 Simple Side Channel Analysis

In 1998, the use of instantaneous observables was initially announced and demonstrated by Paul Kocher et al. [73]. *Simple Power Analysis (SPA)* was introduced as a technique that directly interprets power consumption measurements that are collected during the cryptographic operation [73]. SPA allows to make inner parts of computations of the algorithm visible, just by observing the instantaneous measurement outcomes. Reference [73] also includes the proposal for Differential Power Analysis (DPA) that will be explained as part of Section 3.3.4.

The alternative use of electromagnetic (EM) side channels was proposed by [116] and [53]. In 2001, the authors of [53] conducted experiments on EM side channels. Accordingly, the attack was named *Simple Electromagnetic Analysis (SEMA)*.

In the context of this thesis, the term *Simple Side Channel Analysis (SSCA)* will be used if the methods and results discussed are applicable, regardless of the physical nature of the side channel. This notation has been already introduced, e.g., by [87].

In the experiments of [73], the power consumption of a microprocessor based smart card was measured over a small (e.g., 50 Ω) resistor that was inserted in series with the power or ground input. The side channel used is the continuous voltage measured across this resistor, which is proportional to the driven current according to Ohm's law. In the EM experiments of [53], small probes, e.g., hand-made solenoids of coiled copper with outer diameters of a few hundred micrometers are used. An amplifier for the low output signal of the antenna may be necessary. Additional parameters may have an impact on the observed EM side channel, e.g., one may vary the spatial position of the antenna on top of the chip or even remove the packaging of the chip to position the antenna as close as possible. For the measurement of instantaneous side channels one needs a digital storage oscilloscope or a digitizer. A valuable summary of measurement set-ups for power analysis and EM in the near and far field can be found in [87]. As part of this thesis, the Agilent infiniium 54832D oscilloscope was used. The side channel is *sampled*, i.e., its value is recorded, at evenly spaced intervals in time. One obtains a p-dimensional vector $\vec{i} = (i_1, \ldots, i_p)$ with p being the number

of sampling points. This vector is also said to be a *measurement trace*. For the choice of the sampling rate the sampling theorem [115] should be considered, i.e., the rate should be sufficiently high to give at least two points per cycle of the highest frequency present or the bandwidth should be limited by a known analog filtering of the continuous signal. For further analysis, one usually transfers the measurement data to a general purpose computer. Figure 3.5 shows the principle measurement set-up for both SPA and SEMA.

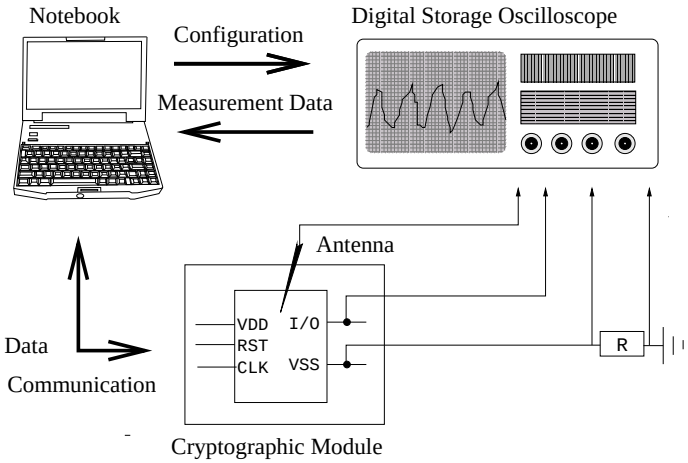


Figure 3.5: Measurement set-up for both power consumption and EM emanation

Both [73] and [53] resolved inner structures of an implementation of the *Data Encryption Standard (DES)* [101]. A plot of the measurement trace can show clearly the timing position of each DES round. One can even go to deeper details of each DES round, e.g., one can break it down to observe single instructions of the microprocessor. It was shown in [53] that the signal shapes of power and EM channel differ, especially the EM signals are of higher frequencies. Figure 3.6 and Figure 3.7 illustrate such findings for the AES implementation on an ATM163 microprocessor that will be under consideration as part of this thesis.

By observing single instructions or a sequence of instructions, SSCA succeeds in breaking a cryptographic implementation if

- the implementation makes use of conditional branching in dependency on key bits which in turn causes key dependent time variations or

- the pattern of the side channel reveals information on the value of key bits, e.g., if the side channel observed differs significantly in magnitude between key bit 0 and 1 at certain instants.

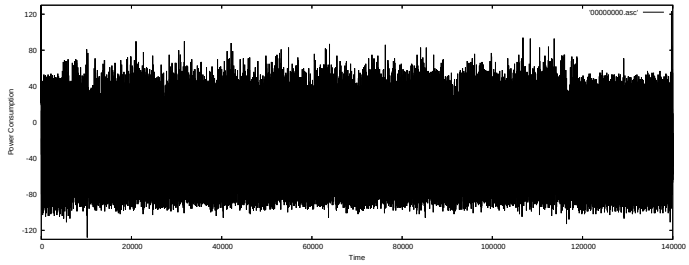


Figure 3.6: Power consumption measurement trace of an AES implementation running on an ATM163 microprocessor at 3.57 MHz. Round structures of the AES are visible. The sampling rate was 100 MHz.

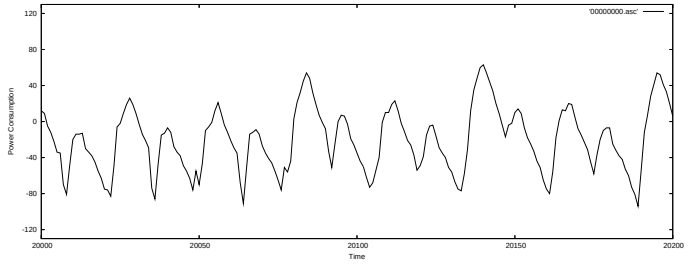


Figure 3.7: Zoom into Figure 3.6. Single instructions can be observed.

SSCA like attacks may differ strongly in the concrete assumptions on the adversary's capabilities. Except for very obvious implementation weaknesses, SSCA attacks often require a profiling stage to localize critical clock cycles of the implementation.

In case the noise floor hides the signal and provided that the cryptographic algorithm can be run multiple times, one can average the traces.

As given in Proposition 2.5 and Proposition 2.6 the precision in determining the empirical mean value is increased with the number of measurements N according to $\sqrt{N}$. If one can use a profiling device with chosen key values, one may ask whether small key dependent side channels can be resolved. For instance, the profiling device may be operated with two key values that differ only in one bit. The set of measurement traces with the relevant key bit set to '0' is said to be S_0, accordingly S_1 is the set of measurement traces at which the relevant key bit is set to '1'. If one can find a significance test that disproves that S_0 and S_1 stem from the same population at some instants in time, one has revealed an exploitable side channel. For the significance test one may use the T-Test of Theorem 2.9 that evaluates significantly different means. The T-Test can also be seen as an improvement for finding key dependent differences if compared to the approach of computing the difference of average traces that is widely used in many SSCA scenarios discussed in more detail below.

Inferential Power Analysis (IPA) as proposed by [49] is an attack with a profiling stage. For application at DES, profiling requires a fixed key and different plaintext data. It is assumed that each DES round is implemented in the same way, so that one can compute the average trace of each round and finally the super-average of all 16 rounds. Typically, before computing the super-average a relative alignment, i.e. a relative displacement in time, of all round average curves is needed. Profiling consists of

- localization of key dependencies and
- key bit identification

in the measurement traces. Key locations are found by computing the difference of the super-average and the round-averages. By doing so, one cancels out the non-key dependent parts of the leakage and may reveal key dependent signals. The identification of the key bit responsible for a signal demands for further analysis, e.g., one may use two fixed keys that differ only in one key bit. As result of profiling, the adversary obtains a list that maps the key dependent locations to each subkey bit. Key recovery is then done by evaluating these locations in the measurement traces of the target device.

More detailed investigations on the power dissipation and noise characteristics of smart cards can be found, e.g., in [94, 44]. Reference

[94] revealed that the activity on the data and address bus is a dominant cause of power consumption changes on a microprocessor. Besides *single-bit* dependent leakage, two types of *multi-bit* information leakage are observed and have been confirmed by many researchers, e.g., [94, 44, 90, 31, 81, 87, 129]:

- *Hamming weight leakage*, i.e., one observes leakage according to the Hamming weight of data x processed, i.e., the number of bits of x that are set to 1, and

- *Hamming distance leakage* (also called *transition count leakage*, e.g. by [94]), i.e., one observes leakage according to the Hamming weight of $(x \oplus x')$ of data x processed and data x' processed at the previous or subsequent computation state.

Hamming weight leakage can be seen as a special case of Hamming distance leakage if x' is zero. One example for Hamming weight leakage is a precharged bus design [94]. Hamming distance leakage is understood by the current driven due to switching gates of the circuit. The interpretation of Hamming distance leakage is limited as long one does not know the related data x'.

SSCA attacks depending on the Hamming weight leakage model have been proposed by [94, 25, 44]. For instance, one may assume for an 8-bit microprocessor, that the Hamming weight of 8-bit portions of the key can be precisely measured, e.g., when bytes are moved to or from internal memories. The average entropy loss caused is then 2.54 per 8-bit subkey. In case of an n-bit key, the entropy loss sums up to $\frac{2.54}{8}n$, leaving a remaining entropy of $\frac{(8.0-2.54)}{8}n$. Hamming weight leakage may lead to a total break if the key schedules includes bit shifts as the DES [25]. DES has 16 rounds and in each round six subkey bytes are computed. If one can carry out precise measurements on the Hamming weight, one obtains a system of 96 equations for 56 unknown key bits which is uniquely solvable [25].

For public key cryptosystems, SSCA has turned out to be very easy at the 'square and multiply' algorithm in Algorithm 3.1 provided that one can observe the 'if' instruction or, alternatively, is able to directly distinguish the squaring $c \cdot c \bmod m$ from the multiplication $c \cdot a \bmod m$. A similar SSCA attack is feasible at straightforward implementation of

the 'double and add' algorithm for *Elliptic Curve Cryptosystems (ECC)* [42].

Reference [95] proposes two further SSCA attacks on the modular exponentiation that both need a profiling stage: *Single-Exponent, Multiple Data (SEMD)* and *Multiple-Exponent, Single Data (MESD)*.

For the profiling stage, SEMD requires to carry out many exponentiations with a known fixed exponent and varying data. In the key recovery stage, many exponentiations with the unknown secret exponent and varying data are carried out. By averaging the traces of both stages and subtracting the resulting average traces of the profiling and target device, one may be able to decide, whether the corresponding exponent bits agree or differ.

MESD requires profiling of the exponentiation with chosen exponents and, additionally, that both the profiling and the target device will exponentiate the same constant value, that need not to be necessarily known by the adversary. For MESD, one uses a recurring cycle of profiling and key recovery, e.g., for each next bit of the exponent. Averaging and subtracting at MESD is much more powerful if compared to SEMD, as variations due to intermediate data computed are not present as long as the exponent bit stream is identical. At that time where intermediate data start to differ, abruptly significant differences are expected to show up.

3.3.4 Differential Side Channel Analysis

Besides Simple Power Analysis (SPA), the fundamental work [73] is dedicated to Differential Power Analysis (DPA) and its application at the Data Encryption Standard (DES) [101]. As discussed in Section 3.3.3, the underlying methodology proposed is generally applicable to all physical instantaneous observables, e.g., one may measure EM emanation instead of power consumption. For the EM channel, [53] introduced the corresponding term *Differential Electromagnetic Analysis (DEMA)*. This thesis follows again the more recent notation, that both DPA and DEMA can be seen as special kinds of *Differential Side Channel Analysis (DSCA)*. DSCA needs the knowledge of either the plaintext or the ciphertext as a pre-condition. This is necessary to compute intermediate results at the beginning or the end of the cryptographic algorithm. DSCA has turned out to be highly effective, especially at implementa-

tions of block ciphers. DSCA does not include a profiling stage and generally does not require special knowledge about the cryptographic implementation or special test conditions. A strength of DSCA lies also in the fact that DSCA, if successful, automatically recovers critical clock cycles of an implementation.

DSCA strongly benefits from the fact that common block ciphers are built as substitution permutation networks, i.e., at each stage they consist of substitution functions and permutation functions. Such a stage is typically called a *round* and has associated *roundkeys* that are derived from the block cipher key. A *subkey* is understood as a small portion of bits of a roundkey that jointly enter a sub-function of a block cipher. The key space of a subkey is sufficiently small so that DSCA hypothesis testing can be done for each subkey candidate.

From a high-level perspective, DSCA consists of the tasks

- **Measurement:** The adversary conducts N side channel measurements of a p-dimensional vector $\vec{i}_j = (i_1, \ldots, i_p)_j = (i_{j1}, \ldots, i_{jp})$ ($j \in \{1, \ldots, N\}$) while the cryptographic device computes parts of a block cipher using either known plaintext p_j (or known ciphertext c_j) with a secret key k_{cipher}.
- **Analysis (Key Recovery):** DSCA aims at the key recovery of one subkey $k^\circ \in \{0,1\}^m$ of k_{cipher}. The adversary applies a statistical test combining an intermediate result of the cipher that is predictable for each subkey hypothesis k and the measurement traces $\vec{i}_j$. The outcome of the statistical test is a p-dimensional measure $\vec{\Delta}_k$ for the probability that indeed this subkey hypothesis k was used in the side channel measurements. As result of DSCA, the adversary outputs a permutation π of all subkey values

$$\pi = \begin{pmatrix} 0 & 1 & . & . & . & 2^m - 1 \\ \tilde{k}_0 & \tilde{k}_1 & . & . & . & \tilde{k}_{2^m-1} \end{pmatrix}$$

  such that $\mathbb{P}(\tilde{k}_0 = k^\circ) \geq \mathbb{P}(\tilde{k}_1 = k^\circ) \geq \ldots \geq \mathbb{P}(\tilde{k}_{2^m-1} = k^\circ)$, i.e., the key hypotheses are sorted for decreasing probability and $\tilde{k}_0$ is the most probable key guess as result of DSCA. DSCA is successful if $\tilde{k}_0 = k^\circ$.

Given DSCA results for many subkeys, the adversary can carry out an exhaustive key search on k_{cipher} starting with the most probable sub-

key candidates if a plaintext-ciphertext pair is available. If a plaintext-ciphertext pair is not available, an exhaustive key search is not applicable and the adversary may try to verify the correctness of key guesses with DSCA means, i.e., the adversary may repeat DSCA either by increasing N or by using other intermediate results of the implementation.

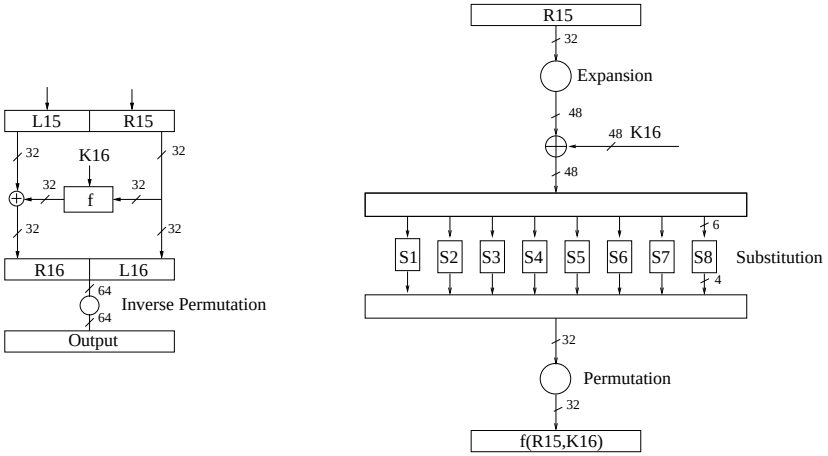


Figure 3.8: Last round of DES (left plot) and DES inner function f in the last round (right plot).

The following explanations give the sketch of the attack on the DES (see Figure 3.8). At DES, the substitution layer consists of 8 different substitution functions, generally referred to as *S-box* lookup, each mapping six input bits to four output bits. The input to the S-box S is the exclusive-or of 6-bit x_j that is part of the ciphertext c_j (or plaintext p_j) and the corresponding subkey $k^\circ \in \{0,1\}^6$, i.e., the operation is $S(x_j \oplus k^\circ)$[1]. As suggested by [73], one can set up 2^6 key hypotheses k on the value of subkey k° entering one S-box. If $k = k^\circ$, the outcome of an intermediate value depending only on known x_j and k is correctly predicted with probability 1. For the remaining $2^6 - 1$ key hypotheses, the prediction of the same intermediate value behaves like a random function with a correct prediction rate of about 0.5.

[1]For simplicity, one S-box is exemplarily chosen for the explanation. Note that there are eight different S-Boxes S_1 to S_8 in DES.

For each key hypothesis k, one partitions the measurement traces $\vec{i}_j$ into two sets $S_{k,0}$ and $S_{k,1}$ according to any deterministic function of x_j and k, e.g., one bit of $S(x_j \oplus k)$. Such a deterministic function is a so-called *selection function* (sometimes also referred to as *partitioning function*, e.g., by [94]). For binary selection functions, each set represents a different bit outcome that may reflect the correct key value in the side channel measurements. If one can find a significance test that disproves that $S_{k,0}$ and $S_{k,1}$ stem from the same population for one key hypothesis k at some instants in time, one has probably identified the correct subkey k_0. Note that DSCA is a probabilistic procedure whereat the significance increases with $\sqrt{N}$. DSCA includes three inner algorithms whose concrete choice may be crucial for DSCA success. The principle of DSCA can be seen as an adaptive modular conception. Iterations of different inner algorithms may improve the assurance in DSCA subkey indexing.

- The **selection function** is used for the partitioning of the measurements in two (or more) sets for each subkey hypothesis k. It depends on the concrete block cipher used and may assume a certain physical leakage model of the cryptographic device.
- The **statistical test** operates on the two (or more) sets and indicates a significance level whether the sets stem from the same population or not. The significance test outputs a p-dimensional test statistic $\vec{\Delta}_k$ for each key hypothesis k considering all p sampled points in time.
- The **key ranking** operates on the outcomes $\vec{\Delta}_k$ of the statistical test for all k and outputs a permutation π of the key hypotheses according to the vectorial test statistic $\vec{\Delta}_k$

$$\pi = \begin{pmatrix} 0 & 1 & . & . & . & 2^m - 1 \\ \tilde{k}_0 & \tilde{k}_1 & . & . & . & \tilde{k}_{2^m-1} \end{pmatrix}$$

  such that $\mathbb{P}(\tilde{k}_0 = k^\circ) \geq \mathbb{P}(\tilde{k}_1 = k^\circ) \geq \ldots \geq \mathbb{P}(\tilde{k}_{2^m-1} = k^\circ)$, i.e., $\tilde{k}_0$ is the most probable key guess as result of DSCA.

The generic algorithm for single-bit DSCA is provided in Algorithm 3.2. It has to be made precise by the choice of inner algorithms. Note that single-bit DSCA does not require any physical leakage model.

For a total key exposure, DSCA has to be also applied at the remaining S-boxes. If one succeeds in revealing each subkey of a DES round, one knows 48 key bits. In case of a single DES and if one has a plaintext-cipher pair available one may find out the 56-bit cipher key by a small exhaustive key search. Otherwise one may repeat DSCA at the second (or second-to-last) round to reveal the missing key bits.

The original work [73] considers a ciphertext-only attack, i.e., DSCA starts at the last DES round. [73] proposes a selection function that uses one bit b of the 32-bit intermediate result L_{15} at the beginning of the sixteenth DES round (see Figure 3.8). This bit b does only depend on a 6-bit subkey k° of the sixteenth DES round and the corresponding 6-bit $x_j \subset c_j$, i.e.

$$d\left(x_j, k\right) = \begin{cases} 1, & \text{if } b = 1; \\ 0, & \text{if } b = 0; \end{cases}$$

For each hypothesis k, a simplified version of the difference of means test is proposed as the statistical test in [73]

$$\vec{\Delta}_k = \frac{\sum\limits_{\vec{i}_j \in S_{k,1}} \vec{i}_j}{|S_{k,1}|} - \frac{\sum\limits_{\vec{i}_j \in S_{k,0}} \vec{i}_j}{|S_{k,0}|}$$

where $|S_{k,0}|$ and $|S_{k,1}|$ denotes the number of elements in $S_{k,0}$ and $S_{k,1}$, respectively. After computing the p-dimensional mean vector of all traces in each set, the differential p-dimensional vector $\vec{\Delta}_k$ is given as the difference of the mean vector in set $S_{k,1}$ and the mean vector in set $S_{k,0}$. If k is incorrect, one expects that $\vec{\Delta}_k$ approaches zero for all p scalar elements of $\vec{\Delta}_k$. However, if k is correct and the side channel depends on the selection function used, one may be able to detect peaks at some scalar elements of $\vec{\Delta}_k$ where the outcome of the selection function is correlated with the outcome of the physical observable. The decision strategy of [73] is built on the visual inspection of the differential traces. One decides in favor of the hypothesis $\tilde{k}_0$ for which some scalar elements of $\vec{\Delta}_{\tilde{k}_0}$ take the absolute maximum values among all other scalar elements of $\vec{\Delta}_k$ with $k \neq \tilde{k}_0$. A simple key ranking algorithm that may be used in an automated test is given in Algorithm 3.3. Algorithm 3.3 may be refined in order to consider additionally the number of significant scalars in $\vec{\Delta}_k$. Another alternative is the use of a binary randomness test as proposed

Algorithm 3.2 A generic (single-bit) DSCA algorithm

Input: (i) Measurement data $(\vec{i}_1, \cdots \vec{i}_N)$ with $\vec{i}_j \in \mathbb{R}^p$ for all $j \in \{1, \ldots, N\}$, while the cryptographic device computes parts of a block cipher using the known data $x_j \in \{0,1\}^m$ and a secret subkey $k^\circ \in \{0,1\}^m$,
(ii) a selection function $d : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$,
(iii) a statistical test $T : S_0 \times S_1 \to \mathbb{R}^p$ operating on two disjoint sets S_0 and S_1 of measurement data $\vec{i}_j \in \mathbb{R}^p$ with $|S_0| + |S_1| = N$, and
(iv) an indexing algorithm $I : \underbrace{\mathbb{R}^p \times \cdots \times \mathbb{R}^p}_{2^m} \to P$ with P being the set of all permutations of the elements $\{0, \ldots, 2^m - 1\}$.

Output: a permutation of subkey hypotheses

$$\pi = \begin{pmatrix} 0 & 1 & . & . & . & 2^m - 1 \\ \tilde{k}_0 & \tilde{k}_1 & . & . & . & \tilde{k}_{2^m-1} \end{pmatrix}$$

such that $\mathbb{P}(\tilde{k}_0 = k^\circ) \geq \mathbb{P}(\tilde{k}_1 = k^\circ) \geq \ldots \geq \mathbb{P}(\tilde{k}_{2^m-1} = k^\circ)$.

1: **for** k from 0 to $2^m - 1$ **do**
2: $S_{k,0} \leftarrow \{\}$; $S_{k,1} \leftarrow \{\}$;
3: **end for**
4: **for** j from 1 to N **do**
5: **for** k from 0 to $2^m - 1$ **do**
6: **if** $(d(x_j, k) = 0)$ **then**
7: $S_{k,0} \leftarrow S_{k,0} \cup \vec{i}_j$; {Partitioning in $S_{k,0}$ for each k}
8: **else**
9: $S_{k,1} \leftarrow S_{k,1} \cup \vec{i}_j$; {Partitioning in $S_{k,1}$ for each k}
10: **end if**
11: **end for**
12: **end for**
13: **for** k from 0 to $2^m - 1$ **do**
14: $\vec{\Delta}_k \leftarrow T(S_{k,0}, S_{k,1})$; {Applying a statistical test for each k}
15: **end for**
16: $\{\tilde{k}_0, \ldots, \tilde{k}_{2^m-1}\} \leftarrow I(\vec{\Delta}_0, \ldots, \vec{\Delta}_{2^m-1})$; {Sorting of key hypotheses}
17: Return $(\{\tilde{k}_0, \ldots, \tilde{k}_{2^m-1}\})$;

by [44] that checks whether features in $\vec{\Delta}_k = (\Delta_{k1}, \ldots, \Delta_{kp})$ can occur by chance when sampling from a uniform distribution.

Algorithm 3.3 A simple indexing algorithm (sorting for maximum absolute scalars)

Input: 2^m p-dimensional samples $\vec{\Delta}_0 = (\Delta_{01}, \ldots, \Delta_{0p}), \ldots, \vec{\Delta}_{2^m-1} = (\Delta_{2^m-1,1}, \ldots, \Delta_{2^m-1,p})$.

Output: a permutation of subkey hypotheses

$$\pi = \begin{pmatrix} 0 & 1 & . & . & . & 2^m-1 \\ \tilde{k}_0 & \tilde{k}_1 & . & . & . & \tilde{k}_{2^m-1} \end{pmatrix}$$

sorted according to maximum absolute scalars of $\vec{\Delta}_0, \ldots, \vec{\Delta}_{2^m-1}$.

1: **for** k from 0 to $2^m - 1$ **do**
2: $\max_k = \max_{l \in \{1,\ldots,p\}} |\Delta_{kl}|$; {Maximum absolute scalar of the vector $\vec{\Delta}_k$ for key hypothesis k}
3: **end for**
4: Compute $\{\tilde{k}_0, \ldots, \tilde{k}_{2^m-1}\}$ by indexing $\{0, \cdots, 2^m - 1\}$ according to $\{\max_0, \cdots, \max_{2^m-1}\}$ such that $\max_{\tilde{k}_0} \geq \max_{\tilde{k}_1} \geq \ldots \geq \max_{\tilde{k}_{2^m-1}}$. {Sorting of key hypotheses}
5: return$(\{\tilde{k}_0, \ldots, \tilde{k}_{2^m-1}\})$

Algorithm 3.4 Vectorial T-Test

Input: m p-dimensional samples $\vec{x}_1, \ldots, \vec{x}_m$ and n p-dimensional samples $\vec{y}_1, \ldots, \vec{y}_n$.

Output: the p-dimensional T-test statistic $\vec{T}$.

1: **for** i from 1 to p **do**
2: Compute T_i using Eq. (2.7) given the m samples $(x_{1i}, \ldots, x_{mi})$ and the n samples $(y_{1i}, \ldots, y_{ni})$.
3: **end for**
4: Return$(\vec{T})$

The difference of means test of [73] is not an optimal choice, especially if the sample variance strongly differs within the trace. The T-Test of Theorem 2.9 is more appropriate and already proposed by [44, 6] to check for significant differences in means. Theorem 2.9 is applied for each component of the p-dimensional vectors of set $S_{k,0}$ and $S_{k,1}$ as shown in Algorithm 3.4. One may use or iterate other statistical tests that check whether differences in the sets $S_{k,0}$ and $S_{k,1}$ are significant or not,

e.g., one may use the F-Test of Theorem 2.10 to check for significantly different variances or apply non-parametric tests. Another approach was followed by [4]: The authors include both sample means and the sample variances into one single test statistic by considering a void hypothesis that is priorly estimated by using a random bifurcation of the traces [4].

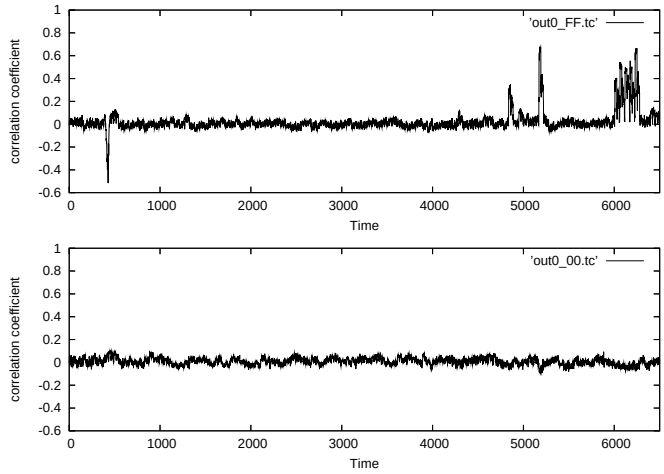


Figure 3.9: DSCA results of an AES implementation that is also used in Chapter 5 and Chapter 6. For this test, the Algorithm 3.5 was applied with the selection function on the 8-bit Hamming weight of one AES S-box outcome and 2000 measurement traces. The resulting traces of $\vec{\Delta}_{255}$ (correct subkey, upper plot) and $\vec{\Delta}_0$ (wrong key guess, lower plot) are shown. DSCA peaks are clearly visible in the upper plot.

Another test that is widely used in side channel literature is the correlation test. Here, the correlation coefficient (see Eq. (2.10)) of the outcome of the selection function and each scalar component of the power trace is computed. This statistical test can be placed in the generic DSCA algorithm of Algorithm 3.2, but this would yield to some unnecessary efforts in description, as one usually combines the selection function and the correlation test. The modified single-bit DSCA algorithm is a special case of Algorithm 3.5.

The use of *multiple-bit* DSCA was introduced by [94] and is built on the use of physical leakage models. One assumes that each bit of an in-

Algorithm 3.5 (Multi-bit) DSCA algorithm using the correlation method

Input: (i) Measurement data $(\vec{i}_1, \cdots \vec{i}_N)$ with $\vec{i}_j \in \mathbb{R}^p$ for all $j \in \{1, \ldots, N\}$, while the cryptographic device computes parts of a block cipher using the known data $x_j \in \{0,1\}^m$ and a secret subkey $k^\circ \in \{0,1\}^m$,
(ii) a selection function $d : \{0,1\}^m \times \{0,1\}^m \to \mathbb{R}$, and
(iii) an indexing algorithm $I : \underbrace{\mathbb{R}^p \times \cdots \times \mathbb{R}^p}_{2^m} \to P$ with P being the set of all permutations of the elements $\{0, \ldots, 2^m - 1\}$.

Output: a permutation of subkey hypotheses

$$\pi = \begin{pmatrix} 0 & 1 & . & . & . & 2^m - 1 \\ \tilde{k}_0 & \tilde{k}_1 & . & . & . & \tilde{k}_{2^m-1} \end{pmatrix}$$

such that $\mathbb{P}(\tilde{k}_0 = k^\circ) \geq \mathbb{P}(\tilde{k}_1 = k^\circ) \geq \ldots \geq \mathbb{P}(\tilde{k}_{2^m-1} = k^\circ)$.

1: **for** l from 1 to p **do**
2: **for** k from 0 to $2^m - 1$ **do**
3: Compute Δ_{kl} by using Eq. (2.10) with the N-dimensional sample $(i_{1l}, d(x_1, k)), \ldots, (i_{Nl}, d(x_N, k))$.
4: **end for**
5: **end for**
6: $\{\tilde{k}_0, \ldots, \tilde{k}_{2^m-1}\} \leftarrow I(\vec{\Delta}_0, \ldots, \vec{\Delta}_{2^m-1})$;
7: Return $(\{\tilde{k}_0, \ldots, \tilde{k}_{2^m-1}\})$;

termediate result contributes uniformly to the side channel leakage. The selection function outputs here the Hamming weight of a predicted intermediate result, e.g., the Hamming weight of the 4-bit outcome of the DES S-Box. Further, [94] presented *address based* DSCA by partitioning according to the predicted transition on an address bus. An approach to apply a selection function based on Hamming distance instead of the Hamming weight is called *Correlation Power Analysis (CPA)* [31]. Herein, one has to set up hypotheses for both the unknown subkey value k° and a fixed related data item x' which leads to an overall subkey space of $\{0,1\}^{2m}$ instead of $\{0,1\}^m$. A generic multiple-bit DSCA algorithm using the correlation method is given in Algorithm 3.5. One should remark that the outcomes of a multi-bit selection function are often bino-

mially distributed with parameter $n = m$ and $\theta = 0.5$ in Eq. (2.2). The binomial distribution converges to the normal distribution $N(\frac{m}{2}, (\frac{m}{4})^2)$ for large m, e.g., $m = 10$ can be seen as acceptable threshold for an approximation.

DSCA has been also applied at public key algorithms. In [95] *Zero-Exponent, Multiple Data (ZEMD)* was proposed. It predicts the Hamming weight of intermediate results of the modular exponentiation in Algorithm 3.1. For the use of DSCA techniques on an RSA-CRT implementation, a special attack called *Modular Reduction on Equidistant Data (MRED)* was demonstrated in [46].

3.3.5 Differential Collision Analysis

The use of an internal collision observed with side channel means was initially proposed by Hans Dobbertin. This class of attacks requires that one can choose the plaintext p_j (or the ciphertext c_j) of a cryptographic encryption (or decryption). Differential collision analysis aims at identifying when an internal sub-state of the cipher collides, e.g., for two different plaintexts p_j and $p_{j'}$ with $p_j \neq p_{j'}$. If such an internal collision occurs, one assumes that the resulting instantaneous side channel traces $\vec{i}_j$ and $\vec{i}_{j'}$ are very similar for that time frame of the internal collision which may take several clock cycles. The detection of a collision can be done, e.g., by applying the least-squares method [130], i.e., evaluating the squared difference within a time frame $[t_0, t_1]$ with $1 \leq t_0 < t_1 \leq p$:

$$R_{jj'} = \sum_{i=t_0}^{t_1} (i_{ji} - i_{j'i})^2$$

Small values of $R_{jj'}$ indicate that the traces $\vec{i}_j$ and $\vec{i}_{j'}$ are very similar in the time frame $[t_0, t_1]$, which in turn is seen as an indication of an internal collision. Each internal collision then provides a differential equation on the key bits and is equivalent to an entropy loss of a few key bits.

The first application was done at DES [132] and required three active S-boxes and 140 messages in average to observe a collision. Improved methods for DES and Feistel ciphers were given in [146, 77]. It was shown that the number of plaintexts can be reduced down to 8 for the DES by using only one active S-Box. On the AES, the use of internal

collisions was studied in [130]. For a survey on internal collision attacks, it is referred to [129].

3.3.6 Countermeasures

Since the invention of side channel attacks two different directions have been followed for providing implementation based countermeasures. A valuable survey and evaluation of circuit based countermeasures can be found in [87].

The first strategy aims at preventing any physical leakage as part of a cryptographic implementation. Such countermeasures are incorporated at the gate level and architecture level of an integrated circuit. At the gate level, some dynamic and differential logic styles are promising such as Sense Amplifier Based Logic (SABL) [140], however, at the cost of an area enlargement by a factor of 3.5 and a power enhancement of a factor of 4.5 [51]. The full custom layout needed is another drawback for SABL in practice. On the architecture level, one may add filters, noise generators, and random delays [87]. This helps in reducing the signal-to-noise level and therefore to increase the number of measurements for DSCA [41].

The second strategy aims at preventing any predictable internal state as part of a cryptographic implementation. This is achieved by using randomization techniques which typically requires an internal random number generator. Countermeasures can be implemented as part of the gate level, architecture level, and on the algorithm level. For the gate level, masked logic gates have been proposed, however, recent results [88, 51] have given evidence that the dissipated energy of nonlinear gates is correlated to the processed values whenever the input values do not arrive simultaneously at a gate. On the hardware architecture level one may mask the data representation used for bus transfers. If randomization is applied on the algorithmic level one calls it *masking* if applied at symmetric ciphers and *blinding* if used at asymmetric primitives. Masking countermeasures can be defeated, e.g., by second-order DSCA (see Section 3.3.7).

3.3.7 Second Order Differential Side Channel Analysis

Proposals for algorithmic defenses of symmetric ciphers on DSCA include boolean and arithmetic masking, e.g., [58, 39, 43]. Though defending on first-order DSCA, it turned out that implementations can still be vulnerable on *second-order DSCA* ([93] and [143]) which was demonstrated using single-bit selection functions at a boolean masking scheme. At second-order DSCA, the adversary analyzes the side channel measurements at two related points in time. [93] defined also an *nth-order DSCA attack* that makes use of n different instants corresponding to n different intermediate values calculated during the execution of an algorithm. Recently, [143] renewed this methodology.

For second-order DSCA, one usually considers a bit b (this may be a predicted outcome of a DSCA selection function based on known data x_j and key hypothesis k) and a random masking bit r that is uniformly distributed and not known by the adversary. Due to masking, bit b cannot be observed directly. However, the adversary can both observe the random bit r and the masked bit $r \oplus b$ in one trace. One usually now makes some assumptions on the leakage model of the cryptographic device, i.e., both the bit r and the masked bit $r \oplus b$ should cause a mean leakage portion of $+\epsilon$ if set to 1 and $-\epsilon$ if set to 0. Bit r is assumed to leak at time t_0 and bit $r \oplus b$ at time t_1. For the correct prediction of bit b, the joint distribution of the side channel leakage is given in Table 3.1. As it can be seen, for bit $b = 0$ the leakage portion at t_0 and t_1 has both the same sign, whereas for bit $b = 1$, the sign of the leakage portion is opposite.

Table 3.1: Possible states of bit r and $r \oplus b$ and the mean leakage portion at t_0 and t_1.

Bit b	Bit r	Bit $r \oplus b$	Mean Leakage of r at t_0	Mean Leakage of $r \oplus b$ at t_1
0	0	0	$-\epsilon$	$-\epsilon$
0	1	1	$+\epsilon$	$+\epsilon$
1	0	1	$-\epsilon$	$+\epsilon$
1	1	0	$+\epsilon$	$-\epsilon$

Single-bit second-order DSCA (2DSCA) can be seen as a special pre-

processing step that is carried out before common DSCA algorithms of Section 3.3.4, e.g., Algorithm 3.2 are applied. Pre-processing maps a trace $\vec{i} \in \mathbb{R}^p$ to a trace $\vec{i'} \in \mathbb{R}^{p'}$ so that the outcome $\vec{i'}$ combines instants at which the mask r and the masked value $r \oplus b$ may leak. *Zero-offset* and *known-offset* 2DSCA were proposed by [143]. Zero-offset 2DSCA is a special case of known-offset 2DSCA, whereat it is assumed that both the mask and the masked value are processed at the same point in time. While [143] suggests to use multiplication to combine both measurements, [93] uses the absolute value of the difference. The pre-processing algorithm for known-offset 2DSCA is given in Algorithm 3.6.

Algorithm 3.6 Pre-processing for known-offset 2DSCA

Input: (i) Measurement data $(\vec{i}_1, \cdots \vec{i}_N)$ with $\vec{i}_j \in \mathbb{R}^p$ for all $j \in \{1, \ldots, N\}$, while the cryptographic device computes parts of a block cipher using the known data $x_j \in \{0,1\}^m$ and a secret subkey $k^\circ \in \{0,1\}^m$,
(ii) Operation $op(a,b) \in \{a \cdot b, |a-b|, |a+b|\}$.
(ii) Data offset $s \in \mathbb{Z}$ with $0 \leq s < p$.
Output: Pre-processed measurement data $(\vec{i'}_1, \ldots, \vec{i'}_N)$ with $\vec{i'}_j \in \mathbb{R}^{p-s}$

```
1: for j from 1 to N do
2:    for l from 1 to p − s do
3:       i'_jl ← op(i_jl, i_j,l+s);
4:    end for
5: end for
6: return(i'⃗);
```

While [143] assumes that the grouping according to the value b is correct only for the correct key hypothesis, [93] mounts second-order DSCA at a linear part of the cipher and uses only two hypotheses on $b = k \oplus x$ depending on the value of one key bit k and one known data bit x. By doing so, the grouping is always correct, the decision strategy has to be adapted according to the sign of a DSCA signal. This typically requires that the side channel leakage is well known or reliably predictable for different bits at two instants in time.

Single-bit second-order DSCA has been extended to multi-bit second order DSCA. A theoretical analysis on the expected heights of DSCA signals based on multi-bit Hamming weight based leakage was done by

[67]. Experimental results can be found in [131, 129].

3.3.8 Multivariate Analysis

So far, Section 3.3.4 and 3.3.7 deal with univariate statistics for hypothesis testing. However, the physical observables are vectors and allow for the use of multivariate statistics. This section provides related work on multivariate analysis and links to the main Chapter 5 and 6 of this thesis.

Template Attacks were introduced by [40] and are said to be the strongest side channel attack possible from an information theoretic point of view. This attack requires both a profiling stage and a key recovery stage. The original publication [40] evaluated the application to the stream cipher RC4, the DES, and the modular exponentiation. Practical tests of the Template Attack were also done by [118].

At the profiling stage, for each key-dependency, a *template*, i.e., a multivariate characterization of the key dependent leakage signal, is produced. Let us follow a general approach, regardless of a specific kind of cryptographic algorithm, and assume K different key-dependent operations O_k with $1 \leq k \leq K$. During profiling, templates T_k, one for each key dependency O_k, are generated from a large number L of measurement traces taken for each key k. Each template represents a key dependent multivariate probability density and consists of two parts. The first part in a template is the sample mean $\vec{\mu}_k$ of all available traces representing the same key-dependency O_k. The second part in a template estimates the covariance matrix of the noise for the key-dependency O_k. Before starting to characterize the noise, it is highly advisable to identify and select a small set of time instants where the sample means $\vec{\mu}_k$ differ significantly in order to reduce computational and storage efforts. Reference [40] proposes to compute the sum of pairwise differences of the mean vectors, $\vec{\mu}_j - \vec{\mu}_l$ for $l \neq j$, and to choose M points $(P_1, \ldots, P_M)$ along the peaks of the resulting difference curve. One assumes that the noise in the side channel has approximately a multivariate normal distribution with respect to the selected instants. An M-dimensional noise vector is extracted from each measurement representing the template's key dependency O_k. Based on these L noise vectors one computes the $(M \times M)$ covariance matrix $\mathbf{C}_k$ for each key dependency k. The algorithm in the profiling stage is printed in Algorithm 3.7.

Algorithm 3.7 A generic Profiling Stage of the Template attack

Input: (i) Measurement data $(\vec{i}_{k_1}, \cdots \vec{i}_{k_L})$ with $\vec{i}_{k_j} \in \mathbb{R}^p$ for all $k \in \{1, \ldots, K\}$ and for all $j \in \{1, \ldots, L\}$ at the profiling device.
(ii) An instant selection algorithm $I : \underbrace{\mathbb{R}^p \times \cdots \times \mathbb{R}^p}_{K} \rightarrow P$ with P being the set of $M (M \leq p)$ chosen points of interest $\{P_1, \ldots, P_M\}$.
Output: (i) The set of points of interest $\{P_1, \ldots, P_M\}$ and
(ii) Templates $((\vec{\mu}_1, \mathbf{C}_1), \ldots, (\vec{\mu}_K, \mathbf{C}_K))$ built at $\{P_1, \ldots, P_M\}$ with $\vec{\mu}_k \in \mathbb{R}^M$ and $\mathbf{C}_k$ an $M \times M$ covariance matrix.

1: **for** k from 1 to K **do**
2: Compute the mean vector $\vec{\mu}_k = \frac{1}{L} \sum_{j=1}^{L} \vec{i}_{k_j}$;
3: **end for**
4: Select instants $\{P_1, \ldots, P_M\} \leftarrow I(\vec{\mu}_1, \ldots, \vec{\mu}_K)$;
5: **for** k from 1 to K **do**
6: **for** j from 1 to L **do**
7: Define the noise vector $\vec{n}_{k_j} \in \mathbb{R}^M$ by $\vec{n}_{k_j} = (n_{k_{j1}}, \ldots, n_{k_{jM}}) := (i_{k_{j,P_1}} - \mu_{k,P_1}, \ldots, i_{k_{j,P_M}} - \mu_{k,P_M})$;
8: **end for**
9: Define $\vec{\mu}_k \in \mathbb{R}^M$ by $(\mu_{k1}, \ldots, \mu_{kM}) := (\mu_{k,P_1}, \ldots, \mu_{k,P_M})$;
10: **end for**
11: **for** k from 1 to K **do**
12: **for** u from 1 to M **do**
13: **for** v from 1 to M **do**
14: Compute the covariance entries $\mathbf{C}_{k_{uv}} = \frac{1}{L} \sum_{j=1}^{L} n_{k_{ju}} n_{k_{jv}} - \left(\sum_{j=1}^{L} n_{k_{ju}}\right) \left(\sum_{j=1}^{L} n_{k_{jv}}\right)$;
15: **end for**
16: **end for**
17: **end for**
18: Return $((\{P_1, \ldots, P_M\}), (\vec{\mu}_1, \mathbf{C}_1), \ldots, (\vec{\mu}_K, \mathbf{C}_K))$;

If $\mathbf{C}_k$ is invertible, the probability density of the noise occurring under key dependency O_k is then given by the M-dimensional multivariate normal distribution $\text{prob}_{\mathbf{C}_k}(\cdot)$ where the probability of observing a noise vector $\vec{z}$ is

$$\text{prob}_{\mathbf{C}_k}(\vec{z}) = \frac{1}{\sqrt{(2\pi)^M \det(\mathbf{C}_k)}} \exp\left(-\frac{1}{2} \vec{z}^T \mathbf{C}_k^{-1} \vec{z}\right), \quad \vec{z} \in \mathbb{R}^M, \quad (3.1)$$

$\det(\mathbf{C}_k)$ denotes the determinant of $\mathbf{C}_k$, and $\mathbf{C}_k^{-1}$ its inverse.

Algorithm 3.8 Key Recovery at the Template attack

Input: (i) Measurement data $\vec{i} \in \mathbb{R}^p$ using the unknown key $k^\circ \in \{1, \ldots, K\}$ at the target device.
(ii) The set of points of interest $\{P_1, \ldots, P_M\}$.
(iii) Templates $((\vec{\mu}_1, \mathbf{C}_1), \ldots, (\vec{\mu}_K, \mathbf{C}_K))$ built at $\{P_1, \ldots, P_M\}$ with $\vec{\mu}_k \in \mathbb{R}^M$, $\mathbf{C}_k$ an $M \times M$ covariance matrix, $\mathbf{C}_k^{-1}$ its inverse and $\det(\mathbf{C}_k)$ the determinant of $\mathbf{C}_k$ for each $k \in \{1, \ldots, K\}$.
(iv) A standard indexing (in descending order) algorithm $I : \underbrace{\mathbb{R} \times \cdots \times \mathbb{R}}_{K} \rightarrow P$ with P being the set of all possible permutations of the elements $\{1, \ldots, K\}$.
Output: a permutation of key hypotheses

$$\pi = \begin{pmatrix} 1 & . & . & . & K \\ \tilde{k}_1 & . & . & . & \tilde{k}_K \end{pmatrix}$$

such that $\mathbb{P}(\tilde{k}_1 = k^\circ) \geq \mathbb{P}(\tilde{k}_2 = k^\circ) \geq \ldots \geq \mathbb{P}(\tilde{k}_K = k^\circ)$.

1: Define the vector $\vec{i'} \in \mathbb{R}^M$ by $= (i'_1, \ldots, i'_M) := (i_{P_1}, \ldots, i_{P_M})$;
2: **for** k from 1 to K **do**
3: $\quad \vec{z} = \vec{i'} - \vec{\mu}_k$;
4: $\quad p_k = \frac{1}{\sqrt{(2\pi)^M \det(\mathbf{C}_k)}} \exp\left(-\frac{1}{2}\vec{z}^T \mathbf{C}_k^{-1} \vec{z}\right)$;
5: **end for**
6: $\{\tilde{k}_1, \ldots, \tilde{k}_K\} \leftarrow I(p_1, \ldots, p_k)$;
7: Return $(\{\tilde{k}_1, \ldots, \tilde{k}_K\})$;

The strategy to classify a single measurement $\vec{i} \in \mathbb{R}^p$ is a maximum likelihood hypothesis test (see Algorithm 3.8). For each hypothetical key dependency O_k, one extracts the noise in $\vec{i}$ by subtracting the mean $\vec{\mu}_k$ at the M selected instants yielding the noise vector $\vec{z}$. One then computes the probability $\text{prob}_{\mathbf{C}_k}(\vec{z})$ to observe such a noise vector using (3.1). The hypothesis O_k maximizing (3.1) is then the best candidate for the observed key dependency.

3.4 Fault Channel Cryptanalysis

Fault channel cryptanalysis uses physical means in order to modify internal states of an implementation, aiming at obtaining additional information for cryptanalysis. A fault is defined as an abnormal condition or defect of a component which may lead to a failure. In integrated circuits a fault may be an unintentional short-circuit, or partial short-circuit, between energized conductors or an energized conductor and ground.

The underlying hypothesis for fault channel cryptanalysis is that the internal state S of a cryptographic device is modified to an internal state S' as result of physical interaction processes applied, e.g., while the device computes a cryptographic function. The transition from S to S' is seen as an implementation dependent probabilistic event constituting the *fault channel*. The task of fault channel cryptanalysis is to recover the internal state S by examining the effects of the erroneous state S' propagating through the computation.

Fault analysis of cryptographic primitives has become a new research area initiated by [28]. A recent survey on fault analysis of cryptographic primitives can be found in [16]. As part of this section the notion of a *security service* is used as a term for any security relevant or security enforcing building block of the cryptographic device. Informally speaking, an adversary is *successful* if the insertion of faults either i) yields access to a security service without knowledge of the required secret or ii) yields partial information about the secret.

Figure 3.10 illustrates the principle instrument set-up for fault channel cryptanalysis. Photographies of advanced set-ups used in practice can be found, e.g., in [14].

This section on fault channel cryptanalysis is structured similarly to the previous section on side channel cryptanalysis. Its focus is on techniques for fault induction in Section 3.4.2 and afterwards on the most relevant analysis methods to provide some basics needed for Chapter 7.

3.4.1 Refinements of Adversary Model

Considering fault channel cryptanalysis one assumes that the adversary is able to cause a change of internal states of the targeted cryptographic implementation. Fault channel cryptanalysis becomes a cryptanalytical problem if the cryptographic device continues computing based on an

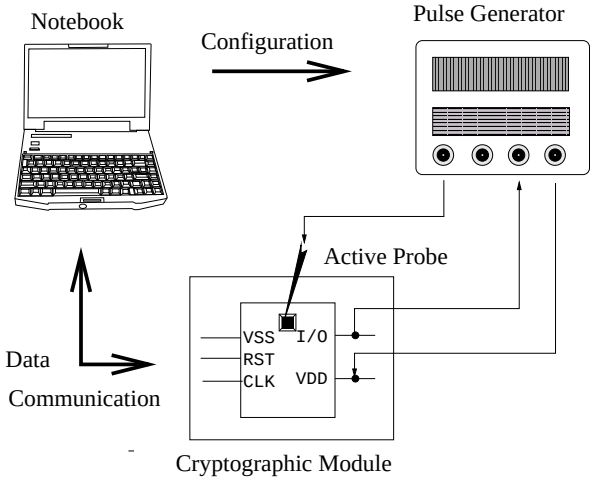


Figure 3.10: Principle of a fault injection instrument set-up for a glitch attack and an active probe connected to internals of the circuit.

erroneous state S' and outputs an erroneous cryptogram. Otherwise the fault may be ineffective for cryptanalysis and one needs auxiliary means, e.g., side channel cryptanalysis. The implementation based attack parameters introduced in Section 3.2.1 are refined in the context of fault channel cryptanalysis.

- *Physical access:* Indirect physical access means that an adversary cannot directly apply short-range physical effects to the cryptographic device and may only change environmental conditions or wait for an accidental fault.
- *Laboratory equipment:* Laboratory equipment means instruments for invoking a physical interaction process in the cryptographic device, as detailed in Section 3.4.2. A valuable survey can also be found in [14].
- *Models and algorithms:* The task of fault channel cryptanalysis is a cryptanalytic problem on the basis of erroneous cryptograms that were obtained as result of fault analysis. Assumptions on the fault model are usually needed for the choice of the algorithms for the cryptanalytic part of the attack.

- *Number of cryptographic devices* The number of cryptographic devices may become a crucial parameter because the physical interaction process may cause the destruction of the cryptographic device.

3.4.2 Techniques for Fault Induction

In normal use, faults occur accidentally in integrated circuits, e.g., due to the wear lifespan of components. Effects that cause aging defects are hot carriers, electromigration, and radiation [100]. Hot carriers are high energetic electrons that cross the potential barrier of the gate oxide because of the so-called tunnel effect. These charges can lead to changes of the threshold voltage of gates. Electromigration is caused by high current densities yielding to deformations of the circuit lines. Such effects can have an impact on the wiring resistance. Radiation induced aging originates by the insertion of charged particles into the circuit. Such effects are also known as *single event upsets* (SEUs) and were noticed first in a space mission in the seventies [14]. A model on reliable cryptographic key lifetimes under 'normal' environmental conditions is provided in [59]. Accidental faults are also known as 'passive fault attacks'.

The sensitivity towards faults and their frequency can be enhanced by external physical means, referred to as *fault injection techniques*. Fault injection aims to cause an interference with the physical implementation and to enforce an erroneous behavior of the implementation.

Physical means can be classified in

- environmental physical means,
- semi-invasive physical means, and
- invasive physical means.

Environmental Physical Means

The importance of protection against environmental fault injection techniques is, e.g., already included in the requirements of [105] as stated below.

> "Deliberate or accidental excursions outside the specified normal operating ranges of voltage and temperature can cause erratic

> operation or failure of the electronic devices or circuitry that can compromise the security of the cryptographic module."

Integrated circuits are specified for a certain range of temperature to guarantee a reliable operation. Faults are more likely to occur outside of this specified range. The effect of 'freezing volatile data memory' (e.g., [144, 136]) can already be observed at temperatures of about $-20°C$. Reference [136] reports that the data retention varies widely, even among devices from the same type. These contributions aim at reading out the data memory even after an activation of a zeroization circuitry (cf. Section 3.1.5). Nevertheless, it can be assumed that freezing may also lead to errors during computations. Overheating of the chip is an alternative attempt which leads to unforeseen effects, e.g., in the CPU.

So-called *glitches* can be injected both by variations of the external clock or the external supply voltage. According to [144], lengthening or shortening of the clock pulses to a clocked circuit such as a microprocessor can subvert its operation. Instructions or tests can be skipped or erratic operation can be induced [10].

Similarly, faults can be induced in the circuit by changing the supply voltage to abnormally high or low values. The erratic behaviour may include the processor misinterpreting instructions, erase or over-write circuitry failing, or memory retaining its data when not desired [144].

In [144] and recently in [17] it has been reported that it is possible to read and write storage cells by using an infrared (IR) laser directed through the bulk silicon side of the chip. This is feasible as silicon is transparent at IR frequencies.

Semi-invasive Physical Means

Semi-invasive fault induction typically requires the depackagaging of the chip. In [137] *optical fault induction* was introduced. Fault injection on SRAM memory is done by illuminating it with a light source such as a photoflash, or alternatively, a laser. Reference [137] mounts the photoflash on the video port of a manual probing station. The SRAM is shielded with an aperture made from aluminum foil. Such an aperture allows to expose only one memory cell to the light and leads to a precise area resolution. The basic interaction mechanism of photons with atoms in semiconductor devices is the photoeffect [7]. The photoeffect leads to the absorption of a photon by the atom which in turn emits

an electron. Basically, the photoeffect generates free charges inside the semiconductor. Optical fault induction is called a semi-invasive attack, as the packaging of the chip has to be removed.

Another semi-invasive attack is presented by [121] that uses a miniature coil supplied with an alternating current. The change of the magnetic field induced causes eddy currents in the chip plane which have an effect, e.g., on the data signals and provoke faults. In comparison to optical fault injection, the area resolution achievable by this kind of electromagnetic induction is less precise [122].

Semi-invasive fault induction using x-rays would be another alternative. To the knowledge of the author, there are no publications up to now. A possible drawback might be the availability and handling of the sources. Further, the interaction with matter is different, as the Compton effect [7], i.e., an elastic scattering process with electrons, becomes the dominant interaction process. Energy is transferred to the electrons and leads to a reduction of the photon's frequency. However, [144] reports that X-ray radiation can imprint CMOS RAM.

Invasive Physical Means

Invasive fault induction requires the depackaging of the chip and additionally the removal of the passivation on top of the chip. In some scenarios, the circuit is directly probed. In [144] it was reported that energy probes can read or write the contents of semiconductor storage, or change control signals. Especially, the electron beam of a conventional scanning electron microscope can be used to read, and possibly write, individual bits in an EPROM, EEPROM, or RAM [144]. Impacts of heavy energetic particles like cosmic rays is known to cause single event upsets, e.g., [83, 84]. Such effects have already been studied in the development of semiconductor devices for space and high energy physics [66].

Invasive attacks are mounted internally, within the integrated circuit. Tools for this kind of modifications are, e.g., focused ion beams, as demonstrated in [60]. Using such tools, complex modifications are feasible which can completely modify the internal construction. These are the most efficient tools for failure analysis of integrated circuits. Further, [144] reports that active probes can inject signals or information into an active system, provided that the active probe has direct electrical

contact to the internal circuitry.

3.4.3 Modelling of Faults

For modelling, assumptions on the kind of faults as well as their consequences are needed. The first publication [28] considers a random transient fault model, i.e., "from time to time the hardware performing the computation may induce errors". The origin of the error is seen as a deficiency of the hardware that occurs by chance without any active enforcement by the adversary. Reference [24] introduces an asymmetric type of memory faults, so that changing one bit from zero to one (or the other way around) is much more likely. Further, Simple Fault Analysis (SFA) based on a permanent destruction of a memory cell is found in [24]. Reference [9] and [98] give examples of processing faults leading to skipping an instruction, e.g., aiming at breaking out of a loop that checks a password or a PIN. Other publications make use of separate fault models on a bit and byte level, e.g., in [27]. A taxonomy on different types of faults can be found in [14].

A fault is said to be *transient* if the device remains fully functional after the fault injection. A fault is called *permanent* if the device stucks, i.e., the fault persists during the life time of the device. If the fault affects data, it is said to be a *data fault*. If the fault affects the processing of the device it is called a *processing fault*.

Remark 3.1. Note that processing faults may jeopardize any security service.

It is said that a data fault has a *preferred direction* if the probability to cause a bit transition from '0' to '1' is significantly different from the probability to cause a bit transition from '1' to '0'.

3.4.4 Simple Fault Analysis

In this category, most variants of fault analysis attacks can be found. Besides scenarios on cryptographic primitives there are other applications of fault induction that target generic (non-cryptographic) building blocks of a security service. The following section provides an overview on methods of Simple Fault Analysis (SFA).

The 'Bellcore' Attack on the RSA-CRT implementation

The first theoretical attack was brought up on an RSA-CRT implementation and requires only one fault injection with very low requirements on the concrete fault occurrence [28].

The RSA cryptosystem is given by the secret RSA primes p and q, the public modulus $N = p \cdot q$, the public exponent e and the secret exponent d with $e \cdot d \equiv 1 \bmod (p-1)(q-1)$. The RSA-CRT algorithm is given in Algorithm 3.9.

Algorithm 3.9 RSA-CRT algorithm

Input: RSA primes p and q with $N = p \cdot q$, secret exponent d, message x
Output: $y = x^d \bmod N$
1: $d_p = d \bmod (p-1)$;
2: $d_q = d \bmod (q-1)$;
3: $p_q = p^{-1} \bmod q$;
4: $q_p = q^{-1} \bmod p$;
5: $v_p = x^{d_p} \bmod p$;{First modular exponentiation}
6: $v_q = x^{d_q} \bmod q$;{Second modular exponentiation}
7: $y = (v_p \cdot q \cdot q_p + v_q \cdot p \cdot p_q) \bmod N$; {Gauss Recombination}

Assume that the first CRT exponentiation of Algorithm 3.9 outputs a wrong result $v'_p \neq v_p$ as result of a fault. Recombination then yields

$$y' = (v'_p \cdot q \cdot q_p + v_q \cdot p \cdot p_q) \bmod N$$

Further assume that a correct result of the RSA-CRT computation using the same value x is available. Then

$$y - y' = v'_p \cdot q \cdot q_p - v_p \cdot q \cdot q_p = q \cdot q_p \cdot (v'_p - v_p)$$

is a multiple of prime q. Application of the greatest common divisor (gcd) algorithm yields the factorization of N:

$$q = gcd(N, y - y');$$

A variant of this attack according to Arjen Lenstra [28] suffices with only one faulty RSA-CRT computation, provided that the public exponent e is known

$$q = gcd(N, (x - (y')^e) \bmod N).$$

The importance of this so-called 'Bellcore-Attack' lies in the fact that it does not require any assumption on the kind of error induced into the computation of one exponentiation. As exponentiation is costly in terms of time and chip area this also does not require a good precision for fault injection. A practical demonstration of the 'Bellcore-Attack' is given in [12].

Other Attacks on Cryptographic Primitives

Modular exponentiation, which is used for RSA as well as ElGamal, Schnorr and DSA signature schemes, can be also attacked by successive fault injections [28]. Another attack tampers with the DSA nonce so that a number of its least significant bits will flip to zero [99]. After obtaining a few tens of faulty signatures the private signature key can be recovered by lattice reduction. Very recently a new key recovery attack was presented in [29] that only corrupts the public RSA modulus.

For stream ciphers, [62] presents fault analysis techniques targeting the linearity of LFSRs which are typical building blocks. Another approach has been presented in [22] for the stream cipher RC4 exploiting the forced induction of impossible states.

A Generic Attack on Asymmetric Memory

A generic attack on cryptographic keys is the following one: If the memory type used for key storage has the special property that flipping a bit from one state to the other is impossible (e.g., from state '1' to state '0'), all key bits will finally accumulate in one state (e.g., state '1') after repetitive fault injections. Assuming the adversary owns cryptograms for each intermediate state, e.g., after each successively induced state transition, the adversary can iterate backwards recursively [23], starting at the known frozen state, yielding finally the original key value.

Defeating Side Channel Countermeasures

A simple proposal to prevent simple side channel analysis of the exponentiation in Algorithm 3.1 is to include 'dummy' multiplications if the exponent bit is zero. In the presence of such an implementation, however, it is possible to induce faults during the exponentiation producing

the probabilistic information 'error detected' or 'no error detected' thus being an oracle leaking information on the secret key successively [68].

Attacks on non-cryptographic Building Blocks

Other fault scenarios aim at infecting control variables or program flow.

- Modification of security states: For cryptographic devices, it is necessary to maintain security states by storing attributes, e.g., related to authorizations and privileges achieved. A fault injection at such a security state may end up in a more privileged state.

- Modification of a security service: Modification of a security service itself can be invoked by fault injection. Bypassing checks of parameter bounds as presented in [9] and [98] is one example for this kind of threat.

- Denial of service: Fault injection can result in a permanent malfunction or destruction of circuit components used by a security service. For example, the destruction of a physical random number generator might be an attractive target.

3.4.5 Differential Fault Analysis

Differential attacks on block ciphers require both a correct cryptogram and some faulty ones for the analysis. In [24], Differential Fault Analysis (DFA) is introduced on DES. The original attack model assumes that faults occur randomly in all rounds of DES and requires about 50-200 faults in this model. If precise fault injection is possible, the number can be reduced to about three faults [16]. For AES, some scenarios are presented in [16] of which the most promising one [114] requires two faults for recovering the AES key.

The following explanations sketch the attack on the DES (see Figure 3.8). Assume that the adversary is able to run a DES implementation twice with the same plaintext and receives the ciphertext. Assume further that the first run yields the correct result and that a fault injection was successful to toggle a few bits in R_{15} at the beginning of the last DES round in the second run. Because of $R_{15} = L_{16}$ the adversary knows

which bits are corrupted in the second run. Let denote the correct values with R_{15} and L_{16} and the corrupted values with R'_{15} and L'_{16}. It is

$$R_{16} = L_{15} \oplus f(R_{15}, K_{16})$$

and

$$R'_{16} = L_{15} \oplus f(R'_{15}, K_{16}).$$

Performing an exclusive-or of the both equations leads to

$$R_{16} \oplus R'_{16} = f(R_{15}, K_{16}) \oplus f(R'_{15}, K_{16}).$$

Except for K_{16} all variables are now known. One obtains an equation for the bits of K_{16}. More precisely, as shown in Figure 3.8, this reduces to an equation for all affected S-boxes on the basis of subkeys, i.e., if the difference $R_{16} \oplus R'_{16}$ at the outcome of one S-box is non-zero, one obtains an equation for the 6-bit subkey entering one S-box. Let Δ_y be the 4-bit differential of $R_{16} \oplus R'_{16}$ for one of the S-box outputs. Let x be the original bits of R_{15} and x' the corrupted bits of R'_{15} for the input to the same S-box. The differential equation is then

$$\Delta_y = S(x \oplus k) \oplus S(x' \oplus k).$$

The adversary tests all 2^6 possible subkey values k and is able to reject subkey hypotheses that do not fulfill this equation for the respective S-box. In practice, repetition of this attack leads to further sieving of subkey candidates, finally yielding all bits of K_{16}.

3.4.6 Countermeasures and Further Directions

Proposals for defenses of fault channel cryptanalysis include hardware and software countermeasures. Reference [14] gives a valuable survey on possible countermeasures that is summarized here. Active circuit based countermeasures are light detectors, supply voltage detectors, frequency detectors, active shields, and various redundancies of building blocks operating in parallel or in sequence. Passive circuit based countermeasures are similar to the ones on side channel attacks, i.e., the insertion of dummy cycles, bus and memory encryption, passive shields, and unstable internal frequency generators. Software countermeasures include well-known checksums applied to data memory contents, randomizations

of the program execution sequence, redundancy of variables, redundancy of program execution and baits (small code fragments performing checks) associated with ratification counters. For more details on countermeasures refer to [14].

Though typically defending on first-order fault channel cryptanalysis, secured implementations can be still vulnerable if multiple fault injections are performed in parallel. Literature on higher-order attacks, especially on any practical results, is not available, yet. However, it appears reasonable that concepts analogous to side channel cryptanalysis on second-order analysis and multivariate analysis will also work for fault channel cryptanalysis. Similarly to differential collision analysis described in Section 3.3.5, a new class of collision attacks based on fault induction was already invented by [26].

Chapter 4

Differential Side Channel Analysis on Boolean and Arithmetic Operations

4.1 Contribution

Differential Side Channel Analysis (DSCA) has turned out to be an efficient toolbox and has been well studied for ciphers that incorporate a nonlinear substitution box as, e.g., in DES and AES.

Other product ciphers and message authentication codes using one-way hash functions are based on the mixing of different algebraic groups and do not use look-up tables. Among these are IDEA [76, 92], the AES finalist RC6 [120], and hash based constructions for message authentication codes (MACs) such as HMAC-SHA-1 and HMAC-RIPEMD-160 [20, 19].

Algorithms that do not incorporate small S-box tables and use algebraic functions instead restrict the use of the DSCA selection function to the Hamming weight and Hamming distance of boolean and arithmetic operations. Whereas S-box tables are sufficiently non-linear and have uniform distributions of output bits for all key values, this, however, is in general not the case for algebraic constructions. Due to their linearity, secondary DSCA peaks occur at related but wrong key hypotheses. Further, for boolean and arithmetic operations, the number of key hypotheses is often directly related to the number of bits used for the DSCA selection function. For these algorithms, multi-bit selection functions offer an improved trade-off between the number of key hypotheses and the number of DSCA calculations compared to single-bit selection functions. Moreover, the use of single-bit selection functions can require detailed information on the implementation [38], i.e., one needs to know the sign of the side channel contribution for each bit. By using multi-bit selection functions key identification is made easier as ambiguities are expected only for a small number of key hypotheses.

This chapter is a revised extension of the work published in [81] and provides a well-founded analysis of DSCA on product ciphers and message authentication codes based on arithmetic and boolean operations. Therefore, DSCA signals resulting from n-bit sized primitive operations such as XOR, addition modulo 2^n, and modular multiplication are studied using multi-bit DSCA selection functions. This analysis makes use of simulating measurement data. It is shown that the DSCA characteristics differ for these basic operations. This fact can support side channel analysis of an unknown implementation and even for an unknown cipher.

The theoretical approach to apply DSCA to ciphers and message

authentication codes based on primitive operations is included, as there are the specific examples of IDEA, RC6 and the HMAC construction based on RIPEMD-160 and SHA-1.

Experimentally, both an IDEA implementation on an 8051 microcontroller and on an AVR ATM163 microcontroller were analyzed. The Hamming weight model was successfully applied at the primitive operations for both architectures and the expected DSCA characteristics were basically confirmed. Whereas the physical leakage of the 8051 microcontroller can be well approximated by the Hamming weight model, one observes more difficulties in case of an AVR microcontroller. As result, it has to be noted that the Hamming weight model is only partially useful for DSCA at an AVR microcontroller and the physical leakage of an AVR microcontroller is more complex and not well approximated by simple Hamming weight or Hamming distance models.

For physical devices which cannot be well approximated by simple leakage models such as the Hamming weight model, it is essential to test the underlying hardware thoroughly in order to optimize DSCA. Methodical approaches to determine the physical leakage are subject of the following chapters Chapter 5 and Chapter 6. These approaches finally help in understanding and modelling physical leakage.

4.2 Previous Work

Since 1998 it is known that Simple Side Channel Analysis (SSCA) and Differential Side Channel Analysis (DSCA) can be applied to extract cryptographic keys by measuring the instantaneous physical leakage of the cryptographic module during the processing of a cryptographic algorithm [73]. Early investigations have been focused on the DES Feistel scheme. For the AES candidates the key whitening process was studied using bitwise key hypotheses [38]. Algorithms that are based on the mixing of different algebraic groups as IDEA and RC6 are theoretically treated in [38] and [110], but not deeply studied in practice, yet.

Software countermeasures to secure cryptographic algorithms with arithmetic and boolean operations turn out to be costly for the conversion algorithm from arithmetic to boolean masking [57, 45]. In constrained environments these performance costs might not be acceptable for iterated product ciphers, so it may be assumed that DSCA remains

an issue.

DSCA as introduced in Section 3.3.4 turned out to be a very efficient side channel attack that makes use of a statistical analysis of the side channel at run-time of a cryptographic algorithm. DSCA requires the knowledge of either the plaintext or the ciphertext as a pre-condition.

Side channel analysis exploits the dependency of the physical leakage by the hardware on the value of intermediate data. The adversary knows or assumes a model for this dependency. Two types of leakage have been confirmed which are caused by the Hamming weight and by the Hamming distance of data (see Section 3.3.3). The Hamming distance model considers the dynamic dissipation due to the number of gates that change the state during a transition. Difficulties may evolve if the Hamming distance corresponding to transition counts is the dominant source for the observed leakage. In this case the selection function should be adapted to the implementation and eventually even restricted to a certain time frame. In case of microcontrollers based on the von Neumann architecture (shared data/address bus), the Hamming distance to an additional unknown reference variable might be incorporated for the optimization of DSCA results [30, 31]. In case of an Harvard architecture, DSCA results depend on the concrete sequence of instructions and registers used by the implementation. In [30, 31] it was reported that Correlation Power Analysis (CPA) is appropriate to get rid of so-called 'ghost peaks', i.e., DSCA peaks that occur at wrong key hypotheses.

The choice of the DSCA key hypotheses and selection functions depends on the cryptographic algorithm and the implementation to be attacked. In case of DES and AES the preferred selection functions focus on S-box look-ups. Both the address of the S-box table look-up, i.e., the S-box input and the S-box output can leak information. In case of DES, a selection function targeting one S-box access makes use of 6-bit key hypotheses. In case of the AES there are 8-bit key hypotheses per S-box. The selection function can either use a single bit or multiple bits of intermediate data.

DSCA identifies the correct key value by statistical methods for hypothesis testing. An attacker does not need to know details of the implementation as DSCA points itself to the relevant points in time. Suitable tests are the 'distance-of-means' test, the student's T-Test and the correlation method (cf. Section 3.3.4).

4.3 DSCA using n-bit sized Basic Operations

Each operation that is considered below is carried out between a known n-bit variable x and a secret n-bit variable k°. As the assumption for DSCA, x is known and follows the uniform distribution while k° is a secret, constant value.

k° and x can be represented as the concatenation of m-bit ($m \leq n$) blocks: $k^\circ = k^\circ_{n/m-1}|k^\circ_{n/m-2}|...|k^\circ_1|k^\circ_0$ and $x = x_{n/m-1}|x_{n/m-2}|...|x_1|x_0$. A common choice may be $m = 8$ for an 8-bit microcontroller. Let define

$$k^\circ_j = (k^\circ \bmod 2^{(j+1)m}) \operatorname{div}(2^{j \cdot m})$$

and

$$x_j = (x \bmod 2^{(j+1)m}) \operatorname{div}(2^{j \cdot m})$$

where $j \in \{0, ..., n/m - 1\}$. In the following, the index j is the block number of a n-bit sized variable.

The key hypotheses are set up on each value of k°_j. There are 2^m key hypotheses H_{jk}, namely for each j

$$H_{jk} \text{ is } \{k^\circ_j = k\}$$

where $k \in \{0, ..., 2^m - 1\}$. From now on, the index k is the key hypothesis for a certain value k°_j.

The selection function is defined by the Hamming weight $\mathcal{W}$ of an intermediate m-bit result of any primitive operation $x \circ k$, wherein $\circ$ marks the actual operation used.

$$d_j(x, k) = \mathcal{W}((x \circ k)_j) - \mathbb{E}_{X,k,j}(\mathcal{W}((X \circ k)_j))$$

with

$$\mathbb{E}_{X,k,j}(\mathcal{W}((X \circ k)_j)) = \sum_{x \in \{0,1\}^n} \text{Prob}(X = x)\mathcal{W}((x \circ k)_j)$$

being the expectation of the Hamming weight for $x \circ k$ using a summation of all possible input values x while fixing k and j. Assuming a uniform distribution of X one obtains $\text{Prob}(X = x) = \frac{1}{2^n}$. Group operations that are bijective and show a uniform distribution of the resulting bits lead to $\mathbb{E}_{X,k,j}(\mathcal{W}((X \circ k)_j)) = m/2$.

If the side channel leakage is dominated by the Hamming distance the selection function is modified to

$$d_j(x,k) = \mathcal{W}(z_j \oplus ((x \circ k)_j)) - \mathbb{E}_{X,k,j}(\mathcal{W}((X \circ k)_j))$$

where z_j is an additional data item (which can be either constant or random, known or secret) that is in conjunction with the predecessor or successor, the secret intermediate value to be attacked. If z_j is zero, the Hamming weight model is revealed as a special case of the Hamming distance model. The application of the Hamming weight model for DSCA while in reality the implementation causes a leakage according to the Hamming distance model can lead to erroneous results, e.g., if $\circ$ is the XOR operation and z_j is a constant non-zero value, DSCA will point to $(z_j \oplus k_j^\circ)$ as the correct key value. Note, that in the case where z_j is a random secret value, (first order) DSCA will fail. Generally, the Hamming distance model requires a more detailed analysis of the implementation if compared to the Hamming weight model.

In the original work of [81] it was suggested to neglect single measurements for DSCA if the selection function $d_j(x,k)$ is zero for certain values of x, otherwise they are weighted according to the result of the selection function. This multi-bit approach is different to the one of [94] which suggested to use only measurements with the extreme results of $d_j(x,k)$, namely $m/2$ and $-m/2$, which, however, results in highly increased measurement costs. Using the work of [81], only $\binom{m}{m/2}$ single measurements are discarded. However, this thesis follows the approach already given in Algorithm 3.5, i.e., it computes the correlation coefficient for all single measurements, regardless of the outcome of $d_j(x,k)$ and gives only in Section 4.3.1 a comparison between this work and the work of [81]. The choice of this thesis is motivated by the fact, that (i) the expected height of the correlation coefficient is more easy to understand in special cases and (ii) the relative difference of the correlation coefficient between related key candidates is enhanced by using Algorithm 3.5.

DSCA tests for significant statistical side channel differences of single measurements with positive and negative values of $d_j(x,k)$. According to [94] it is assumed that the data dependent side channel difference

$$\Delta i_t(x_i, k_j^\circ) = i_t(x_i, k_j^\circ) - \frac{1}{N}\sum_{i=1}^{N} i_t(x_i, k_j^\circ)$$

is proportional to the Hamming weight of processed data. Herein, N denotes the overall number of measurements with randomly chosen inputs x_i for $1 \leq i \leq N$ and k_j° is the j-th unknown secret key block. This difference $\Delta i_t(x_i, k_j^\circ)$ is the signal to be detected.

The DSCA results presented here were produced by using the correlation method in Algorithm 3.5, i.e., the correlation coefficient is computed as

$$c_{j,t}(k) = \frac{\sum_{i=1}^{N} d_j(x_i, k)\ \Delta i_t(x_i, k_j^\circ)}{\sqrt{\sum_{i=1}^{N} d_j(x_i, k)^2}\sqrt{\sum_{i=1}^{N} \Delta i_t(x_i, k_j^\circ)^2}}$$

for all key candidates $k \in \{0, \ldots, 2^m - 1\}$. The correlation coefficient $c_{j,t}(k)$ is near zero if the selection function $d_j(x_i, k)$ and $\Delta i_t(x_i, k_j^\circ)$ are uncorrelated. In case of an ideal correlation $c_{j,t}(k)$ approaches 1 or -1 at some t. The decision is made for the key k' defined by

$$k' = \arg \max_{k \in \{0,1\}^m} |c_{j,t}(k)| \ .$$

The following subsections exclusively deal with the assumption of the Hamming weight model. The selection functions are to be modified if the Hamming distance is the major source of correlation signals.

In this chapter, some DSCA results were gained by using simulated data in a Hamming weight model. The stochastic simulation is carried out by randomly choosing input data x for a fixed k° and outputting simulated side channel traces based on the Hamming weight of intermediate processed data, i.e., outputting $\mathcal{W}((x \circ k^\circ)_j)$. These simulated traces are then worked on with the DSCA procedures that are also used for experimental data. Simulation offers the benefits of fast data generation resulting from a given leakage model and due to its versatility it helps in understanding the impact of a model. Simulation results are generated by assuming an 8-bit or 16-bit Hamming weight. Experimental analysis was done on 8-bit microcontrollers.

4.3.1 Boolean Operation XOR

XOR is the most widely used boolean operation in cryptographic algorithms. The selection function is

$$d_j(x, k) = \mathcal{W}(x_j \oplus k) - m/2 \ .$$

Note that if $m = 1$ one has (only) two hypotheses on the value of a key bit. If one observes a DSCA peak, the peak is present for both key hypotheses, but with opposite sign. The interpretation of the results obtained directly depends on the leakage model for the one bit under question (see Table 4.1).

Table 4.1: DSCA result of bit $b = (x \oplus k^\circ)_j$. $\nearrow$ means that the leakage increases with the value of b, $\searrow$ means that the leakage decreases with the value of b.

Bit b	Leakage contribution of b	Sign of DSCA peak
0	$\nearrow$	negative
1	$\nearrow$	positive
0	$\searrow$	positive
1	$\searrow$	negative

The correlation coefficient of $d_j(x_i, k)$ and the side channel $\Delta i_t(x_i, k_j^\circ)$ reaches the maximum if k equals k_j° and the minimum if k is $\neg k_j^\circ$. The absolute value of the correlation coefficient for both cases is the same. If the side channel increases with the Hamming weight (which is often seen as the normal case), the correct key hypothesis has a positive correlation coefficient; otherwise the correlation coefficient is negative. If the attacker does not know the sign of the linear dependency, a small brute-force analysis has to be applied. Besides the correct value and its bitwise inverted value, less significant correlation coefficients occur at other key hypotheses that differ by less than $m/2$ bits regarding the correct key hypothesis or the bitwise inverted key hypothesis. Key hypotheses that differ by $m/2$ bits are uncorrelated. The number of key hypotheses that differ by w bits regarding a certain correct key hypothesis is given by the binomial coefficient $\binom{m}{w}$.

Figure 4.1 shows exemplary results by applying Algorithm 3.5 and $m = 8$ on simulated measurement data. Here, one obtains correlation coefficients of approximately ± 1, ± 0.75, ± 0.5, ± 0.25, and 0. In contrast, by applying the algorithm of [81] that disregards samples for which the selection function $d(\cdot, \cdot)$ outputs zero, one obtains correlation coefficients of about ± 1, ± 0.80, ± 0.57, ± 0.28 and 0. If one compares the relative distance between the correct key value and key candidates differing by

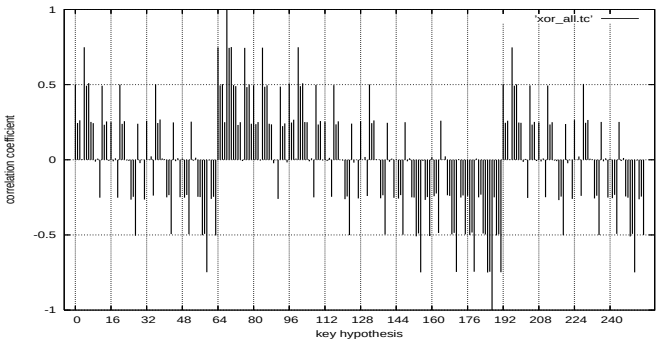


Figure 4.1: Correlation coefficient (y-axis) versus all key hypotheses (x-axis) for a XOR operation by applying Algorithm 3.5. These DSCA results were obtained by using simulation data ($N = 10,000$) generated with the correct key value $k_j^\circ = 68$ ($0x44$).

one bit, Algorithm 3.5 is the preferred choice as it yields an enhanced distance of correlation coefficients.

Other binary operations such as OR, AND, NOR, NAND, do not form a group operation on the set $\mathbb{Z}_n$. A corresponding m-bit selection function leads to the fact that $\mathbb{E}_{X,k,j}(\mathcal{W}((X \circ k)_j))$ is dependent on the key hypothesis.

4.3.2 Addition modulo 2^n

Addition and XOR operation are related primitive operations with the difference that the carry propagates between the bit positions. The selection function uses the addition modulo 2^n which is denoted by the symbol $\boxplus$. For the case $j = 0$ the selection function is

$$d_0(x, k) = \mathcal{W}(x_0 \boxplus k) - m/2 \ .$$

In case of $j > 0$ the carry of all previous additions has to be incorporated as
$C(x_0, k'_0, ..., x_{j-1}, k'_{j-1}) \in \{0, 1\}$. This results in

$$d_j(x, k) = \mathcal{W}(x_j \boxplus k \boxplus C(x_0, k'_0, ..., x_{j-1}, k'_{j-1})) - m/2$$

In contrast to boolean operations there is a preferred direction for DSCA starting from the least significant block $j = 0$ to incorporate

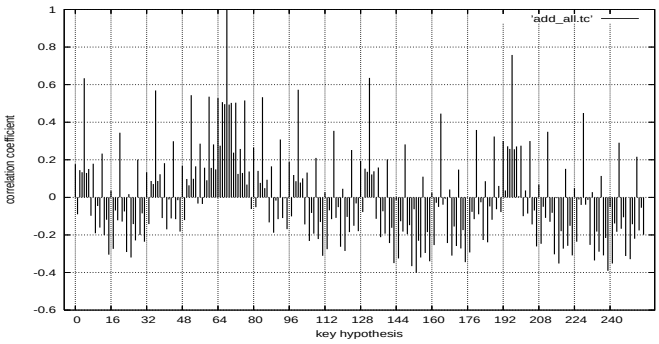


Figure 4.2: Correlation coefficient versus all key hypotheses in case of an addition modulo 2^n with the correct key hypothesis 68 ($0x44$). The results were obtained using simulation data applying Algorithm 3.5.

the carry. The correlation coefficient of $d_j(x_i, k)$ and the side channel $\Delta i_t(x_i, k_j^\circ)$ reaches the maximum if k equals k_j°.

Besides the correct value less significant correlation coefficients occur at related hypotheses. The ranking of these hypotheses is $\{k_j^\circ, k_j^\circ \pm 2^{m-1}, k_j^\circ \pm 2^{m-2}, ...\}$ and can be explained by the carry propagation. The result of the selection function using $k_j^\circ \pm 2^{m-1}$ differs for all possible values of x_i only by one bit with respect to the correct Hamming weight assuming that not all more significant bits of k° are set to one. The two hypotheses $k_j^\circ \pm 2^{m-2}$ lead for 2^{m-1} values to a one bit difference, for 2^{m-2} values to a zero bit difference, but for 2^{m-2} values the Hamming weight differs by two. If the hypotheses differ only at the least significant bit position with respect to the correct key value, the carry propagation leads to a maximal mismatch of the prediction at the transition values 0 and $2^m - 1$.

4.3.3 Modular Multiplication

The set $\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n | gcd(a, n) = 1\}$ forms a group with the operation multiplication modulo n, whereas the set $\mathbb{Z}_n$ is not a group. For IDEA a modified multiplication modulo $n = 2^{16} + 1$ is relevant which is denoted by $\odot$. The number of key hypotheses, i.e., 2^{16}, for DSCA is computationally costly, but still feasible using standard tools. This algebraic operation can be interpreted as a large S-box though it is not

implemented as a look-up table. The selection function is

$$d_j(x,k) = \mathcal{W}((x \odot k)_j) - m/2.$$

Simulation data was generated by assuming 16-bit Hamming weight (see Figure 4.3) and 8-bit Hamming weight (see Figure 4.4). In case of an 8-bit hardware architecture, both the least significant byte and the most significant byte of the intermediate result can be used for DSCA. The correct key value is more highlighted at the least significant byte of the selection function. Simulation results show that some DSCA signals occur at related hypotheses.

One can observe related hypotheses which are given by four sequences $k_{1,s}$,$k_{2,s}$, $k_{3,s}$ and $k_{4,s}$ ($s \in \{0,1,2,3,...\}$), namely

1. $k_{1,s} = 2^s k^\circ \bmod n$,
2. $k_{2,s} = 2^s(n - k^\circ) \bmod n$,
3. the following recursive sequence of divisors starting with $k_{3,0} = k^\circ$:
   if $k_{3,s}$ is even, then $k_{3,s+1} = \frac{k_{3,s}}{2}$;
   otherwise, $k_{3,s+1} = \frac{(n-k_{3,s})}{2}$.
4. $k_{4,s} = (n - k_{3,s})$

To give an example regarding Figure 4.3 and Figure 4.4: let the correct key value be 0x4931. Then the related key hypotheses are

1. $k_1 = \{$`0x4931, 0x9262, 0x24C3, 0x4986, 0x930C,...` $\}$
2. $k_2 = \{$`0xB6D0, 0x6D9F, 0xDB3E, 0xB67B, 0x6CF5,...` $\}$
3. $k_3 = \{$`0x4931, 0x5B68, 0x2DB4, 0x16DA, 0x0B6D,...` $\}$
4. $k_4 = \{$`0xB6D0, 0xA499, 0xD24D, 0xE927, 0xF494,...` $\}$

As the number of key hypotheses is increased to 2^{16}, DSCA on an unknown implementation of IDEA's multiplication is much more time-consuming compared to DES or AES. A two-stage approach is desirable that first localizes the relevant points in time and afterwards applies DSCA using all key hypotheses. The selection function at the most significant byte can be used for this purpose. For instance, a test based on all hypotheses showed that more than 99.9 % of them are detected

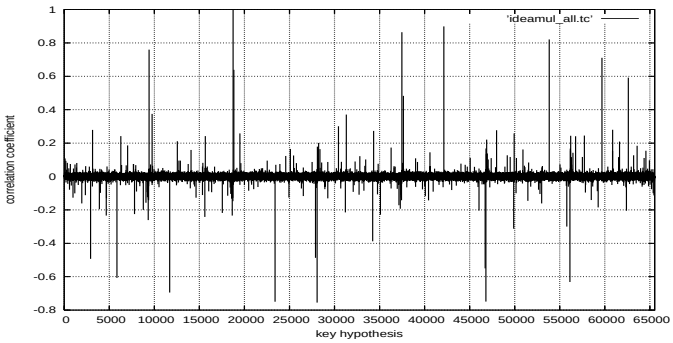


Figure 4.3: Correlation coefficient versus all key hypotheses in case the key hypothesis 18737 (0x4931) is correct. The results were obtained using simulation data generated for 16-bit Hamming weight. The selection function used also 16 bit.

at the first stage DSCA step using 2^{14} key hypotheses. For this test we assumed that secondary correlation signals can be detected for $s < 5$ for all four sequences. Further improvements are likely to succeed.

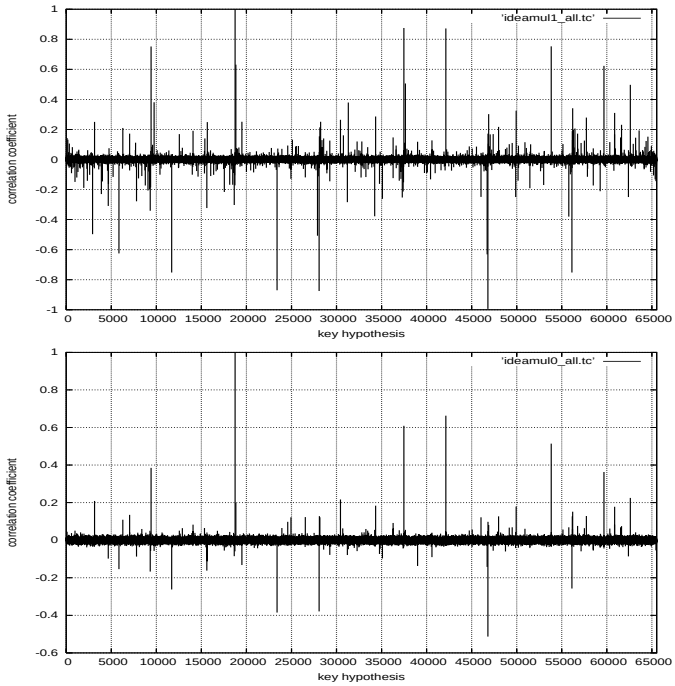


Figure 4.4: Correlation coefficient versus all key hypotheses in case the key hypothesis 18737 (0x4931) is correct. The results were obtained using simulation data generated for 8-bit Hamming weight. The selection function used was on the most significant 8 bit (upper plot) and on the least significant 8 bit.

4.4 Application to Cryptographic Algorithms

This section gives some prominent cryptographic algorithms used for encryption as well as message authentication which are based upon boolean and arithmetic operations. For each algorithm, the principle approach for DSCA is provided.

4.4.1 IDEA

IDEA uses three 16-bit group operations as there are XOR ($\oplus$), addition modulo 2^{16} ($\boxplus$) and the multiplication modulo $2^{16}+1$ ($\odot$) which treats the all-zero subblock as 2^{16} [76, 92]. The IDEA encryption is reprinted below.

Algorithm 4.1 IDEA Encryption

Input: 64-bit plaintext $M = m_1 \dots m_{64}$; 128-bit key $K = k_1 \dots k_{128}$.
Output: 64-bit ciphertext block $Y = (Y_1, Y_2, Y_3, Y_4)$.

1: Compute 16-bit subkeys $K_1^{(r)}, \dots, K_6^{(r)}$ for rounds $1 \leq r \leq 8$ and $K_1^{(9)}, \dots, K_4^{(9)}$ for the output transformation.
2: $X_1 \leftarrow (m_1 \dots m_{16})$; $X_2 \leftarrow (m_{17} \dots m_{32})$; $X_3 \leftarrow (m_{33} \dots m_{48})$; $X_4 \leftarrow (m_{49} \dots m_{64})$, i.e., X_i is a 16-bit data store.
3: **for** r from 1 to 8 **do**
4: $X_1 \leftarrow X_1 \odot K_1^{(r)}; X_4 \leftarrow X_4 \odot K_4^{(r)}; X_2 \leftarrow X_2 \boxplus K_2^{(r)}; X_3 \leftarrow X_3 \boxplus K_3^{(r)}$;
5: $t_0 \leftarrow K_5^{(r)} \odot (X_1 \oplus X_3)$;
6: $t_1 \leftarrow K_6^{(r)} \odot (t_0 \boxplus (X_2 \oplus X_4))$;
7: $t_2 \leftarrow t_0 \boxplus t_1$;
8: $X_1 \leftarrow X_1 \oplus t_1; X_4 \leftarrow X_4 \oplus t_2; a \leftarrow X_2 \oplus t_2; X_2 \leftarrow X_3 \oplus t_1; X_3 \leftarrow a$;
9: **end for**
10: $Y_1 \leftarrow X_1 \odot K_1^{(9)}; Y_4 \leftarrow X_4 \odot K_4^{(9)}; Y_2 \leftarrow X_3 \boxplus K_2^{(9)}; Y_3 \leftarrow X_2 \boxplus K_3^{(9)}$;

IDEA decryption follows Algorithm 4.1, too, however, the key schedule for decryption keys requires to compute additive and multiplicative inverses of the encryption keys (see [92] for details).

Due to the IDEA key schedule the first eight 16-bit subkeys directly give the original IDEA key. The DSCA selection functions used are set up on the operations $\boxplus$ and $\odot$ (cf. Algorithm 4.1). DSCA on the

Algorithm 4.2 IDEA Key Schedule (Encryption)

Input: 128-bit key $K = k_1 \dots k_{128}$.

Output: 52 16-bit key sub-blocks $K_i^{(r)}$ for 8 rounds r and the output transformation.

1: Order the subkeys $K_1^1 \dots K_6^1$, $K_1^2 \dots K_6^2$, ..., $K_1^8 \dots K_6^8$, $K_1^9 \dots K_4^9$

2: Partition K into eight 16-bit blocks; assign these directly to the first 8 subkeys.

3: Do the following until all 52 subkeys are assigned: cyclic shift K left 25 bits; partition the result into 8 blocks; assign these blocks to the next 8 subkeys.

subkey values $K_1^{(r)}$, $K_4^{(r)}$, $K_5^{(r)}$ and $K_6^{(r)}$ uses the operation $\odot$ for the selection function. DSCA on the subkey values $K_2^{(r)}$ and $K_3^{(r)}$ uses the operation $\boxplus$. The operation $\oplus$ can also serve as an additional selection function that reduces remaining candidates of previous results.

4.4.2 RC6

The design of RC6 makes use of simple primitive operations (integer addition modulo 2^w $(+)$, integer multiplication modulo 2^w $(\times)$, bitwise exclusive-or and key-dependent bit rotations). RC6-$w/r/b$ [120] works on four w-bit registers A, B, C and D which contain the plaintext and the corresponding ciphertext. The number of rounds is given by r and b denotes the number of key bytes. The key schedule of RC6 yields $2r + 4$ w-bit subkeys $S[i]$, with $i \in \{0, 1, ..., 2r + 3\}$. The RC6-$w/r/b$ encryption is reprinted in Algorithm 4.3. Here, $a \lll b$ rotates the w-bit word a to the left by the amount given by the least significant lg w bits of b with lg w denoting the base-two logarithm of w.

During encryption of RC6-$w/r/b$, key addition is done by using addition modulo 2^w. The first keys are $S[0]$ and $S[1]$ using the known values B and D. During each iteration, A, B, C, D, t and u are known if all previous subkeys are already recovered by DSCA. The key hypotheses are set up on $S[2i]$ and $S[2i + 1]$. The selection function is always addition modulo 2^w. DSCA signals are expected in the subsequent iteration wherein this intermediate value acts as partial multiplicator. Due to the 'one-way' key schedule (see Algorithm 4.4) all r rounds of RC6-$w/r/b$

Algorithm 4.3 RC6-$w/r/b$ Encryption

Input: Plaintext stored in four w-bit input registers A, B, C, D;
Number r of rounds;
w-bit round keys $S[0, \ldots, 2r+3]$
Output: Ciphertext stored in A, B, C, D;

1: $B = B + S[0]$;
2: $D = D + S[1]$;
3: **for** i from 1 to r **do**
4: $t = (B \times (2B+1)) \lll \lg\ w$;
5: $u = (D \times (2D+1)) \lll \lg\ w$;
6: $A = ((A \oplus t) \lll u) + S[2i]$;
7: $C = ((C \oplus u) \lll t) + S[2i+1]$;
8: $(A, B, C, D) = (B, C, D, A)$;
9: **end for**
10: $A = A + S[2r+2]$;
11: $C = C + S[2r+3]$;

Algorithm 4.4 RC6-$w/r/b$ Key Schedule

Input: User-supplied b byte key preloaded into the c-word array $L[0, \ldots, c-1]$;
Number r of rounds;
Constants P_w and Q_w;
Output: w-bit round keys $S[0, \ldots, 2r+3]$;

1: $S[0] = P_w$;
2: **for** i from 1 to $2r+3$ **do**
3: $S[i] = S[i-1] + Q_w$;
4: **end for**
5: $A = B = i = j = 0$;
6: $v = 3 \times \max\{c, 2r+4\}$;
7: **for** s from 1 to v **do**
8: $A = S[i] = (S[i] + A + B) \lll 3$;
9: $B = L[j] = (L[j] + A + B) \lll (A + B)$;
10: $i = (i+1) \bmod (2r+4)$;
11: $j = (j+1) \bmod c$;
12: **end for**

have to be attacked by DSCA iteratively.

4.4.3 HMAC-Construction

The specification of the HMAC construction can be found in [20] and [19]. The HMAC makes use of a secure hash function H, as e.g., RIPEMD-160 and SHA-1. Let $Text$ be the input message to be secured for message authentication and let K be the secret key used. The HMAC is a nested construction that uses two calls to the hash function H.

$$HMAC(Text, K) = H(K \oplus opad, H(K \oplus ipad, Text)) \tag{4.1}$$

It makes use of two fixed strings *ipad* and *opad*, both of the size of one message block to H. As the first message block for each call to H is a constant value that depends only on K these two values are precalculated and stored instead of K in efficient implementations. Let the two secret values for the inner and outer hash function be defined as follows:

$$K_i = H(K \oplus ipad) \tag{4.2}$$

and

$$K_o = H(K \oplus opad)\ . \tag{4.3}$$

In the HMAC construction the secret values K_i and K_o are initialization vectors (IVs) for the hash computation. DSCA is applied on the first iterations of the inner hash function of the HMAC and after the disclosure of K_i on the first iterations of the outer hash function to reveal K_o.

DSCA selection functions depend on the hash function used. The concrete procedures are given for RIPEMD-160 and SHA-1 below. The DSCA problem is to find the secret IV by analyzing the first iterations of the hash function.

HMAC-RIPEMD-160

In [48] RIPEMD-160 is specified including a pseudo code in Annex A. As constants are not relevant for the DSCA approach they are not listed in Algorithm 4.5 and it is referred to [48]. The addition '+' is modulo 2^{32}.

The target for DSCA is to reveal the secret IV that is split into the five 32-bit words A, B, C, D and E as well as A', B', C', D' and E'. In principle, both parallel processing parts of the main loop can be used

Algorithm 4.5 RIPEMD-160

Input: 160-bit initial value $IV = (h_0, h_1, h_2, h_3, h_4)$;
t times sixteen 32-bit message blocks $X_i[j]$ with $0 \leq i \leq t-1$ and $0 \leq j \leq 15$
nonlinear functions

$$f(j,x,y,z) = \begin{cases} x \oplus y \oplus z, & \text{if } 0 \leq \text{j} \leq 15; \\ (x \wedge y) \vee (\neg x \wedge z), & \text{if } 16 \leq \text{j} \leq 31; \\ (x \vee \neg y) \oplus z, & \text{if } 32 \leq \text{j} \leq 47; \\ (x \wedge z) \vee (y \wedge \neg z), & \text{if } 48 \leq \text{j} \leq 63; \\ x \oplus (y \vee \neg z), & \text{if } 64 \leq \text{j} \leq 79; \end{cases}$$

dependent on internal round j
constants $K(j)$ and $K'(j)$ dependent on internal round j (see [48])
permutation indexes $r(j)$ and $r'(j)$ dependent on internal round j (see [48])
number of rotations $s(j)$ and $s'(j)$ dependent on internal round j (see [48])

Output: 160-bit hash code $Y = (h_0, h_1, h_2, h_3, h_4)$.

1: **for** i from 0 to $t-1$ **do**
2: $A := h_0; B := h_1; C := h_2; D := h_3; E := h_4;$
3: $A' := h_0; B' := h_1; C' := h_2; D' := h_3; E' := h_4;$
4: **for** j from 0 to 79 **do**
5: $T := rol_{s(j)}(A + f(j,B,C,D) + X_i[r(j)] + K(j)) + E;$
6: $A := E; E := D; D := rol_{10}(C); C := B; B := T;$
7: $T' := rol_{s'(j)}(A' + f(79-j,B',C',D') + X_i[r'(j)] + K'(j)) + E';$
8: $A' := E'; E' := D'; D' := rol_{10}(C'); C' := B'; B' := T';$
9: **end for**
10: $T := h_1 + C + D'; h_1 := h_2 + D + E'; h_2 := h_3 + E + A';$
11: $h_3 := h_4 + A + B'; h_4 := h_0 + B + C'; h_0 := T;$
12: **end for**

to mount DSCA. However, as the nonlinear function used is different in both lines the approach slightly differs.

First, it is focused on the calculation of the five 32-bit words A, B, C, D, and E. For the first sixteen iterations, $r(j)$ equals j, $K(j)$ is zero, and the compression function f is $f(x,y,z) = x \oplus y \oplus z$. It turns out

that this processing part yields the most straight-forward DSCA attack path.

The DSCA selection functions are applied at successive intermediate results d_1 to d_5 that occur during the processing of the first three iterations in the main loop and depend both on known varying data parts and a static secret. The intermediate results at d_1, d_2, and d_4 are revealed by an addition modulo 2^{32}, d_3 and d_5 are obtained by an XOR operation. Note that one deals with 32-bit intermediate results that are usually subsequently targeted, e.g., byte-by-byte. The static key value to be attacked at each selection function is included in brackets '[' and ']'. An additional subindex is used for variables A, B, C, D, E, and T which denotes the current iteration number, i.e., A_0 is A in the first iteration and A_1 is A in the second iteration of the main loop. At the first iteration, two intermediate results can be used to mount DSCA.

$$\begin{aligned} d_1 &= [A_0 + (B_0 \oplus C_0 \oplus D_0)] + X_0 \\ d_2 &= T_0 \\ &= rol_{11}(A_0 + (B_0 \oplus C_0 \oplus D_0) + X_0) + [E_0] \end{aligned}$$

Here, X_0 serves as known varying data for d_1 while at d_2, the term $A_0 + (B_0 \oplus C_0 \oplus D_0) + X_0$ is presumed to be known provided that DSCA was successful at d_1. In the second iteration of RIPEMD-160 it is presumed that $B_1 = T_0$ is known because of DSCA success at d_2.

$$\begin{aligned} d_3 &= B_1 \oplus [C_1 \oplus D_1] \\ &= T_0 \oplus [(B_0 \oplus rol_{10}(C_0))] \\ d_4 &= T_1 \\ &= rol_{14}(A_1 + (B_1 \oplus C_1 \oplus D_1) + X_1) + [E_1] \\ &= rol_{14}(E_0 + (T_0 \oplus (B_0 \oplus rol_{10}(C_0)) + X_1) + [D_0] \end{aligned}$$

In the third iteration one intermediate result remains.

$$\begin{aligned} d_5 &= B_2 \oplus C_2 \oplus [D_2] \\ &= T_1 \oplus (B_1 \oplus [rol_{10}(C_1)]) \\ &= T_1 \oplus (T_0 \oplus [rol_{10}(B_0)]) \end{aligned}$$

If DSCA is successful, the results of all selection functions can be combined to reveal A_0, B_0, C_0, D_0 and E_0 which is the secret IV.

For the second parallel processing part, another non-linear compression function is used which makes the main difference except for some other constants. The DSCA approach is quite similar at the first two iterations of RIPEMD-160. It is

$$\begin{aligned}
d'_1 &= [A'_0 + (B'_0 \oplus (C'_0 \vee \neg D'_0)] + X_5 + K'(0) \\
d'_2 &= T'_0 \\
&= rol_8(A'_0 + (B'_0 \oplus (C'_0 \vee \neg D'_0)) + X_5 + K'(0)) + [E'_0] \\
d'_3 &= B'_1 \oplus [C'_1 \vee \neg D'_1] \\
&= T'_0 \oplus [(B'_0 \vee \neg rol_{10}(C'_0))] \\
d'_4 &= T'_1 \\
&= rol_9(A'_1 + (B'_1 \oplus (C'_1 \vee \neg D'_1)) + X_{14} + K'(0)) + [E'_1] \\
&= rol_9(E'_0 + (T'_0 \oplus (B'_0 \vee \neg rol_{10}(C'_0)) + X_{14} + K'(0)) + [D'_0]
\end{aligned}$$

The third iteration involves some complication because of the compression function used. One may think of defining the intermediate result $C'_2 \vee \neg[D'_2]$, however, $\vee$ is not a group operation. Because of this, it is more adequate to set up hypotheses on the value D'_2 but to use the result of $C'_2 \vee \neg[D'_2]$ for DSCA at the XOR operation with B'_2.

$$\begin{aligned}
d'_5 &= (B'_2 \oplus (C'_2 \vee \neg[D'_2])) = T'_1 \oplus (B'_1 \vee \neg[rol_{10}(C'_1)]) \\
&= T'_1 \oplus (T'_0 \vee \neg[rol_{10}(B'_0)])
\end{aligned}$$

HMAC-SHA-1

In [106] SHA-1 is specified. The algorithm description is provided below, again concrete constants are not given here, as they are not important for the DSCA approach.

If compared to RIPEMD-160 the DSCA approach at SHA-1 depends more strongly on implementation choices, i.e., the order of summands can be exchanged which may change the number of DSCA selection functions per iteration. For the following considerations it is assumed that the sequence of operations is identical to Algorithm 4.6. Again, an additional subindex is used to indicate the iteration number. For the first SHA-1 iteration there is only one intermediate result to be attacked:

$$\begin{aligned}
d_1 &= T_0 \\
&= [rol_5 A_0 + ((B_0 \wedge C_0) \vee (\neg B_0 \wedge D_0)) + E_0] + K_0 + W_0
\end{aligned}$$

Algorithm 4.6 SHA-1

Input: 160-bit initial value $IV = (h_0, h_1, h_2, h_3, h_4)$;
t times sixteen 32-bit message blocks $X_i[j]$ with $0 \leq i \leq t-1$ and $0 \leq j \leq 15$
Nonlinear functions dependent on internal round j:

$$f(j,x,y,z) = \begin{cases} (x \wedge y) \oplus (\neg x \wedge z), & \text{if } 0 \leq \text{j} \leq 19; \\ x \oplus y \oplus z, & \text{if } 20 \leq \text{j} \leq 39; \\ (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z), & \text{if } 40 \leq \text{j} \leq 59; \\ x \oplus y \oplus z, & \text{if } 60 \leq \text{j} \leq 79; \end{cases}$$

Message blocks dependent on internal round j:

$$W_i[j] = \begin{cases} X_i[j], & \text{if } 0 \leq \text{j} \leq 15; \\ rol_1(Z) & \text{if } 16 \leq \text{j} \leq 79; \end{cases}$$

whereat $Z := W_i[j-3] \oplus W_i[j-8] \oplus W_i[j-14] \oplus W_i[j-16]$;
Constants $K(j)$ dependent on internal round j (see [106]);
Output: 160-bit hash code $Y = (h_0, h_1, h_2, h_3, h_4)$.

1: **for** i from 0 to $t-1$ **do**
2: $A := h_0; B := h_1; C := h_2; D := h_3; E := h_4;$
3: **for** j from 0 to 79 **do**
4: $T := rol_5 A + f(j,B,C,D) + E + K_j + W_i[j];$
5: $E := D; D := C; C := rol_{30}(B); B := A; A := T;$
6: **end for**
7: $h_0 = h_0 + A; h_1 = h_1 + B; h_2 = h_2 + C; h_3 = h_3 + D; h_4 = h_4 + E;$
8: **end for**

In the second SHA-1 iteration one can mount DSCA at two secrets. Here, d_3 and d_{3a} attack the same secret, but with different intermediate

results.

$$\begin{aligned} d_2 &= rol_5 A_1 + [((B_1 \wedge C_1) \vee (\neg B_1 \wedge D_1))] \\ &= rol_5 T_0 + [((A_0 \wedge rol_{30} B_0) \vee (\neg A_0 \wedge C_0))] \\ d_3 &= d_2 + [E_1] \\ &= d_2 + [D_0] \\ d_{3a} &= T_1 \\ &= d_2 + [E_1] + K_1 + W_1 \\ &= d_2 + [D_0] + K_1 + W_1 \end{aligned}$$

In the third SHA-1 iteration one has to set up hypotheses on the joint values of C_2 and D_2. If one uses a byte-per-byte approach for each value this leads to 2^{16} key hypotheses. There is one additional intermediate result d_5 which might be useful either if the value of C_0 can not uniquely be determined or if the implementation prevents DSCA results at the intermediate value d_2.

$$\begin{aligned} d_4 &= (B_2 \wedge [C_2]) \vee (\neg B_2 \wedge [D_2]) \\ &= (A_1 \wedge [rol_{30} B_1]) \vee (\neg A_1 \wedge [C_1]) \\ &= (T_0 \wedge [rol_{30} A_0]) \vee (\neg T_0 \wedge [rol_{30} B_0]) \\ d_5 &= d_4 + [E_2] \\ &= d_4 + [D_1] \\ &= d_4 + [C_0] \end{aligned}$$

4.5 Experimental Results of an IDEA Implementation

For the experimental testing IDEA was chosen as it uses three algebraic groups. The IDEA implementation was carried out in Assembly on an 8051 microcontroller (CISC, von-Neumann architecture) and on an 8-bit ATM163 AVR micro-controller (RISC, Harvard architecture). It was assured that both implementations have a constant execution time to exclude broad correlation signals based on timing displacements. The implementations did not include software countermeasures to counteract DSCA.

In both tests the DSCA characteristics of the simulation results using the Hamming weight model were basically confirmed. For the 8051 microcontroller nearly perfect DSCA signals were obtained revealing that the leakage model based on the Hamming weight is very well suited to the real physical leakage of an 8051 microcontroller. The analysis turned out to be more difficult for the ATM163 AVR microcontroller, because of that, experimental results are discussed in more detail for the AVR microcontroller below.

4.5.1 8051 Microcontroller

For the experimental analysis, 5000 single measurements were recorded at a sampling rate of 200 MHz using a standard DSCA measurement set-up on the ground pins of the microcontroller. The IDEA key used was in hexadecimal notation:

`'F1 3C 06 5E 14 A1 55 00 12 FF 52 AA 1B 71 3A 08'`

The correct subkey values could be easily determined by DSCA, both for the selection functions at the modular addition and at the IDEA multiplication. As part of this thesis, for one relevant point in time experimental DSCA results are shown for the least and most significant key byte of $K_2^{(1)}$='06 5E' by using modular addition (see Figure 4.5) for all key hypotheses. In summary, for an 8051 microcontroller the Hamming weight leakage model has turned out to be an appropriate approximation of the real physical leakage.

4.5.2 AVR Microcontroller

The target platform is a smart card embedded ATM163 microcontroller that runs a basic variant of the open source smart card operating system SOSSE [34]. At a previous characterisation step of the ATM163 the following properties were determined.

- The outstanding DSCA signals are caused by Hamming distances of data that is subsequently transferred on the internal bus.
- Correlation signals on the input data to an operation can be revealed with sufficient clarity using the Hamming weight model whereas correlation signals on the output data of an operation are difficult to prove.

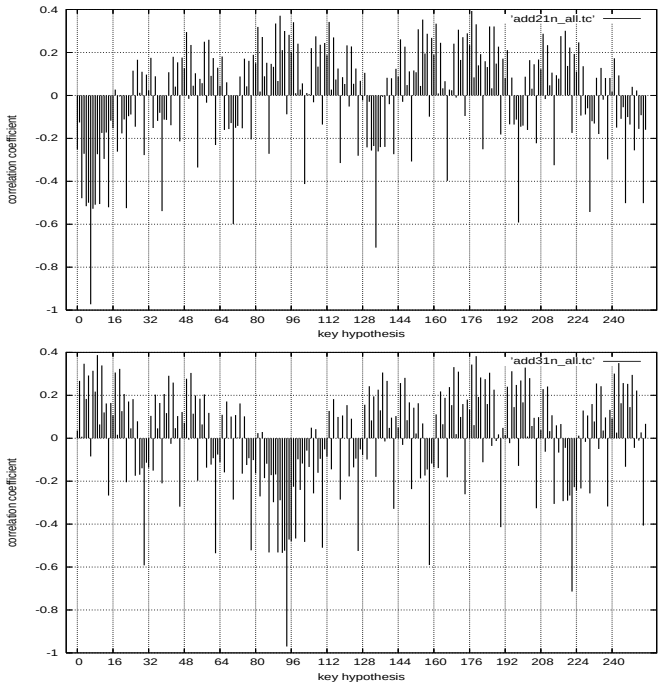


Figure 4.5: 8051 microcontroller. Correlation coefficient versus all key hypotheses for one relevant instant in time. Here, the key hypothesis 1630 (0x065E) is the correct one for $K_2^{(1)}$. The results were obtained using power consumption measurement data of an 8051 microcontroller. The selection function used was the addition modulo 2^{16}. The DSCA selection function was mounted on the least significant 8 bit (lower plot) and afterwards on the most significant 8 bit (upper plot). Note that the DSCA results are of negative sign if compared to Section 4.3.2.

Consequently, the Hamming weight model is expected to be successful at the points in time that process the output data of previous operations as the input values. An additional result is that care has to be taken at the load sequence when alternating key data with known input/output data at subsequent instructions at an AVR core. If known data and secret data are moved one after the other from the SRAM to the working registers using the `ldd` instruction, nearly perfect correlation signals are revealed

using the 8-bit XOR selection function. Note, that this observation was also made if two `ldd` instructions are separated by some arithmetic instructions. An appropriate countermeasure would be to encapsulate the transfer of secret key data by the load of internal, random data.

For the experimental analysis, again 5000 single measurements were accumulated at a sampling rate of 200 MHz using a standard DSCA measurement on the ground pin of the ATM163 microcontroller. The IDEA key loaded was in hexadecimal notation:

`'7E 24 95 E1 E5 0C 86 CE 8C C3 1B 80 C0 65 B2 AF'`

Addition modulo 2^{16}:

Generally, if not superposed by strong correlation signals on the input data, the correct key values are revealed by DSCA using 8-bit selection functions for the modular addition. The particular points in time that show correlation signals on the input data can be independently identified by correlation signals on the input data of IDEA.

The experimental DSCA characteristics do not always correspond to the expected ones (see Figure 4.6). The deviations can be explained by the superposing of signals, especially by leakage of the input data. The analysis on the primary origin of each signal obtained turns out to be a difficult task on the ATM163.

The following is the actual code sequence of the modular addition:

```
ldd r0,Z+2 ; 1st addition:  load subkey bytes from SRAM
ldd r1,Z+3
add r5,r1 ; addition with input bytes
adc r4,r0
ldd r0,Z+4 ; 2nd addition:  load subkey bytes from SRAM
ldd r1,Z+5
add r21,r1 ; addition with input bytes
adc r20,r0
```

The `add` instruction turns out to be extremely vulnerable against the 8-bit XOR selection function if certain registers are used. In the current example, the instruction `add r5,r1` yields significant DSCA signals using the 8-bit XOR selection function at the least significant key byte (see Figure 4.7). However, this strong dependency was not

confirmed at the instruction `add r21,r1`.

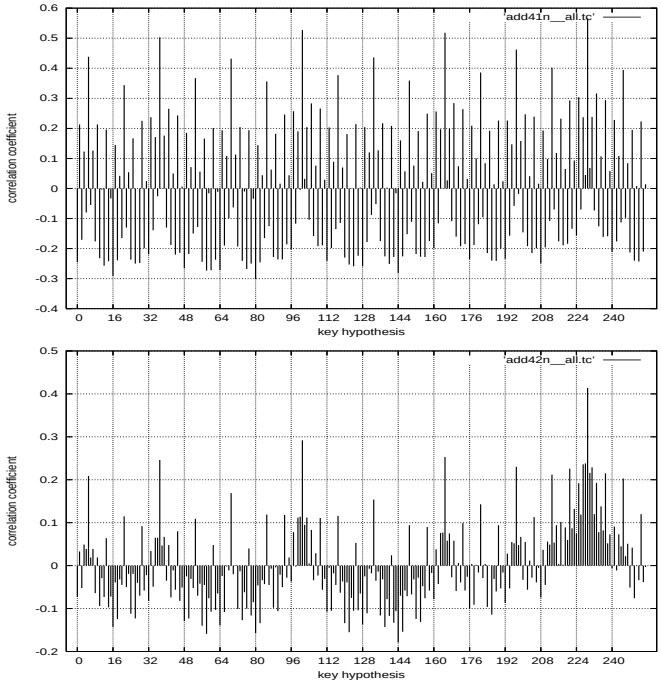


Figure 4.6: AVR microcontroller. Correlation coefficient versus all key hypotheses using the selection function on the modular addtion at two different points in time. The correct key value 229 (0xE5) for the most significant byte of $K_3^{(1)}$ is revealed, but only the characteristic in the lower plot points to a pure signal. During the time of the upper plot (negative) correlation signals on the input data are also proven.

Multiplication modulo $2^{16} + 1$:

The points in time that yield high signals are identified using the advantage that the key is known. DSCA yielded clear correlation signals for the least and most significant byte of the selection function at all relevant positions in time (see Figure 4.8). The experimental DSCA characteristics are consistent with the expected ones.

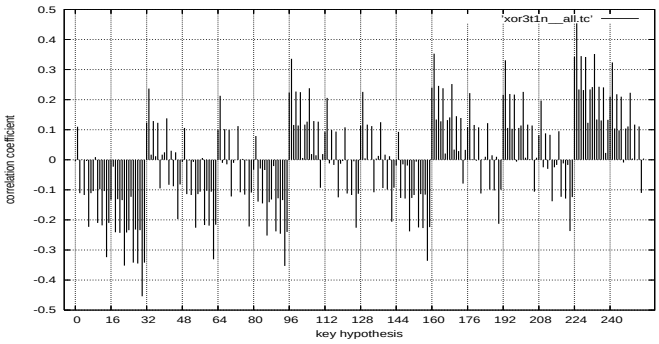


Figure 4.7: AVR microcontroller. Correlation coefficient versus all key hypotheses using the XOR selection function at the ldd instruction `ldd r1,Z+3`. The correct key value 225 (0xE1) for the least significant byte of $K_2^{(1)}$ is revealed.

As result, the Hamming weight selection function was successfully applied, even in presence of a hardware platform that leaks for the most part differential signals.

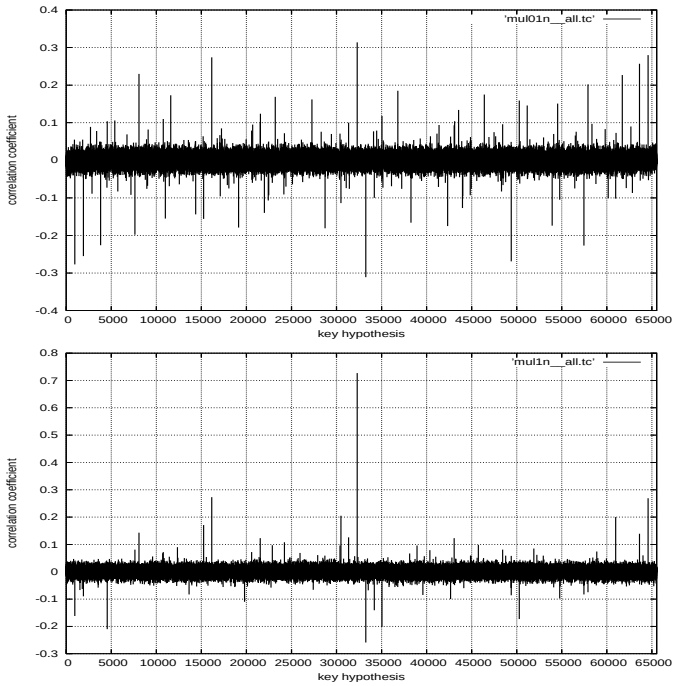


Figure 4.8: AVR microcontroller. Correlation coefficient versus all key hypotheses. The key value $K_1^{(1)}$=32292 (0x7E24) is confirmed. The selection function used was at the most significant byte (upper plot) and at the least significant byte (lower plot). The instants for the plots were different.

Chapter 5

Stochastic Methods for Differential Side Channel Analysis

5.1 Contribution

This chapter presents a new multivariate approach to optimize the efficiency of differential side channel cryptanalysis with means of stochastic methods. The new idea is to profile the real physical leakage by approximation within a suitable chosen vector subspace that is spanned by basis functions of the overall data space.

This stochastic approach is a two stage DSCA method, i.e., it consists of a profiling and a key recovery phase. Under appropriate conditions – which are generally fulfilled for block ciphers – profiling requires only one test key.

This chapter introduces two main methods as part of the stochastic model: the *minimum principle* and the *maximum likelihood principle*. Both differ in certain parts of the profiling and key recovery phase. While it is sufficient for the minimum principle to profile the deterministic side channel leakage at relevant instants, the maximum likelihood principle additionally requires an estimation of the multivariate noise during profiling. Moreover, it is shown that profiling can be even done without knowing the key. The generalization to comprehend both masking countermeasures as well as the usage of multiple physical channels is included.

The suitability of the model and algorithms for profiling and key recovery are tested and confirmed by experiments using an AVR microcontroller. If common leakage models such as the Hamming weight or Hamming distance are not appropriate for the real physical leakage common DSCA methods (see Section 3.3.4) are suboptimal and may not perform well. Here, the new stochastic methods are able to fill this gap as they output an approximation on the real physical leakage. It is experimentally demonstrated that the adaptation of probability densities is clearly advantageous regarding the DSCA correlation method. First of all, multiple leakage signals at different times can be jointly evaluated. But even if considering only one instant also an efficiency gain in key recovery is observed, however, this efficiency gain may be reduced if the Hamming weight leakage is a good approximation for the real physical leakage.

Though the efficiency at key recovery is limited by template attacks profiling is much more efficient. This makes the stochastic methods attractive if the adversary is limited in capabilities such as the number

of measurements or the change of keys and data at the profiling stage. Moreover, future cryptographic algorithms and operations may be designed for enlarged word sizes, e.g., 16 and 32-bit word size. On such processors profiling for all subkeys according to the approach of template attacks becomes a hard problem in practice.

The main part of this chapter is a revised extension of the work published in [126]. Section 5.5 contains a new application of the stochastic model in the presence of masking. Masking implies algorithmic countermeasures which are commonly used to secure implementations of block ciphers to prevent DSCA success. This chapter shows how masking can be defeated by using high-order analysis in the stochastic model.

In summary, the stochastic model improves the understanding of the source of an attack and its true risk potential. It is therefore important for a developer of a cryptographic system for implementing effective and reliable countermeasures that prevent also privileged attacks.

5.2 Previous Work

Side channel cryptanalysis exploits physical information that is leaked during the computation of a cryptographic device. The most powerful leakage consists of instantaneous physical signals which are direct responses on the internal processing. These instantaneous observables can be obtained by measuring the power dissipation or the electromagnetic emanation of the cryptographic device as a function of time. Power analysis, which was first introduced in [73] and electromagnetic analysis [53] are based on the dependency of the side channel information on the value of intermediate data, which is in turn caused by the physical implementation.

Differential side channel cryptanalysis identifies the correct key value by statistical methods for hypothesis testing. Differential Power Analysis (DPA) [73] turned out to be a very powerful technique on unknown implementations. The single measurements are partitioned according to the result of a selection function that depends both on known data and on key hypotheses. DSCA selection functions either target a single bit of an intermediate result or one decides in favour of a common leakage model such as the Hamming weight and Hamming distance. If one uses single-bit DSCA one typically looses available information leaked

by other bits of an intermediate result. Also the Hamming weight and Hamming distance model is not always appropriate to describe the real physical leakage, see for instance Chapter 4 regarding the experimental results on the AVR microcontroller. For the statistical tools [73] suggested to just use the difference of means for the two sets of single measurements. Improved statistics are the student's T-Test and the correlation method which are given in [6]. Additional guidelines for testing the susceptibility of an implementation are presented in [44].

Other contributions assume that the adversary is more powerful, e.g, that the adversary is able to load key data into the cryptographic device. Profiling as a preparation step of power analysis was first described by [49]. Probably the most sophisticated strategy is a template based attack [40] which aims to optimize Simple Power Analysis (SPA) and requires a precise characterization of the side channel. Profiling in a template attack means that one measures the physical leakage for each subkey dependency. Moreover, physical information can be captured simultaneously by different measurement setups, e.g., by measuring the EM emanation and the power consumption in parallel (multi channel attacks) [4].

Advanced stochastic methods have turned out to be efficient tools to optimize pure timing and combined timing and power attacks. Using such methods, the efficiency of some known attacks could be increased considerably (up to a factor of fifty), some attacks could be generalized and new attacks were conceived [123, 124, 125].

The stochastic approach for differential side channel analysis that is presented in this chapter is a multivariate approach. While DSCA standard methods are univariate, the Template Attack [40] was previously proposed as the first multivariate side channel attack, see Section 3.3.8. The use of templates is a two stage SSCA attack that requires profiling for each subkey dependency. As result of profiling, for each subkey dependency one has to build a template, i.e. a multivariate Gaussian probability density to indeed observe this subkey. Classification is done by a maximum likelihood approach, i.e., one decides in favour of that subkey that yields the maximum probability value among all subkey values.

In response to Differential Side Channel Analysis (DSCA) developers of cryptographic implementations may include randomization techniques such as secret splitting or masking schemes, e.g., [39, 43]. These ran-

domization techniques shall prevent from predicting any relevant bit in any cycle of the implementation. As result, statistical tests using physical observables at one instant cannot be assumed to be successful in key recovery. However, as already indicated by [73] high-order differential analysis can combine multiple samples from within a measurement trace. Previous work on second-order DSCA [93, 143] constructs a new leakage signal by multiplying (or subtracting) the observables at related time instants before statistics is applied. This reduction generally loses information if compared to a multivariate analysis. By assuming that the leakage signals follow the n-bit Hamming weight model [67] provided a derivation on the height of the expected second-order DPA signals and [112] uses predicted probability density functions to improve second-order power analysis. Further practical results for second- and higher-order DPA acting on the Hamming weight assumption are given in [109, 131]. Moreover, Template-enhanced DPA attacks were introduced in [5] to defeat masking under the assumption that the adversary has access to an implementation with a biased random number generator during profiling. In [108] different types of template attacks attacks on masked implementations are studied and it is concluded that a template based DPA attack leads to the best results.

5.3 The Stochastic Model and Algorithms

5.3.1 The Principle Idea

In this section the principle idea for the new stochastic model for DSCA is introduced. This model assumes that the adversary measures physical observables at time t in order to guess a subkey $k \in \{0,1\}^s$. The letter $x \in \{0,1\}^d$ denotes a known part of the data, i.e., plaintext or ciphertext, respectively. A physical observable $I_t(x,k)$ at time t is seen as a random variable

$$I_t(x,k) = h_t(x,k) + R_t \ . \tag{5.1}$$

The first summand $h_t(x,k)$ quantifies the deterministic part of the measurement as far it depends on x and k. The term R_t denotes a random variable that does not depend on x and k. R_t includes all kinds of noise as there are intrinsic and external noise, noise of the measurement

apparatus and algorithmic noise that stems from deterministic contributions that do not depend on x and k. The random variable $I_t(x,k)$ can be interpreted as a 'displaced' noise R_t with mean displacement $h_t(x,k)$. Without loss of generality one may assume that $\mathbb{E}(R_t) = 0$ since otherwise one could replace $h_t(x,k)$ and R_t by $h_t(x,k)+\mathbb{E}(R_t)$ and $R_t-\mathbb{E}(R_t)$, respectively. It follows $\mathbb{E}(I_t(x,k)) = h_t(x,k)$. It is noted again that random variables are denoted with capital letters while their realizations (measured quantities), i.e. values assumed by these random variables, are denoted with the respective small letters.

Example 5.1. For an AES implementation one may target the intermediate result $x_0 := S(x \oplus k)$, i.e., the 8-bit output of the AES S-box S given 8-bit data x and an 8-bit subkey k. Then t is a time instant, e.g., during the first round. Physical observables that result from further processing, e.g., of the remaining S-boxes, are contained in the term R_t.

Example 5.2. Because of $\mathbb{E}(I_t(x,k)) = h_t(x,k)$ one can determine $h_t(x,k)$ by computing the empirical mean of several measured quantities $i_t(x,k)$ for each fixed pair $(x,k) \in \{0,1\}^d \times \{0,1\}^s$.

The function $h_t(\cdot,\cdot)$:$\{0,1\}^d \times \{0,1\}^s \to \mathbb{R}$ is typically not known to the adversary which constitutes a problem for the efficiency of side channel cryptanalysis. Further, the determination of $h_t(x,k)$ as done in Example 5.2 for all 2^{d+s} pairs of (x,k) may be costly in terms of measurements or even infeasible due to limited capabilities of an adversary, e.g., the adversary may not be allowed to load keys into the profiling device. Determining $h_t(x,k)$ for 2^{d+s} pairs of (x,k) is indeed the first step of a Template Attack.

In the stochastic model profiling aims to determine a function $h_t^*(\cdot,\cdot)$ that is 'close' to the unknown function $h_t(\cdot,\cdot)$. For simplicity attention is restricted to the set of functions $\mathcal{F}_{u;t}$, that is a real vector subspace spanned by u known functions $g_{tl}\colon \{0,1\}^d \times \{0,1\}^s \to \mathbb{R}$ for each time instant t.

$$\mathcal{F}_{u;t} := \{h'\colon \{0,1\}^d \times \{0,1\}^s \to \mathbb{R} \mid \sum_{l=0}^{u-1} \beta_l g_{tl} \text{ with } \beta_l \in \mathbb{R}\} \tag{5.2}$$

One may assume that the functions g_{tl} are linearly independent so that $\mathcal{F}_{u;t}$ is isomorphic to $\mathbb{R}^u$. $\mathcal{F}_{u;t}$ may either contain the searched function h_t itself or at least a function h_t^* that is sufficiently 'close' (to be made precise in Section 5.3.2) to h_t.

> **Example 5.3.** For an n-bit intermediate result x_0 one may define an $n+1$-dimensional vector subspace that is spanned by the function 1 and the single bits of x_0.

While Section 5.3.2 provides facts on the distance between the function h_t and functions $h' \in \mathcal{F}_{u;t}$ for *one* fixed subkey value k, the basic question to be solved is to obtain profiling densities for *all* subkey values as result of the profiling phase. This can be achieved if the EIS property introduced in Section 5.3.3 holds which is typically given for block ciphers.

The concrete procedure used to determine $h_t^*(\cdot,\cdot)$ is the method of general linear least squares that is already introduced as part of multiple regression analysis in Section 2.1.3. Its application to the profiling stage of the new stochastic model is given in Section 5.3.4 that contains all algorithms for profiling. Especially, it also answers the question of how to carry out profiling without knowing the key. Section 5.3.5 explains the algorithms for key recovery, while Section 5.3.6 and Section 5.3.7 deal with the generalizations to a masked implementation and the use of multiple physical channels. Experimental applications of the stochastic methods are given for two implementations on AVR microcontrollers. Section 5.4 contains an analysis of an AES implementation whereas Section 5.5 shows how stochastic methods can be applied at a boolean masking scheme, i.e., the most general case for higher order analysis.

5.3.2 Distance between the true function h_t and functions $h' \in \mathcal{F}_{u;t}$

The central goal is to estimate the distribution of the random vector $(I_{t_1}(x,k),\ldots,I_{t_m}(x,k))$ where $t_1 < \cdots < t_m$ are different time instants that are part of the side channel measurements.

Definition 5.1. $\|\cdot\|\colon \mathbb{R}^n \to \mathbb{R}$ is also called the *Euclidean norm*, that is $\|(z_1,z_2,...,z_n)\|^2 = \sum_{j=1}^n z_j^2$.

Definition 5.2. The term $\widetilde{f}$ denotes an estimator of a value f.

The random variables R_t, X and K are defined over the same probability space $(\Omega, \mathfrak{A}, \mathbb{P})$ of sample space Ω, σ-algebra $\mathfrak{A}$ consisting of subsets of Ω, and probability measure $\mathbb{P}$ on $\mathfrak{A}$. More precisely, $R_t \colon \Omega \to \mathbb{R}$; $X \colon \Omega \to \{0,1\}^d$ (random data parts) and $K \colon \Omega \to \{0,1\}^s$ (random subkey). By assumption, the random variables R_t, X and K are independent. For the sake of readability in (5.5), for instance, $\mathbb{E}$ is used as abbreviation for $\mathbb{E}_{X,R_t,K=k}$.

Theorem 5.1 that is referred from [126] will turn out to be crucial for the following. In the following h_t^* will always denote an element in $\mathcal{F}_{u;t}$ where (5.4) and (5.5) attain their minimum.

Theorem 5.1. *Let $k \in \{0,1\}^s$ denote the correct subkey.*
(i) For each $h' \in \mathcal{F}_{u;t}$ we have

$$\begin{aligned} &\mathbb{E}\left(\left(I_t(X,k) - h'(X,k)\right)^2\right) - \mathbb{E}\left(\left(I_t(X,k) - h_t(X,k)\right)^2\right) \\ &= \mathbb{E}_X\left(\left(h_t(X,k) - h'(X,k)\right)^2\right) \geq 0 \end{aligned} \tag{5.3}$$

where $\mathbb{E}_X(\cdot)$ denotes the expectation with respect to the random variable X, i.e. the right-hand term equals
$\sum_{x \in \{0,1\}^d} \mathrm{Prob}(X = x)\left(h_t(x,k) - h'(x,k)\right)^2$.

$$(ii)\ \ \mathbb{E}_X\left(\left(h_t(X,k) - h_t^*(X,k)\right)^2\right) = \min_{h' \in \mathcal{F}_{u;t}} \mathbb{E}_X\left(\left(h_t(X,k) - h'(X,k)\right)^2\right) \tag{5.4}$$

implies

$$\mathbb{E}\left(\left(I_t(X,k) - h_t^*(X,k)\right)^2\right) = \min_{h' \in \mathcal{F}_{u;t}} \mathbb{E}\left(\left(I_t(X,k) - h'(X,k)\right)^2\right). \tag{5.5}$$

(iii) Let $t_1 < t_2 \cdots < t_m$. If $h'_j \in \mathcal{F}_{u,t_j}$ for all $j \leq m$ then

$$\begin{aligned} &\mathbb{E}\left(\| \left(I_{t_1}(X,k) - h'_1(X,k), \ldots, I_{t_m}(X,k) - h'_m(X,k)\right) \|^2\right) \\ = \ &\mathbb{E}\left(\| \left(I_{t_1}(X,k) - h_{t_1}(X,k), \ldots, I_{t_m}(X,k) - h_{t_m}(X,k)\right) \|^2\right) + \\ &\sum_{j=1}^{m} \mathbb{E}_X\left(\left(h_{t_j}(X,k) - h'_j(X,k)\right)^2\right). \end{aligned} \tag{5.6}$$

Proof. Because of (5.1), $\mathbb{E}(R_t) = 0$ and the independence of X and R_t it is

$$\begin{aligned}
&\mathbb{E}\left((I_t(X,k) - h'(X,k))^2\right) - \mathbb{E}\left((I_t(X,k) - h_t(X,k))^2\right) \\
&= \mathbb{E}\left((h_t(X,k) + R_t - h'(X,k))^2\right) - \mathbb{E}\left(\left(R_t^2\right)\right) \\
&= \mathbb{E}\left(h_t(X,k)^2\right) - \mathbb{E}\left(2h'(X,k)h_t(X,k)\right) + \mathbb{E}\left(h'(X,k)^2\right) \\
&= \mathbb{E}\left((h_t(X,k) - h'(X,k))^2\right) \\
&= \mathbb{E}_X\left((h_t(X,k) - h'(X,k))^2\right) \geq 0
\end{aligned}$$

which proves (i). (ii) and (iii) are immediate consequences from (i). □

Remarks 5.1. (i) If X is uniformly distributed on $\{0,1\}^d$ then the term $\mathbb{E}_X((h_t(X,k) - h'(X,k))^2) = \frac{1}{2^d} \sum_{x \in \{0,1\}^d} (h_t(x,k) - h'(x,k))^2$ equals (apart from a constant) the squared Euclidean distance of 2^d dimensional vectors $h_t(\cdot,k)$ and $h'(\cdot,k)$.
(ii) $h_t^*(\cdot,k)$ is the orthogonal projection of $h_t(\cdot,k)$ onto $\mathcal{F}_{u;t}$.
(iii) It is natural to select the function $h_t^* \in \mathcal{F}_{u;t}$ that is 'closest' to h_t, i.e., that minimizes $\mathbb{E}_X((h_t(X,k) - h'(X;k))^2)$ on $\mathcal{F}_{u;t}$. Theorem 5.1 says that h_t^* can alternatively be characterized by another minimum property (5.5).
(iv) The approximators $\widetilde{h}_{t_1}^*, \ldots, \widetilde{h}_{t_m}^*$ can be determined separately. Section 5.4.1 provides a concrete formula to estimate the unknown coefficients $\beta_{0,t}^*, \ldots, \beta_{u-1,t}^*$ of h_t^* with respect to the base $g_{t,0}, \ldots, g_{t,u-1}$.
(v) An appropriate choice of the functions $g_{t,0}, \ldots, g_{t,u-1}$, i.e., of $\mathcal{F}_{u;t}$, is essential for the success rate of the attack. Of course, the vector subspace $\mathcal{F}_{u;t}$ should have a small distance to the unknown function h_t. An appropriate choice may require some insight in the qualitative behaviour of the side channel observables. From the theoretical point of view $\mathcal{F}_{u_1,t} \subseteq \mathcal{F}_{u_2,t}$ implies that $h_{u_2,t}^*$ is at least as good $h_{u_1,t}^*$.

5.3.3 The EIS Property

The basic problem to be solved is that the function $h_t(\cdot,\cdot)$ is typically not known in advance. The estimation of $h_t(\cdot,\cdot)$ can be achieved in a profiling stage. By using the methodology of Template Attacks [40] this

means that $h_t(\cdot,\cdot)$ is profiled for each combination of $k \in \{0,1\}^s$ and $x \in \{0,1\}^d$ ending up with a profiling space of $\{0,1\}^{s+d}$. Properties of typical block cipher design can reduce the profiling space to $\{0,1\}^s$ if $d = s$ and the function h_t has Property (EIS) [126].

Definition 5.3. Let V denote an arbitrary set and let $\phi\colon \{0,1\}^d \times \{0,1\}^s \to V$ be a mapping for which the images $\phi\left(\{0,1\}^d \times k'\right) \subseteq V$ are equal for all subkeys $k' \in \{0,1\}^s$. The function h_t is said to have *Property (EIS) ('equal images under different subkeys')* if $h_t = \overline{h}_t \circ \phi$ for a suitable mapping $\overline{h}_t\colon V \to \mathbb{R}$, i.e., $h_t(x,k)$ can be expressed as a function of $\phi(x,k)$.

> **Example 5.4.** $d = s$, $\phi(x,k) := x \odot k$ where $\odot$ denotes any group operation on $\{0,1\}^d =: V$ (e.g., '$\oplus$').

Lemma 5.2. *Assume that $h_t(\cdot,\cdot)$ has property (EIS). Then for any pair $(x',k') \in \{0,1\}^d \times \{0,1\}^s$ there exists an element $x'' \in \{0,1\}^d$ with $h_t(x',k') = h_t(x'',k)$.*

Proof. By assumption, $\phi\left(\{0,1\}^d, k\right) = \phi\left(\{0,1\}^d, k'\right)$. Consequently, there exists an $x'' \in \{0,1\}^d$ with $\phi(x'',k) = \phi(x',k')$ and hence $h_t(x'',k) = h_t(x',k')$. □

If EIS property is fulfilled then the distance between the unknown function h_t and the closest approximation h_t^* is minimal for the correct key value at key recovery. Otherwise if EIS property is not fulfilled this may result in the observation that key recovery points to a wrong key hypothesis. Whether the invariance assumption (EIS) is really justified for $h_t(\cdot,\cdot)$ may be checked by repeating the profiling stage with another subkey.

If considerations on the fundamental properties of the physical observables suggest that $h_t(\cdot,\cdot)$ meets (at least approximately) the invariance property EIS it is reasonable to select functions g_{tl} that allow representations of the form $g_{tl} = \overline{g}_{tl} \circ \phi$ with $\overline{g}_{tl}\colon V \to \mathbb{R}$. Then

$$h_t^* = \overline{h}_t^* \circ \phi \text{ with } \overline{h}_t^*(y) := \sum_{l=0}^{u-1} \beta_{tl} \overline{g}_{tl}(y) \tag{5.7}$$

(see Section 5.4.1). As an important consequence it is fully sufficient to determine $\widetilde{h}_t^*(\cdot,k) \in \mathcal{F}_{u;t}$ for any single subkey $k \in \{0,1\}^s$, which is

an enormous advantage over a pure template attack which requires 2^{d+s} templates. An advanced template attack that exploits EIS, however, suffices also with 2^s templates. If feasible it is recommended that plaintexts stem from a uniform distribution so that deviations $|h_t(x,k) - h_t^*(x,k)|$ count equally to the distance for all (x,k).

5.3.4 Profiling Phase

For profiling it is assumed that the adversary has access to a cryptographic device with a known key and can measure instantaneous physical observables of the cryptographic operation while it operates on known data. It is even possible to do profiling without knowing the key which is also explained in this section. Then the recovery of the key is a by-product of the profiling phase. Due to the algorithmic properties of the cipher it is assumed that the cryptographic key is composed of (or used in) multiple small portions during computation, i.e., so called subkeys. This subsection details on profiling of *one* subkey k. This procedure has to be repeated for other subkeys in order to obtain profiling characteristics for a sufficient number of subkeys that finally can yield a total break of the secret key at the key recovery phase.

This chapter introduces two main methods as part of the stochastic model: the *minimum principle* and the *maximum likelihood principle*. Both differ in certain parts of the profiling and key recovery phase. While it is sufficient for the minimum principle to profile the deterministic side channel leakage (Algorithm 5.1) and to select relevant instants, the maximum likelihood principle additionally requires an estimation of the noise (Algorithm 5.2). Table 5.1 summarizes the differences. Note that for a first try the maximum likelihood principle may also use all N measurements for the estimation of the deterministic part and the selection of instants.

Estimation on the deterministic part of the side channel

Here it is explained how one finds approximators of $h_t(\cdot,\cdot)$, or more precisely, of $h_t^*(\cdot,\cdot)$ and the distribution of the noise vector $(R_{t_1},\ldots,R_{t_m})$. The 'relevant parts' $x_1, x_2, \ldots, x_N$ (i.e., input for the function h_t) of known data parts are interpreted as realization of independent random variables $X_1, X_2, \ldots, X_N$ that are distributed as X. The Law of Large

Table 5.1: Tasks for the minimum principle and the maximum likelihood principle. Note that the number of measurements N is split into two disjoint subsets N_1 and N_2 with $N_1 + N_2 = N$ for the maximum likelihood principle.

Method	Minimum Principle	Maximum Likelihood Principle
Estimation of the deterministic part:	yes (N samples)	yes (N_1 samples)
Selection of instants:	yes (N samples)	yes (N_1 samples)
Estimation of the noise:	no	yes (N_2 samples)

Numbers implies

$$\frac{1}{N}\sum_{i=1}^{N}\left(i_t(x_i,k)-h'(x_i,k)\right)^2 \overset{N\to\infty}{\longrightarrow} \mathbb{E}\left(\left(I_t(X,k)-h'(X,k)\right)^2\right) \tag{5.8}$$

with probability 1 for any $h'\colon \{0,1\}^d \times \{0,1\}^s \to \mathbb{R}$. Here $i_t(x_i,k)$ denotes the measurement at time t for curve i which belongs to the data $x_i \in \{0,1\}^d$.

Definition 5.4. In this work, terms $\vec{b}^T$ and $\mathbf{A}^T$ stand for the transpose of the vector $\vec{b}$ and the matrix $\mathbf{A}$, respectively. An element in the i-th row and j-th column of a matrix $\mathbf{A}$ is denoted by a_{ij}. The row vector of a matrix $\mathbf{A}$ is denoted with $\vec{a}_i^{row}$ while a column vector of a matrix $\mathbf{A}$ is denoted by $\vec{a}_j^{col}$.

For the following explanations it turned out to be more appropriate to define a matrix for the measurement values. Let $\mathbf{I}$ be the $p \times N$ matrix defined by the N measurement vectors $\vec{i}_i \in \mathbb{R}^p$ with $1 \le i \le N$.

$$\mathbf{I} = \begin{pmatrix} i_{11}(x_1,k) & i_{12}(x_1,k) & . & . & . & . & i_{ip}(x_1,k) \\ i_{21}(x_2,k) & i_{22}(x_2,k) & . & . & . & . & i_{2p}(x_2,k) \\ . & . & & & & & . \\ . & . & & & & & . \\ . & . & & & & & . \\ i_{N1}(x_N,k) & i_{N2}(x_N,k) & . & . & . & . & i_{Np}(x_N,k) \end{pmatrix} \tag{5.9}$$

The i-th row vector $\vec{i}_i^{row}$ is the original i-th measurement vector $\vec{i}_i^T \in \mathbb{R}^p$. For each row vector there is an associated data item $x_i \in \{0,1\}^d$ which

is the plaintext (or ciphertext). The j-th column vector of matrix $\mathbf{I}$ is $\vec{i_j}^{col} := (i_{1j}(x_1,k), i_{2j}(x_2,k), \ldots, i_{Nj}(x_N,k))^T$ and includes all measurement values for one instant j. In the following the notation $i_{ij}(x_i,k)$ is used instead of $i_t(x_i,k)$. Also $\mathcal{F}_{u;t}$ is replaced by $\mathcal{F}_{u_j;j}$ outlining the fact that the choice of vector subspaces and its dimension u_j can depend on j.

Profiling of the deterministic part is done separately for each instant, i.e., for profiling purposes the column vector $\vec{i_j}^{col}$ is the starting point. The fitting problem to be solved is to find coefficients $\vec{\beta_j} := (\beta_{j0}, \ldots, \beta_{j,u_j-1})^T$ of (5.2) such that the measurement quantities i_{ij} and control quantities $g_{j0}(x_i,k), \ldots, g_{j,u_j-1}(x_i,k)$ are linked by

$$i_{ij}(x_i,k) = \beta_{j0} + \sum_{l=1}^{u_j-1} \beta_{jl} g_{jl}(x_i,k) \; \forall i \in \{1,\ldots,N\}$$

in $\mathcal{F}_{u_j;j}$. This yields an approximation on the deterministic part of the side channel for the instantiation of the stochastic variable I_j at sampled instant j given the control variables $g_{j0}(x,k), \ldots, g_{j,u_j-1}(x,k)$ with $g_{j0}(x,k)$ being the constant function 1. The coefficient β_{j0} gives the expectation value of the non-data dependent signal part and the coefficients β_{jl} with $l \neq 0$ are the data dependent signal portions.

In the stochastic model, the coefficients $\vec{\beta_j}$ are approximated with least squares estimates. For this, Proposition 2.14 is applied in the following setting: $m := u_j - 1$, $n := N$, $\alpha := \beta_{j0}$, $\vec{\beta} := (\beta_{j1}, \ldots, \beta_{ju_j})^T$ and the vector $\vec{y} := (i_{1j}, \ldots, i_{Nj})^T$. The $N \times u_j$ design matrix is

$$\mathbf{M} = \begin{pmatrix} 1 & g_{j1}(x_1,k) & g_{j2}(x_1,k) & . & . & . & g_{j,u_j-1}(x_1,k) \\ 1 & g_{j1}(x_2,k) & g_{j2}(x_2,k) & . & . & . & g_{j,u_j-1}(x_2,k) \\ . & . & & & & & . \\ . & . & & & & & . \\ . & . & & & & & . \\ 1 & g_{j1}(x_N,k) & g_{j2}(x_N,k) & . & . & . & g_{j,u_j-1}(x_N,k) \end{pmatrix}. \quad (5.10)$$

Application of Proposition 2.14 yields the least square estimates of $\vec{\beta_j}$ by replacing $\beta^*_{j0} := \hat{\alpha}$, $\left(\beta^*_{j1}, \ldots, \beta^*_{j,u_j-1}\right) := \hat{\vec{\beta}}$ as

$$\vec{\beta^*_j} = \left(\mathbf{M}^T\mathbf{M}\right)^{-1} \mathbf{M}^T \vec{i_j}^{col} \quad (5.11)$$

provided that $\mathbf{M}^T\mathbf{M}$ is regular.

These minimizing elements $\vec{\beta_j^*}$ are unique if the system of basis vectors spanning $\mathcal{F}_{u_j;j}$ is linearly independent. $\vec{\beta_j^*}$ minimizes the sum of squares of errors, i.e.,

$$\sum_{i=1}^{N}\left(i_{ij}(x_i,k)-\sum_{l=0}^{u_j-1}\beta_{jl}g_{jl}(x_i,k)\right)^2 = \|\vec{i_j^{col}}-\langle\vec{\beta_j},\vec{g_j}(x_i,k)\rangle\|^2 \quad (5.12)$$

Due to (5.8) one can use the estimator

$$\begin{aligned}\widetilde{h_j}^*(x,k) &= \beta_{j0}^*+\sum_{l=1}^{u_j-1}\beta_{jl}^*g_{jl}(x,k) \qquad (5.13)\\ &= \sum_{l=0}^{u_j-1}\beta_{jl}^*g_{jl}(x,k)\end{aligned}$$

for the deterministic part of the side channel as far as it depends on x and k. The algorithm of the estimation on h_j^* is summarized in Algorithm 5.1.

Algorithm 5.1 Estimation of the deterministic part

Input: (i) The $p \times N$ matrix $\mathbf{I}$ containing the N measurement vectors $(\vec{i_1},\cdots\vec{i_N})$ with $\vec{i_i} \in \mathbb{R}^p$ for all $i \in \{1,\ldots,N\}$.
(ii) Known data $x_i \in \{0,1\}^d$ for all $i \in \{1,\ldots,N\}$ and one fixed known subkey $k \in \{0,1\}^s$ at the profiling device.
(iii) For each sampled instant $j \in \{1,\ldots,p\}$: the u_j functions g_{jl} with $0 < l \leq u_j$ that span the vector subspace $\mathcal{F}_{u_j;j}$.

Output: For each sampled instant $j \in \{1,\ldots,p\}$: the real-valued coefficients $\beta_{j0}^*,\ldots,\beta_{j,u_j-1}^*$ for the least squares estimator $\widetilde{h}_j^*(\cdot,\cdot) = \sum_{l=0}^{u_j-1}\beta_{jl}^*g_{jl}(\cdot,\cdot)$.

1: **for** j from 1 to p **do**
2: Compute the design matrix $\mathbf{M}$ according to equation (5.10);
3: Solve equation (5.11) to obtain the u_j dimensional solution vector $\vec{\beta_j^*} = (\beta_{j0}^*,\ldots,\beta_{j,u_j-1}^*)^T$;
4: **end for**

Remarks 5.2. (i) A natural first choice for Algorithm 5.1 is to define *one* vector subspace $\mathcal{F}_u$ that is common for all instants j. Then the solution vectors $\vec{\beta}^*_{j0}, \ldots, \vec{\beta}^*_{j,u-1}$ build a $p \times u$ matrix. The row vectors of this matrix give then the coefficient for one basis vector as a function of time.
(ii) Instead of one fixed subkey, Algorithm 5.1 can also be applied if the subkey k depends on the measurement i.
(iii) If $\widetilde{h}^*_j(\cdot,\cdot)$ has the property EIS profiling needs to be done only for one subkey k.
(iv) Whether the EIS property of $\widetilde{h}^*_j(\cdot,\cdot)$ is indeed (at least approximately) fulfilled can be checked by repeating Algorithm 5.1 with another subkey $k' \neq k$.

Profiling without knowing the key

As already outlined it is even not necessary that the adversary needs to know the subkey k. Then the adversary applies Algorithm 5.1 to all possible subkeys $k' \in \{0,1\}^s$ and computes the respective coefficient vectors $\vec{\beta}^{*'}_{jl}$. If $k' \neq k$ Algorithm 5.1 indeed determines an optimal function $\widetilde{h}^{*'}_j \in \mathcal{F}_{u_j,j}$.

The situation here is quite similar to standard DSCA methods as the measured quantities $\vec{i^{col}_j}$ implicitly depend on the (unknown) correct subkey k. If $k' \neq k$ then it is $g_{jl}(\phi(x_i,k)) = g_{jl}(\phi(x'_i,k')) \neq g_{jl}(\phi(x_i,k'))$ considering outcomes of non-constant functions $g_{jl}(\cdot)$ for which $g_{jl}(0) = 0$ holds. Provided that x_i are uniformly distributed one can expect that coefficients $\beta^{*'}_{jl}$ vanish for non-constant functions $g_{jl}(\cdot)$ if the property EIS is fulfilled. This approach checks for significant non-zero entries in all coefficients of non-constant basis functions and decides in favour of the key hypothesis that yields an absolute maximum value, e.g., of one coefficient or a weighted sum of several coefficients with Algorithm 3.3.

An alternative approach is to use the sum of squared errors as given in equation (5.12) for key recovery. If the number of measurements is sufficiently high so that noise is sufficiently suppressed it is likely that

$$\|\vec{i^{col}_j} - \langle \vec{\beta^*_j}, \vec{g_j}(x_i,k)\rangle\|^2 < \|\vec{i^{col}_j} - \langle \vec{\beta^{*'}_j}, \vec{g_j}(x_i,k')\rangle\|^2 \tag{5.14}$$

holds for some instants j contributing to the deterministic side channel part. Equation (5.14) can be used as statistical test for hypothesis testing on an enlarged time frame, i.e., the adversary just adds these squared

norms for each admissible subkey over all instants, and decides for the subkey for which the sum

$$\frac{1}{N}\sum_{i=1}^{N} \|\vec{i_j}^{col}(x_i,k) - \widetilde{\vec{h}^*}_j(x_i,k')\|^2 \ . \tag{5.15}$$

is minimal (see Section 5.4.1 for an experimental verification).

In fact, the recovery of k is a by-product of the profiling phase which does not cost any additional measurements and reduces the assumptions on the applicability of the profiling stage. This observation could also be used for a direct attack without profiling, which yet requires a sufficient number of measurements. In a more general case if, e.g., the EIS property is only partly fulfilled for the measurement outcomes the methods discussed here are still expected to work, however, the coefficients $\beta_{jl}^{*'}$ do generally not vanish for $k' \neq k$. Once the subkey k is known the next parts of the profiling stage can be carried out yielding to a refined side channel analysis that requires less measurements at key recovery.

Selection of Instants

As result of Algorithm 5.1 one obtains a least square estimation of the deterministic part in the chosen vector subspace $\mathcal{F}_{u_j;j}$ for all j. DSCA signals, however, show typically up only at a few distinct instants at which the measured quantities depend on a basis function of $\mathcal{F}_{u_j;j}$. It is therefore desirable to sort out instants that do not contribute to the deterministic leakage in order to reduce noise as well as the dimension of the characterization problem for the maximum likelihood principle.

Remark 5.1. Concrete, but still heuristic algorithms for instant selection are given in Algorithm 5.7 and Algorithm 5.8. From the current perspective, Algorithm 5.9 is seen as the most promising.

The Estimation of the Noise

As summarized in Table 5.1 one needs an estimation of the noise for the maximum likelihood principle, but not for the minimum principle as result of profiling. It is assumed that the estimation on the deterministic part in Algorithm 5.1 is already carried out by using N_1 measurements. For the estimation of the noise it is strongly recommended that a choice

of instants $(t_1, \ldots, t_m)$ has been carried out beforehand. The notation used for a given set of instants is defined in Definition 5.5.

Definition 5.5. $\vec{R}_{\vec{t}}$ denotes the random vector $(R_{t_1}, \ldots, R_{t_m})$ in the following. Similarly, the abbreviations $\vec{I}_{\vec{t}}(x,k)$, $\vec{i}_{\vec{t}}(x_j,k)$, $\vec{h}_{\vec{t}}(x,k)$ and $\vec{h}^*_{\vec{t}}(x,k)$ are used, where $\vec{t}$ stands for $(t_1, \ldots, t_m)$.

After having determined the approximators $\widetilde{h}^*_{t_1}, \ldots, \widetilde{h}^*_{t_m}$ the adversary uses a complementary set that consists of $N_2 = N - N_1$ measurement curves to estimate the distribution of the m-dimensional random vector $\vec{R}_{\vec{t}} = \vec{I}_{\vec{t}}(X,k) - \vec{h}_{\vec{t}}(X,k)$. In general the components $R_{t_1}, \ldots, R_{t_m}$ of $\vec{R}_{\vec{t}}$ are not independent, and unlike the functions h_{t_j} they hence cannot be guessed separately. In the most general case the adversary interpolates the N_2 vectors $\{\vec{z}_i := \vec{i}_{\vec{t}}(x_i,k) - \widetilde{\vec{h}^*}_{\vec{t}}(x_i,k) \mid N_1 < i \leq N\}$ by a smooth probability density f_0.

For practical purposes it is assumed that the random vector $\vec{R}_{\vec{t}}$ is jointly normally distributed with covariance matrix $\mathbf{C} = (c_{uv})_{1 \leq u,v \leq m}$, i.e. $c_{uv} := \mathbb{E}(R_{t_u} R_{t_v}) - \mathbb{E}(R_{t_u})\mathbb{E}(R_{t_v})$. If the covariance matrix $\mathbf{C}$ is regular the random vector $\vec{R}_{\vec{t}}$ has the m-dimensional density $f_0 := f_{\mathbf{C}}$ with

$$f_{\mathbf{C}} \colon \mathbb{R}^m \to \mathbb{R} \qquad f_{\mathbf{C}}(\vec{z}) = \frac{1}{\sqrt{(2\pi)^m \det \mathbf{C}}} \exp\left(-\frac{1}{2} \vec{z}^T \mathbf{C}^{-1} \vec{z}\right) \tag{5.16}$$

(see (2.9)). Note that the adversary merely has to estimate the components c_{uv} for $u \leq v$ in Algorithm 5.2 since the covariance matrix is symmetric.

5.3.5 Key Recovery Phase

The task of the key recovery phase is to determine the correct subkey value by using N_3 measurements carried out with an unknown subkey $k^\circ \in \{0,1\}^s$ at the target device.

For the entire key recovery the ranking of the candidates for all relevant subkeys has to be available first. The ranking list of subkey candidates is the outcome of the algorithms of the stochastic model in the key recovery phase. Assuming that one plaintext-ciphertext pair is known, 'candidate keys' can be checked by applying a sorting algorithm

Algorithm 5.2 Estimation of the multivariate Noise

Input: (i) The $m \times N_2$ matrix $\mathbf{I}$ containing the N_2 measurement vectors $\vec{i_t} \in \mathbb{R}^m$ for the selected instants $t_1, \ldots, t_m$ of the profiling device.
(ii) Known data $x_i \in \{0,1\}^d$ for all $i \in \{1, \ldots, N_2\}$ and one fixed known subkey $k \in \{0,1\}^s$ at the profiling device.
(iii) Least square estimators $\widetilde{h}_t^*(\cdot,\cdot)$ with coefficients β_{jl}^* for the u_j basis functions with $0 \leq l < u_j$ at each sampled instant j determined with N_1 measurements as result of Algorithm 5.1.

Output: An $m \times m$ covariance matrix $\mathbf{C} = (c_{uv})$.

for i from 1 to N_2 **do**
  for j from 1 to m **do**
    $z_{ij} = i_{ij} - \widetilde{h}_j^*(x_i, k)$;{Compute the noise vector.}
  end for
end for
for u from 1 to m **do**
  for v from 1 to u **do**
    $c_{vu} = c_{uv} = \frac{1}{N_2} \sum_{i=1}^{N_2} z_{iu} z_{iv} - \left(\sum_{i=1}^{N_2} z_{iu}\right)\left(\sum_{i=1}^{N_2} z_{iv}\right)$.
    {Compute the covariance entries.}
  end for
end for

starting with combinations of the most probable subkeys. If a plaintext-ciphertext pair is not available an enhanced assurance in the success of the subkey recovery may be needed which typically requires an increased number of measurements N_3. It is noted that also sieving of candidate subkeys can be applied here, for example, one may apply the stochastic model to an intermediate result that depends on two or more subkeys. After an initial key recovery procedure for each subkey separately, the sieving step takes only those subkey candidates into consideration that have appeared among the top at the initial ranking lists.

The decision strategies and algorithms for the minimum principle and the maximum likelihood principle are detailed below.

Minimum Principle

The minimum principle is exclusively based on the estimation of the deterministic part. The adversary computes the term

$$\alpha_{MP}(x_1,\ldots,x_{N_3};k) := \frac{1}{N_3}\sum_{i=1}^{N_3} \|\vec{i}_{\vec{t}}(x_i,k^\circ) - \widetilde{\vec{h}^*}_{\vec{t}}(x_i,k)\|^2 \ . \tag{5.17}$$

for all possible subkey values $k \in \{0,1\}^s$ and decides for the subkey $k' \in \{0,1\}^s$ that minimizes $\alpha_{MP}(x_1,\ldots,x_{N_3};k)$:

$$k' = \arg\min_{k\in\{0,1\}^s} \alpha_{MP}(x_1,\ldots,x_{N_3};k) \ . \tag{5.18}$$

The minimum principle is motivated by equation (5.12). As the outcome $\vec{i}_{\vec{t}}(x_i,k^\circ)$ implicitly depends on x_i and k° the sum of squared errors is assumed to be minimal for the correct subkey k° provided that the basis functions are appropriately chosen and that EIS property is fulfilled. Accordingly, the guess of a subkey k is probably not correct if high values of the term $\|\vec{i}_{\vec{t}}(x_i,k^\circ) - \widetilde{\vec{h}^*}_{\vec{t}}(x_i,k')\|^2$ are attained. The adversary is successful if $k' = k^\circ$.

Maximum Likelihood Principle

The stochastic model is based on the assumption that the m-variate noise $\vec{R}_{\vec{t}}$ is obtained by computing $\vec{R}_{\vec{t}} = \vec{I}_{\vec{t}}(x,k) - \vec{h}_{\vec{t}}(x,k)$ given $(x,k) \in \{0,1\}^d \times \{0,1\}^s$ and $\mathbb{E}(R_{t_j}) = 0$ for each $j \le m$. $\vec{R}_{\vec{t}}$ has the density $f_0 \colon R^m \to [0,\infty)$. In practice one assumes $f_0 = f_{\mathbf{C}}$, i.e., an m-variate Gaussian probability function with a suitable covariance matrix $\mathbf{C}$.

After having observed N_3 measurement curves with known parts $x_1,\ldots,x_{N_3}$ and unknown subkey k° at the target device the adversary evaluates the product

$$\alpha_{MLP}(x_1,\ldots,x_{N_3};k) := \prod_{i=1}^{N_3} \widetilde{f}_0\left(\vec{i}_{\vec{t}}(x_i,k^\circ) - \widetilde{\vec{h}^*}_{\vec{t}}(x_i,k)\right) \tag{5.19}$$

for each subkey $k \in \{0,1\}^s$ where $\widetilde{f}_0$ denotes the approximation of the exact density f_0 that the adversary has determined in the profiling phase. It is assumed that the density $\widetilde{f}_0$ is a 'good' characterization of the

Algorithm 5.3 Minimum Principle

Input: (i) The $m \times N_3$ matrix $\mathbf{I}$ containing the N_3 measurement vectors $\vec{i}_{\vec{t}} \in \mathbb{R}^m$ for the selected instants $t_1, \ldots, t_m$ of the target device.
(ii) Known data $x_i \in \{0,1\}^d$ for all $i \in \{1, \ldots, N_3\}$ and one fixed unknown subkey $k^\circ \in \{0,1\}^s$ at the target device.
(iii) Least square estimators $\widetilde{h}^*_{\vec{t}}(\cdot,\cdot)$ with coefficients β^*_{jl} for the u_j basis functions with $0 \leq l < u_j$ at each selected instant $j \in \{t_1, \ldots, t_m\}$.
(iv) A standard indexing (in ascending order) algorithm $I : \underbrace{\mathbb{R} \times \cdots \times \mathbb{R}}_{2^s} \rightarrow P$ with P being the set of all possible permutations of the elements $\{0, \ldots, 2^s - 1\}$.
Output: a permutation of subkey hypotheses

$$\pi = \begin{pmatrix} 0 & 1 & . & . & . & 2^s - 1 \\ \tilde{k}_0 & \tilde{k}_1 & . & . & . & \tilde{k}_{2^s-1} \end{pmatrix}$$

such that $s_{\tilde{k}_0} \leq s_{\tilde{k}_1} \leq \cdots \leq s_{\tilde{k}_{2^s-1}}$ corresponding to $\mathbb{P}(\tilde{k}_0 = k^\circ) \geq \mathbb{P}(\tilde{k}_1 = k^\circ) \geq \ldots \geq \mathbb{P}(\tilde{k}_{2^s-1} = k^\circ)$.

```
1: for k from 0 to 2^s − 1 do
2:    s_k = 0;
3:    for i from 1 to N_3 do
4:       for j from 1 to m do
5:          s_k = s_k + ( i_ij − h̃*_j(x_i, k) )^2 ;
6:       end for
7:    end for
8: end for
9: {k̃_0, ..., k̃_{2^s−1}} ← I(s_0, ..., s_{2^s−1});
```

remaining noise and that this noise itself does not heavily depend on the subkeys. Note that $\vec{i}_{\vec{t}}(x_i, k^\circ)$ are instantiations of observables that depend implicitly on the correct subkey k°. The adversary hence decides for the subkey

$$k' = \arg \max_{k \in \{0,1\}^s} \alpha_{MLP}(x_1, \ldots, x_{N_3}; k) \tag{5.20}$$

that maximizes (5.19) (maximum likelihood principle). The adversary

is successful if $k' = k^\circ$.

One may interprete the procedure used in (5.19) as shifting the density of $\vec{I}_{\vec{t}}(x,k)$ by $\widetilde{\vec{h}^*}_{\vec{t}}(x,k)$ to obtain the density of $\vec{R}_{\vec{t}}$ for which a density is already estimated in the profiling phase. If $k \neq k^\circ$ the displacement of $\widetilde{\vec{h}^*}_{\vec{t}}(x,k)$ does probably not result in the high-probability areas of $\widetilde{f}_0$. However, if $k = k^\circ$ the displacement is assumed to fit approximately to the real noise distribution so that $f_0(\vec{i}_{\vec{t}}(x_i,k^\circ) - \widetilde{\vec{h}^*}_{\vec{t}}(x_i,k^\circ))$ achieves high probability outcomes in average. Equation (5.19) considers the outcomes of the 'shifted' probability densites for all N_3 measurements.

Algorithm 5.4 summarizes the requirements and processing steps for the application of the maximum likelihood principle with a multivariate Gaussian distribution.

Remark 5.2. As the covariance matrix $\mathbf{C}$ is identical for all subkeys it is sufficient to compute the exponent of the Gaussian distribution (5.16) in Algorithm 5.4 to obtain the relative probabilities for the subkey ranking. Note that maximizing the Gaussian distribution in (5.16) is then equivalent to minimizing the exponent $\vec{z}^T\mathbf{C}^{-1}\vec{z}$ as used in Algorithm 5.4.

Template attacks aim at h_t itself whereas the stochastic model estimates h_t^*. Hence the key recovery efficiency of the template attacks gives an upper bound for the stochastic approach. However, if the vector subspace $\mathcal{F}_{u_j;j}$ has been chosen appropriately this efficiency gap can be small.

Algorithm 5.4 Maximum Likelihood Principle (multivariate Gaussian distribution)

Input: (i) The $m \times N_3$ matrix $\mathbf{I}$ containing the N_3 measurement vectors $\vec{i_t} \in \mathbb{R}^m$ for the selected instants $t_1, \ldots, t_m$ of the target device.
(ii) Known data $x_i \in \{0,1\}^d$ for all $i \in \{1, \ldots, N_3\}$ and one fixed unknown subkey $k^\circ \in \{0,1\}^s$ at the target device.
(iii) Least square estimators $\widetilde{h}^*_{\vec{t}}(\cdot,\cdot)$ with coefficients β^*_{jl} for the u_j basis functions with $0 \leq l < u_j$ at each selected instant $j \in \{t_1, \ldots, t_m\}$.
(iv) The $m \times m$ inverse covariance matrix $\mathbf{C}^{-1} = (c^{-1}_{uv})$
(v) A standard indexing (in ascending order) algorithm $I : \underbrace{\mathbb{R} \times \cdots \times \mathbb{R}}_{2^s} \to P$ with P being the set of all possible permutations of the elements $\{0, \ldots, 2^s - 1\}$.
Output: a permutation of subkey hypotheses

$$\pi = \begin{pmatrix} 0 & 1 & . & . & . & 2^s - 1 \\ \tilde{k}_0 & \tilde{k}_1 & . & . & . & \tilde{k}_{2^s-1} \end{pmatrix}$$

such that $s_{\tilde{k}_0} \leq s_{\tilde{k}_1} \leq \cdots \leq s_{\tilde{k}_{2^s-1}}$ that corresponds to $\mathbb{P}(\tilde{k}_0 = k^\circ) \geq \mathbb{P}(\tilde{k}_1 = k^\circ) \geq \ldots \geq \mathbb{P}(\tilde{k}_{2^s-1} = k^\circ)$.

1: **for** k from 0 to $2^s - 1$ **do**
2: $s_k = 0$;
3: **for** i from 1 to N_3 **do**
4: **for** j from 1 to m **do**
5: $z_j = i_{ij} - \widetilde{h}^*_j(x_i, k)$;
6: **end for**
7: $s_k = s_k + \sum_{u=1}^m z_u \left(\sum_{v=1}^m c^{-1}_{uv} z_v\right)$;
8: **end for**
9: **end for**
10: $\{\tilde{k}_0, \ldots, \tilde{k}_{2^s-1}\} \leftarrow I(s_0, \ldots, s_{2^s-1})$;

5.3.6 Generalization to Masked Implementations

In response to DSCA developers of cryptographic implementations may include randomization techniques such as secret splitting or masking schemes, e.g., [39, 43]. These randomization techniques shall prevent from predicting any relevant bit in any cycle of the implementation. As result, statistical tests using physical observables at *one* instant, i.e., first order side channel analysis, cannot be assumed to be successfully applied in key recovery.

The underlying assumption for the stochastic model in equation (5.1) is not appropriate if the device under test applies algorithmic masking mechanisms that use internal (pseudo-) random numbers. However, (5.1) allows a straight-forward generalization. The term $h_t\colon \{0,1\}^d \times \{0,1\}^s \to \mathbb{R}$ can be replaced by $h_t\colon \{0,1\}^d \times \{0,1\}^v \times \{0,1\}^s \to \mathbb{R}$ where the second argument denotes the random number that is used for masking. This generalized model assumes that the adversary measures a physical observable $I_t(x,y,k)$ at time t that additionally depends on a mask $y \in \{0,1\}^v$:

$$I_t(x,y,k) = h_t(x,y,k) + R_t \tag{5.21}$$

The first summand $h_t(x,y,k)$ quantifies the deterministic part of the measurement as far it depends on x, y, and k. The term R_t denotes a random variable that does not depend on x, y, and k and fulfills $\mathbb{E}(R_t) = 0$. Y denotes a new random variable that is independent of X and R_t and models the random numbers used for masking.

The EISM Property

Moreover, in Definition 5.3 the function ϕ can be simply replaced by $\phi\colon \{0,1\}^d \times \{0,1\}^v \times \{0,1\}^s \to V$.

Definition 5.6. Let V denote an arbitrary set and let $\phi\colon \{0,1\}^d \times \{0,1\}^v \times \{0,1\}^s \to V$ be a mapping for which the images $\phi\left(\{0,1\}^d \times \{0,1\}^v \times \{k'\}\right) \subseteq V$ are equal for all subkeys $k' \in \{0,1\}^s$. The function h_t is said to have Property (EISM) ('equal images under different subkeys with masking') if $h_t = \overline{h}_t \circ \phi$ for a suitable mapping $\overline{h}_t\colon V \to \mathbb{R}$, i.e., if $h_t(x,y,k)$ can be expressed as a function of $\phi(x,y,k)$.

> **Example 5.5.** $d = v = s$, $\phi(x,y,k) := x \odot y \odot k$ where $\odot$ denotes any group operation on $V := \{0,1\}^d$ (e.g., '$\odot = \oplus$').

Profiling

Under the assumption that the developer has access to the random numbers used for masking profiling works similarly as described in Section 5.3.4. For the estimation of the deterministic part at a masked implementation, a real vector subspace $\mathcal{F}_{u;t}$ is used that is spanned by u known functions $g_{tl}\colon \{0,1\}^d \times \{0,1\}^v \times \{0,1\}^s \to \mathbb{R}$ for each instant t:

$$\mathcal{F}_{u;t} := \{h'\colon \{0,1\}^d \times \{0,1\}^v \times \{0,1\}^s \to \mathbb{R} \mid \sum_{l=0}^{u-1} \beta_l g_{tl} \text{ with } \beta_l \in \mathbb{R}\} \tag{5.22}$$

In the presence of masking the vector subspace is reasonably spanned by two (or more) intermediate results that occur during computation. This yields a joint density at two (or more) intermediate results that allows for key recovery.

> **Example 5.6.** One choice for profiling is to choose the n-bit intermediate results y and $x \oplus y \oplus k$ in case of boolean masking. One may define a $2n+1$ dimensional vector subspace that is spanned by the function 1 and the single bits of y and $x \oplus y \oplus k$.

Once the vector subspace is defined profiling can be carried out in the same way as in Section 5.3.4 yielding a set of instants, the estimator $\widetilde{\vec{h}^*}_{\vec{t}}$, and for the maximum likelihood principle additionally a density $\widetilde{f}_0\colon \mathbb{R}^m \to \mathbb{R}$.

Key Recovery

In the key recovery phase, however, knowledge of the masking numbers $y_1, \ldots, y_{N_3}$ cannot be assumed. Note that the measured quantities $\vec{i}_{\vec{t}}(x_i, y_i, k^\circ)$ depend on two unknown values, the ever-changing value y_i and the fixed value k°.

From a logical point of view masking reduces the adversary's information. At a non-masked implementation the adversary is able to predict any intermediate result if $k = k^\circ$. Therefore the adversary is able to estimate on $\widetilde{\vec{h}^*}_{\vec{t}}(x_i, k)$ and inserts the result in equation (5.17) or (5.19)

to obtain a probability measure to indeed observe each subkey k. At a masked implementation the actual intermediate result has to be treated as an unknown number in the most general case. Instead of *one* intermediate result and therefore *one* estimated density a masked intermediate result can attain *all* outcomes $\widetilde{\vec{h}^*}_{\vec{t}}(x_i, 0, k)$ up to $\widetilde{\vec{h}^*}_{\vec{t}}(x_i, 2^v - 1, k)$. The best adversarial strategy is to use *all* possible outcomes weighted with the probability for each random number $y' \in \{0,1\}^v$. If these random numbers are unbiased and independent then $\mathbb{P}(y_i = y') = 2^{-v}$ for all $i \leq N_3$ and $y' \in \{0,1\}^v$.

Minimum Principle

The adversary evaluates

$$\alpha_{MP}(x_1, \ldots, x_{N_3}; k) := \frac{1}{N_3} \sum_{i=1}^{N_3} \min_{y' \in \{0,1\}^v} \|\vec{i}_{\vec{t}}(x_i, y_i, k^\circ) - \widetilde{\vec{h}}^*_{\vec{t}}(x_i, y', k)\|^2 \tag{5.23}$$

and decides for the subkey

$$k' = \arg \min_{k \in \{0,1\}^s} \alpha_{MP}(x_1, \ldots, x_{N_3}; k) \tag{5.24}$$

that minimizes $\alpha_{MP}(x_1, \ldots, x_{N_3}; k)$. Equations (5.23) and (5.24) are referred to as the minimum principle in the presence of masking. Small values of the squared Euclidean norm $\|\vec{i}_{\vec{t}}(x_i, y_i, k^\circ) - \widetilde{\vec{h}}^*_{\vec{t}}(x_i, y', k)\|^2$ indicate small deviations of the side channel leakage from the deterministic part and therefore enhance the probability for indeed observing the event of $y' = y_i$ and $k = k^\circ$. Accordingly, the guess of a subkey k is probably not correct if high values of the term $\|\vec{i}_{\vec{t}}(x_i, y_i, k^\circ) - \widetilde{\vec{h}}^*_{\vec{t}}(x_i, y', k)\|^2$ are attained for all possible masks.

Algorithm 5.5 Minimum Principle in the presence of masking

Input: (i) The $m \times N_3$ matrix $\mathbf{I}$ containing the N_3 measurement vectors $\vec{i_t} \in \mathbb{R}^m$ for the selected instants $t_1, \ldots, t_m$ of the target device.
(ii) Known data $x_i \in \{0,1\}^d$ for all $i \in \{1, \ldots, N_3\}$ and one fixed unknown subkey $k^\circ \in \{0,1\}^s$ at the target device.
(iii) Least square estimators $\widetilde{h}^*_{\vec{t}}(\cdot,\cdot,\cdot)$ with coefficients β^*_{jl} for the u_j basis functions with $0 \leq l < u_j$ at each selected instant $j \in \{t_1, \ldots, t_m\}$.
(iv) A standard indexing (in ascending order) algorithm $I : \underbrace{\mathbb{R} \times \cdots \times \mathbb{R}}_{2^s} \to P$ with P being the set of all possible permutations of the elements $\{0, \ldots, 2^s - 1\}$.
Output: a permutation of subkey hypotheses

$$\pi = \begin{pmatrix} 0 & 1 & . & . & . & 2^s - 1 \\ \tilde{k}_0 & \tilde{k}_1 & . & . & . & \tilde{k}_{2^s-1} \end{pmatrix}$$

such that $s_{\tilde{k}_0} \leq s_{\tilde{k}_1} \leq \cdots \leq s_{\tilde{k}_{2^s-1}}$ corresponding to $\mathbb{P}(\tilde{k}_0 = k^\circ) \geq \mathbb{P}(\tilde{k}_1 = k^\circ) \geq \ldots \geq \mathbb{P}(\tilde{k}_{2^s-1} = k^\circ)$.

1: **for** k from 0 to $2^s - 1$ **do**
2: $s_k = 0$;
3: **for** i from 1 to N_3 **do**
4: **for** y' from 0 to $2^v - 1$ **do**
5: $t_{y'} = 0$;
6: **for** j from 1 to m **do**
7: $t_{y'} = t_{y'} + \left(i_{ij} - \widetilde{h}^*_j(x_i, y', k)\right)^2$;
8: **end for**
9: **end for**
10: $s_k = s_k + \min_{y' \in \{0,1\}^v} t_{y'}$;
11: **end for**
12: **end for**
13: $\{\tilde{k}_0, \ldots, \tilde{k}_{2^s-1}\} \leftarrow I(s_0, \ldots, s_{2^s-1})$;

Maximum Likelihood Principle

The adversary hence decides for the subkey

$$k' = \arg \max_{k \in \{0,1\}^s} \alpha_{MLP}(x_1, \ldots, x_{N_3}; k) \tag{5.25}$$

that maximizes the term $\alpha_{MLP} := \alpha_{MLP}(x_1, \ldots, x_{N_3}; k)$

$$\alpha_{MLP} := \prod_{i=1}^{N_3} \sum_{y' \in \{0,1\}^v} \mathbb{P}(y_i = y') \widetilde{f}_{\mathbf{C}} \left(\vec{i}_{\vec{t}}(x_i, y_i, k^\circ) - \widetilde{\vec{h}}^*_{\vec{t}}(x_i, y', k) \right) \tag{5.26}$$

among all $k \in \{0,1\}^s$. The mixture of densities on the right-hand side of (5.26) also depends on the unknown random numbers $y_1, \ldots, y_{N_3}$.

Maximizing the term $\alpha_{MLP}(x_1, \ldots, x_{N_3}; k)$ in (5.26) is equivalent to maximizing $\ln(\alpha_{MLP}(x_1, \ldots, x_{N_3}; k))$. By using $\vec{z}_{i,y',k} = \vec{i}_{\vec{t}}(x_i, y_i, k^\circ) - \widetilde{\vec{h}}^*_{\vec{t}}(x_i, y', k)$ in the multivariate Gaussian density and neglecting constant factors of the Gaussian distribution in equation (5.16), $\ln(\alpha_{MLP}) := \ln(\alpha_{MLP}(x_1, \ldots, x_{N_3}; k))$ results in

$$\ln(\alpha_{MLP}) = \sum_{i=1}^{N_3} \ln \left(\sum_{y' \in \{0,1\}^v} \mathbb{P}(y_i = y') \exp \left(-\frac{1}{2} \vec{z}^T_{i,y',k} \mathbf{C^{-1}} \vec{z}_{i,y',k} \right) \right) . \tag{5.27}$$

For high values of N_3, using the term in (5.27) is the practical method of choice for guessing the subkey

$$k' = \arg \max_{k \in \{0,1\}^s} \ln(\alpha_{MLP}(x_1, \ldots, x_{N_3}; k)) . \tag{5.28}$$

For the algorithmic description, see Algorithm 5.6.

5.3.7 Generalization to Multi Channels

Reference [4] considers the case where signals from several physical channels are measured simultaneously, e.g., by one power and one EM probe or different EM probes. Stochastic methods can also be generalized to this situation.

The estimation of the deterministic part as well as the selection of contributing instants is then done independently for each physical channel at the profiling phase. For multiple channels the set of instants $\vec{t}$ is

Algorithm 5.6 Maximum Likelihood Principle (multivariate Gaussian distribution) in the presence of masking

Input: (i) The $m \times N_3$ matrix $\mathbf{I}$ containing the N_3 measurement vectors $\vec{i_t} \in \mathbb{R}^m$ for the selected instants $t_1, \ldots, t_m$ of the target device.
(ii) Known data $x_i \in \{0,1\}^d$ for all $i \in \{1, \ldots, N_3\}$ and one fixed unknown subkey $k^\circ \in \{0,1\}^s$ at the target device.
(iii) Least square estimators $\widetilde{h}^*_t(\cdot,\cdot,\cdot)$ with coefficients β^*_{jl} for the u_j basis functions with $0 \leq l < u_j$ at each selected instant $j \in \{t_1, \ldots, t_m\}$.
(iv) The $m \times m$ inverse covariance matrix $\mathbf{C}^{-1} = (c^{-1}_{uv})$
(v) A standard indexing (in descending order) algorithm $I : \underbrace{\mathbb{R} \times \cdots \times \mathbb{R}}_{2^s} \rightarrow P$ with P being the set of all possible permutations of the elements $\{0, \ldots, 2^s - 1\}$.
Output: a permutation of subkey hypotheses

$$\pi = \begin{pmatrix} 0 & 1 & . & . & . & 2^s - 1 \\ \tilde{k}_0 & \tilde{k}_1 & . & . & . & \tilde{k}_{2^s-1} \end{pmatrix}$$

such that $s_{\tilde{k}_0} \geq s_{\tilde{k}_1} \geq \cdots \geq s_{\tilde{k}_{2^s-1}}$ that corresponds to $\mathbb{P}(\tilde{k}_0 = k^\circ) \geq \mathbb{P}(\tilde{k}_1 = k^\circ) \geq \ldots \geq \mathbb{P}(\tilde{k}_{2^s-1} = k^\circ)$.

1: **for** k from 0 to $2^s - 1$ **do**
2: $s_k = 0$;
3: **for** i from 1 to N_3 **do**
4: $\rho = 0$;
5: **for** y' from 0 to $2^v - 1$ **do**
6: **for** j from 1 to m **do**
7: $z_j = i_{ij} - \widetilde{h}^*_j(x_i, y', k)$;
8: **end for**
9: $\rho = \rho + 2^{-v} \exp(-0.5 \cdot \sum_{u=1}^{m} z_u \left(\sum_{v=1}^{m} c^{-1}_{uv} z_v\right))$;
10: **end for**
11: $s_k = s_k + \ln(\rho)$;
12: **end for**
13: **end for**
14: $\{\tilde{k}_0, \ldots, \tilde{k}_{2^s-1}\} \leftarrow I(s_0, \ldots, s_{2^s-1})$;

defined by the joint set of all channels $\vec{t} := (t_1^1, \ldots, t_{m_1}^1, \ldots, t_1^Q, \ldots, t_{m_Q}^Q)$ wherein $\{t_1^q, \ldots t_{m_q}^q\}$ is the set of chosen time instants for channel $q \in \{1, \ldots, Q\}$.

With this definition of $\vec{t} \in \mathbb{R}^{m_1+\cdots+m_Q}$, the profiling stage for the estimation of the noise as well as the key recovery using the minimum principle (5.17) and maximum likelihood principle (5.19) are directly applicable. In case of multi channels the covariance matrix then contains the joint noise distribution of different channels.

Remark 5.3. An experimental evaluation of the use of multi channel based analysis in the stochastic model is provided in [55]. This analysis uses the power channel and one EM channel at observing the AES implementation on the ATM163 microcontroller.

5.4 Experimental Analysis of an AES Implementation

An implementation of the Advanced Encryption Standard (AES) [102] on an 8-bit ATM163 microcontroller was used for the experimental analysis of the efficiency achieved by the minimum principle and maximum likelihood principle. The ATM163 microcontroller is embedded in a smart card and is programmed with a basic variant of the open source smart card operating system SOSSE [34]. The AES is implemented in Assembly language and does not include any side channel countermeasures except that the computation time is constant. The side channel information was gained by measuring the instantaneous current consumption in the ground line while the ATM163 microcontroller computes the AES. Four measurement series were recorded using 2000 single measurements each. Each series uses a different fixed AES key $\vec{k} = \{k_1, ..., k_{16}\}$. The random input data $\vec{x} = \{x_1, ..., x_{16}\}$ were chosen independently from a uniform distribution whereby $x_i \in \{0,1\}^8$ and $k_i \in \{0,1\}^8$ with $i \in \{1, ..., 16\}$.

5.4.1 Profiling Phase

Profiling Phase: Estimation on the deterministic part

For profiling the choice of the selection function was $S(\phi(x_i, k_i))$ for the AES S-Box S with $\phi(x_i, k_i) = x_i \oplus k_i$. This selection function combines an 8-bit subkey k_i and an 8-bit portion of the plaintext x_i as part of the first encryption round as shown in Figure 5.1. Note that this selection function can be also applied at (standard) DSCA.

In this section chosen vector subspaces are based on this selection function, i.e. one 8-bit outcome of the SubBytes transformation in the first AES round. To simplify notation the index of the byte-number i is suppressed from now on. It is plaintext $x := x_1$ and subkey $k := k_1$, i.e., the experimental application is demonstrated for byte number $i = 1$ and has to be repeated for the remaining fifteen subkeys.

Profiling is presented in more detail for the nine-dimensional bitwise coefficient model, referenced as vector subspace $\mathcal{F}_9$. An evaluation of different vector subspaces regarding to their key recovery efficiency is given in Section 5.4.3. $\mathcal{F}_9$ is spanned by the constant function 1 and

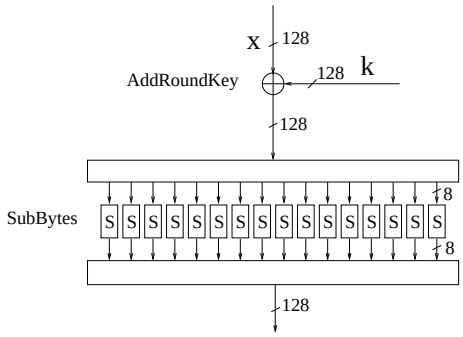


Figure 5.1: Processing of the AES at the beginning of the first round. A suitable intermediate result for DSCA is an 8-bit portion of the outcome of SubBytes as it depends only on an 8-bit subkey and an 8-bit plaintext.

eight functions $g_l : \{0,1\}^8 \to \{0,1\}$ so that $g_l(\cdot)$ is the l-th bit of $S(\cdot)$. The bit ordering is from the most significant bit ($l = 1$) to the least significant bit ($l = 8$).

Example 5.7. For the AES S-box one obtains $S(0) = 63_h = (01100011)_2$, i.e., $g_1(0) = 0$, $g_2(0) = 1$, $g_3(0) = 1$, $g_4(0) = 0$, $g_5(0) = 0$, $g_6(0) = 0$, $g_7(0) = 1$, and $g_8(0) = 1$.

According to equation (5.7) with $u = 9$ and equation (5.13) the deterministic side channel contribution $h_t(\phi(x,k))$ is approximated by

$$\widetilde{h}_j^*(\phi(x,k)) = \beta_{j0}^* + \sum_{l=1}^{8} \beta_{jl}^* \cdot g_l(\phi(x,k)) \; . \tag{5.29}$$

The coefficient β_{j0}^* gives the expectation value of the non-data dependent signal part at instant j and the coefficients β_{jl}^* with $j \neq 0$ are the bitwise data dependent signal portions. Though the internal processing of the implementation is deterministic, the measured quantities are not: Noise is an important contribution to the physical signal. The coefficients β_{jl}^* are revealed by solving an overdetermined system of N linear equations (see equation (5.11) and Algorithm 5.1). Note that the chosen vector subspace is applied to the overall measurement time frame, i.e., combinations of several vector subspaces at different instants are not discussed as part of this section.

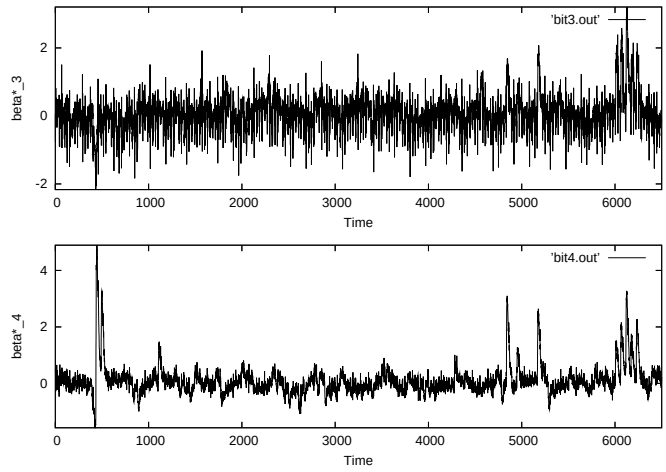


Figure 5.2: Bit-wise coefficients β^*_{j3} and β^*_{j4} as result of profiling at one measurement series with $N = 2000$.

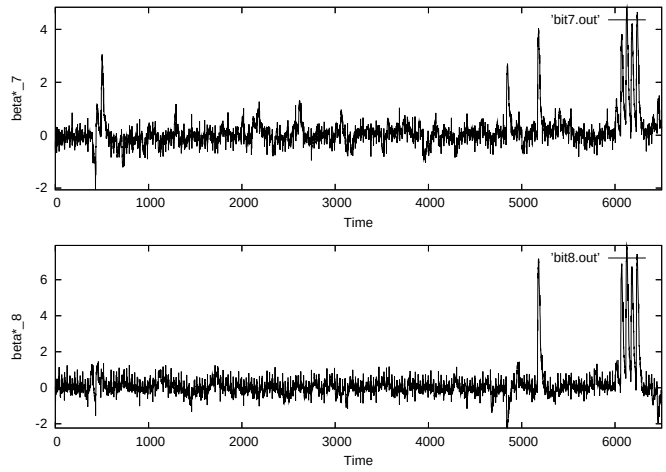


Figure 5.3: Bit-wise coefficients β^*_{j7} and β^*_{j8} as result of profiling at one measurement series with $N = 2000$.

The experimental results, e.g, in Figure 5.2 and Figure 5.3 show that the resulting coefficients β^*_{jl} strongly differ in shape and amplitude. The

signals of bit no. 8 (least significant bit) turned out to be the most significant ones. These estimations show the contributions of single bits at an intermediate result to the overall physical leakage and confirm the results of Chapter 4 that the use of the Hamming weight model cannot be of high quality at an AVR microcontroller.

The coefficients β_{jl}^* were computed on all four measurement series independently. As it can be exemplary seen in Figure 5.4 the deviations of coefficients revealed at the four series are relatively small. As the four series were done with different AES keys, these experimental results confirm the assumptions of Lemma 5.2 saying that it is justified to perform the profiling of $h_t^*(\cdot, k)\colon \{0,1\}^d \to \mathbb{R}$ for only one subkey $k \in \{0,1\}^s$.

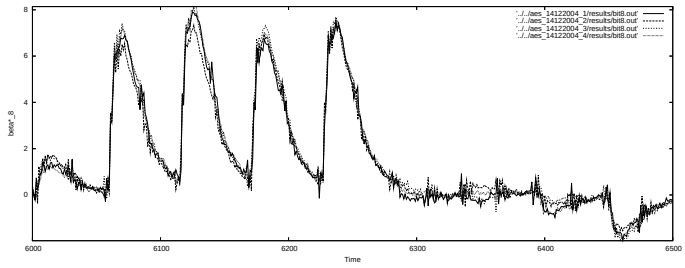


Figure 5.4: Coefficient β_{j8}^* for all four measurement series as a function of time. It is $N = 2000$.

Profiling without Knowing the Key

In case that the subkey k° is unknown the estimation of h_t^* may be performed for all possible key values $k \in \{0,1\}^8$ as indicated in Section 5.3.4. It was experimentally confirmed that equation (5.15) indeed was minimal for the correct subkey k°. By analyzing the relevant time frame of 6500 instants the difference between the first and the second candidate was 1.9 times larger than the difference between the second and the last candidate (see Table 5.2).

Figure 5.5 shows the squared Euclidean norm $\|b\|^2$ of coefficients $\vec{b} = (\beta_{j1}^*, \ldots, \beta_{j,u_j-1}^*)$ in $\mathcal{F}_9$ for three different subkeys. It is obvious that the correct subkey k° can be easily identified, e.g., with the help of Algorithm 3.3. This approach is an alternative to DSCA and makes the

Table 5.2: Ranking list for determining the subkey at profiling.

Subkey	Result of Equation (5.15)
FF	7.54343e+08
62	7.60230e+08
F3	7.60278e+08
05	7.60328e+08
C4	7.60443e+08
.	.
.	.
.	.
60	7.63139e+08
25	7.63141e+08
53	7.63208e+08
73	7.63221e+08
45	7.63305e+08

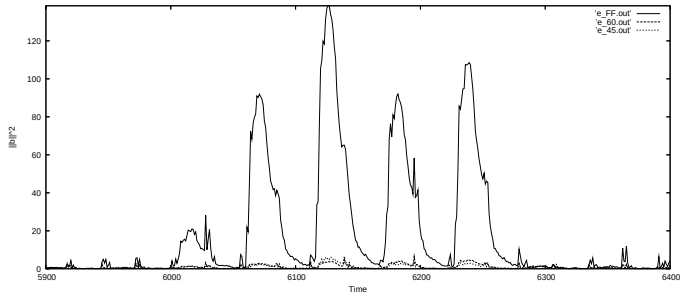


Figure 5.5: Squared Euclidean norm $\|b\|^2$ of coefficients $\vec{b} = (\beta^*_{j1}, \ldots, \beta^*_{j,u_j-1})$ in $\mathcal{F}_9$ for three different subkeys as result of profiling. The correct subkey value yields outstanding signals. It is $N = 2000$.

toolbox of the stochastic model applicable even if the correct key value is not known beforehand at profiling. However, it is noted that DSCA requires less computational efforts to determine an unknown subkey k°.

5.4.2 Selection of Instants

As part of this chapter several algorithms for instant selection have been applied and tested for efficiency at key recovery. Algorithm 5.7 and Algorithm 5.8 are used as the starting point for instant selection. Algo-

rithm 5.7 solely considers the squared Euclidean norm of the data dependent least square coefficients, whereas Algorithm 5.8 considers both the squared Euclidean norm and the empirical variance. The motivation for Algorithm 5.7 lies in the assumption that data dependent least square coefficients vanish if the physical signal does solely depend on noise. Algorithm 5.8 compares the squared Euclidean norm of the data dependent least square coefficients with the empirical variance of the noise. Here, one decides in favour of instants where the empirical variance of the noise is low if compared to the squared Euclidean norm of the data dependent least square coefficients. Though originally proposed as part of the work described in Section 6.6 another reasonable algorithm is also re-printed here as it is now seen as the most efficient one. Algorithm 5.9 considers differences in the deterministic leakage for all possible values of the profiled intermediate result by using the least square estimators $\widetilde{h}_j^*(\cdot)$. For instant selection, points in time are preferred at which the sum of squared pairwise t-differences for the estimated deterministic leakage assumes significantly high values. Algorithm 5.7, Algorithm 5.8, and Algorithm 5.9 require a threshold $\tau_c \in \mathbb{R}$ or a factor $c \in \mathbb{R}$ as input. These constants are adjusted after visible inspection of the significance vectors, e.g., Figure 5.6 representing $\tau \in \mathbb{R}^p$ in Algorithm 5.7 indicates that $\tau_c = 30$ is suitable.

Algorithm 5.7 Instant selection based on the squared Euclidean norm.

Input: (i) Least square estimators $\widetilde{h}_j^*(\cdot)$ with coefficients β_{jl}^* for the u_j basis functions with $0 \leq l < u_j$ at all sampled instants $j \in \{1, \ldots, p\}$. (ii) A significance threshold $\tau_c \in \mathbb{R}$.

Output: A set of m ($m \leq p$) chosen points of interest $\{t_1, \ldots, t_m\}$

1: $P \leftarrow \emptyset$;
2: **for** j from 1 to p **do**
3: $\quad \vec{b} = (\beta_{j1}^*, \ldots, \beta_{j,u_j-1}^*)$
4: $\quad \tau_j = \sum_{l=1}^{u_j-1} (\beta_{jl}^*)^2$; {Compute the squared Euclidean norm.}
5: $\quad$ **if** $\tau_j \geq \tau_c$ **then**
6: $\quad\quad P \leftarrow P \cup j$;
7: $\quad$ **end if**
8: **end for**

Algorithm 5.8 Instant selection based on the squared Euclidean norm and empirical variance.

Input: (i) The $p \times N$ matrix $\mathbf{I}$ containing the N vectors $(\vec{i_1}, \cdots \vec{i_N})$ with $\vec{i_i} \in \mathbb{R}^p$ for all $i \in \{1, \ldots, N\}$ and the p vectors $(\vec{i_1}, \cdots \vec{i_p})$ with $\vec{i_j}^{col} \in \mathbb{R}^N$ for all $j \in \{1, \ldots, p\}$.
(ii) Least square estimators $\widetilde{h}_j^*(\cdot)$ with coefficients β_{jl}^* for the u_j basis functions with $0 \leq l < u_j$ at each sampled instant j.
(iii) A factor $c \in \mathbb{R}$.

Output: A set of m ($m \leq p$) chosen points of interest $\{t_1, \ldots, t_m\}$

1: $P \leftarrow \emptyset$;
2: **for** j from 1 to p **do**
3: $\quad \vec{b} = (\beta_{j1}^*, \ldots, \beta_{j,u_j-1}^*)$;
4: $\quad \tau_j = \sum_{l=1}^{u_j-1} (\beta_{jl}^*)^2$; {Compute the squared Euclidean norm.}
5: $\quad \overline{i_j} = \sum_{i=1}^{N} i_{ij}$;
6: $\quad S_j^2 = \frac{1}{N-1} \sum_{i=1}^{N} (i_{ij} - \overline{i_j})^2$; {Sample Variance, equation (2.6).}
7: $\quad$ **if** $\tau_j \geq c \cdot S_j^2$ **then**
8: $\quad\quad P \leftarrow P \cup j$;
9: $\quad$ **end if**
10: **end for**

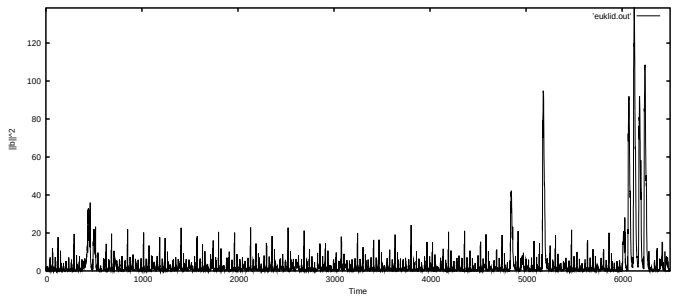


Figure 5.6: Squared Euclidean norm $\|\vec{b}\|^2$ of coefficients $\vec{b} = (\beta_{j1}^*, \ldots, \beta_{j,u_j-1}^*)$ as result of profiling at one measurement series with $N = 2000$.

Algorithm 5.9 Instant selection based on the sum of squared pairwise t-differences.

Input: (i) The $p \times N$ matrix $\mathbf{I}$ containing the N vectors $(\vec{i_1}, \cdots \vec{i_N})$ with $\vec{i_i} \in \mathbb{R}^p$ for all $i \in \{1, \ldots, N\}$ and the p vectors $(\vec{i_1}, \cdots \vec{i_p})$ with $\vec{i_j^{col}} \in \mathbb{R}^N$ for all $j \in \{1, \ldots, p\}$.
(ii) Least square estimators $\widetilde{h}_j^*(\cdot)$ for one d-bit intermediate result with coefficients β_{jl}^* for the u_j basis functions with $0 \leq l < u_j$ at all sampled instants $j \in \{1, \ldots, p\}$.
(iii) A significance threshold $\tau_c \in \mathbb{R}$.

Output: A set of $m (m \leq p)$ chosen points of interest $\{t_1, \ldots, t_m\}$

$P \leftarrow \emptyset$;
for j from 1 to p **do**
  $\tau_j = 0$;
  for i from 0 to $2^d - 1$ **do**
    $\mu_{ij} = h_j^*(i)$; {Estimate the deterministic leakage for all possible values of the intermediate result.}
  end for
  $S_j^2 = \frac{1}{N-1} \sum_{i=1}^{N} (i_{ij} - \overline{i_j})^2$; {Sample Variance, equation (2.6).}
  for i from 0 to $2^d - 1$ **do**
    for l from $i + 1$ to $2^d - 1$ **do**
      $\tau_j = \tau_j + \frac{(\mu_{ij} - \mu_{lj})^2}{S_j^2}$; {Sum up the squared differences weighted by the empirical variance.}
    end for
  end for
  if $\tau_j \geq \tau_c$ **then**
    $P \leftarrow P \cup j$;
  end if
end for

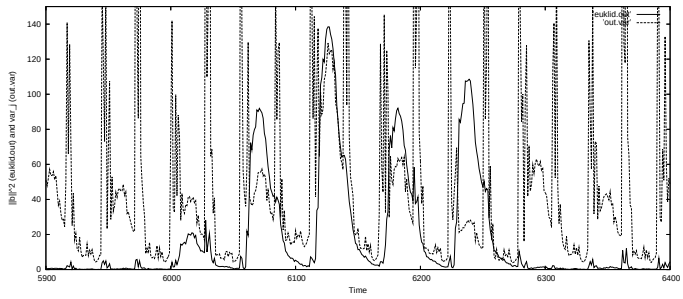


Figure 5.7: Squared Euclidean norm $\|\vec{b}\|^2$ of coefficients $\vec{b} = (\beta^*_{j1}, \ldots, \beta^*_{j,u_j-1})$ and empirical variance at one measurement series with $N = 2000$. High empirical variance can be seen at the clock edges.

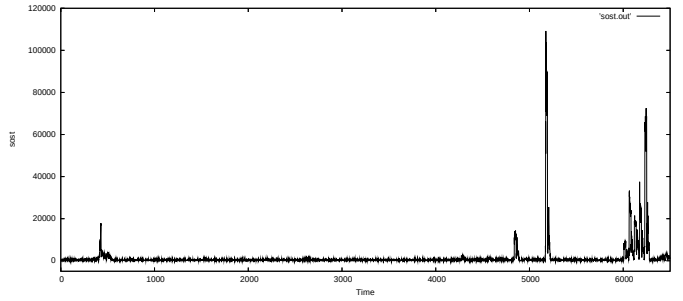


Figure 5.8: Squared sum of pairwise $t-$differences in $\mathcal{F}_9$ at one measurement series with $N = 2000$. High values indicate significant differences for the deterministic part considering all possible values of the intermediate result. Note that remaining noise is much better suppressed than in Figure 5.6.

Concretely, the following sets of instants have been chosen.

S_1: By selecting all instants with $\|b\|^2 \geq \tau_c$[1] Algorithm 5.7 finds seven different signals[2] and the number of instants was $m = 147$. For each signal, most instants are in series.

S_2: This selection is a refinement (reduction) of S_1. For each signal with $\|b\|^2 \geq \tau_c$ of S_1 only one instant was selected that yields the maximum value of $\|b\|^2$. Here, seven different instants were obtained.

S_3: This selection is a refinement (reduction) of S_1. Only the instant of S_1 yielding the maximum value of $\|b\|^2$ is selected.

S_4: This is an application of Algorithm 5.8 with $c = 1$. Here, $m = 100$ different instants were selected, but only at five different signals.

S_5: This selection is a refinement of S_4 and S_1. S_4 is extended by all instants that fulfill $\|b\|^2 > \tau$ at the remaining two signals that are not found by S_4. Altogether, it is $m = 120$.

S_6: This is a manual refinement (reduction) of S_1. For each of the seven signals with $\|b\|^2 \geq \tau$ three instants were chosen by visual inspection so that the instants chosen are spread over one signal. For the selection S_6 it is $m = 21$.

S_7: This selection applies Algorithm 5.9 with $c = 2000$. Altogether, 455 instants are found that occur at eleven signals. S_7 outputs the maximum number of signals among all selections.

Profiling Phase: Estimation of the Noise

The characterization of the noise was done independently of the estimation of the deterministic part. Concretely, as preparation step for the maximum likelihood principle the recovery of the least square estimates was repeated with $N_1 = 1000$. The choice of instants was done with S_2 and S_6, i.e., low-dimensional sets of instants. The computations of the covariance matrix $\mathbf{C} = (c_{uv})_{1 \leq u,v \leq m}$ for sets of m points were done with

[1] For $\mathcal{F}_9$ the choice was $\tau_c = 30$.

[2] All instants that occur during one instruction cycle are assigned to one signal.

$N_2 = 1000$ and $N_2 = 5000$. For the case $N_2 = 5000$ three measurement series were combined, except for the one that is used for the key recovery later on.

5.4.3 Key Recovery Phase

Key Recovery Phase: Minimum Principle

For the minimum principle given by equation (5.17) and Algorithm 5.3 the estimation of h_t^* is needed, but not the estimation of the noise contribution. One measurement series served for the profiling step ($N = 2000$) and the key recovery is applied at another series. The minimum value of equation (5.17) is computed for all subkeys $k' \in \{0,1\}^8$. For the analysis, the minimum principle was applied to all selections of instants introduced in Section 5.4.2.

In this contribution efficiency is assessed by the average number of single measurements needed to achieve a certain success rate using a given number N_3 of single measurements taken from the same measurement set. The success rate (SR) was tested by ten thousand random choices of N_3 single measurements from one series. It can be seen in Table 5.3 that 10 single measurements yield already a success rate of about 75 % and beyond 30 single measurements the success rate can be above 99.9 %. The best results were gained at the selections S_5 and S_6, i.e., selections which include post-processings. However, it is worth mentioning that Algorithm 5.7 achieves an improved performance if compared to Algorithm 5.8. This can be seen as an indication that the number of contributing signals is the key parameter. Further, the loss of efficiency between S_1 and S_2 is small; by considering only 7 instants instead of 147 one achieves nearly the same key recovery efficiency. Table 5.3 also impressively shows the superiority of multivariate analysis if compared to the univariate analysis by using S_3.

Choice of Vector Subspaces

Different vector spaces have been evaluated regarding their efficiency. The choice of high-dimensional vector spaces, e.g, by including all terms of $g_i(\phi(x,k))g_{i'}(\phi(x,k))$ $(i \neq i')$ (see (5.7) and (5.29)) did not lead to great improvements. Only weak contributions of second-order coeffi-

Table 5.3: Success Rate (SR) that the correct subkey value is the best candidate as result of (5.17) and Algorithm 5.3 by using N_3 randomly chosen measurements for the analysis at the set of instants S_1 to S_6. The vector space used was $\mathcal{F}_9$.

N_3	SR for S_1	SR for S_2	SR for S_3	SR for S_4	SR for S_5	SR for S_6
2	5.57 %	5.64 %	1.06 %	3.31 %	6.35 %	6.36 %
3	12.06 %	11.14 %	1.65 %	7.49 %	13.21 %	13.57 %
5	29.14 %	28.47 %	3.00 %	21.43 %	32.81 %	33.40 %
7	50.39 %	48.20 %	4.39 %	39.41 %	54.23 %	53.88 %
10	75.29 %	73.45 %	8.29 %	65.45 %	78.97 %	78.69 %
15	94.27 %	92.92 %	14.68 %	89.22 %	95.77 %	95.15 %
20	98.57 %	98.31 %	22.26 %	97.59 %	99.17 %	98.82 %
30	99.92 %	99.89 %	39.34 %	99.85 %	99.97 %	99.95 %

cients were observed that even vanish at many combinations. Results are presented for

1. $\mathcal{F}_2$: the 8-bit Hamming weight model ($u = 2$),
2. $\mathcal{F}_5$: a set of four bit-wise coefficients ($u = 5$) (these are the most significant bit-wise coefficients of $\mathcal{F}_9$),
3. $\mathcal{F}_{10}$: a set of the bit-wise coefficient model and one carefully chosen second-order coefficient ($u = 10$), and
4. $\mathcal{F}_{16}$: the bit-wise coefficient model extended by seven consecutive second order coefficients ($u = 16$).

Key recovery efficiency for the minimum principle is summarized in Table 5.4. The time instants were chosen in the same way as described for $\mathcal{F}_9$ with S_1 at the beginning of Section 5.4.3 and the thresholds τ_c are indicated. Figure 5.9 illustrates the different thresholds τ_c in the vector subspaces $\mathcal{F}_2$ and $\mathcal{F}_{16}$.

High-dimensional vector spaces require more measurement curves than low-dimensional ones. It was experimentally confirmed that the threshold τ_c can be lowered with increasing N for a fixed vector subspace. There is a trade-off between the number of measurements used

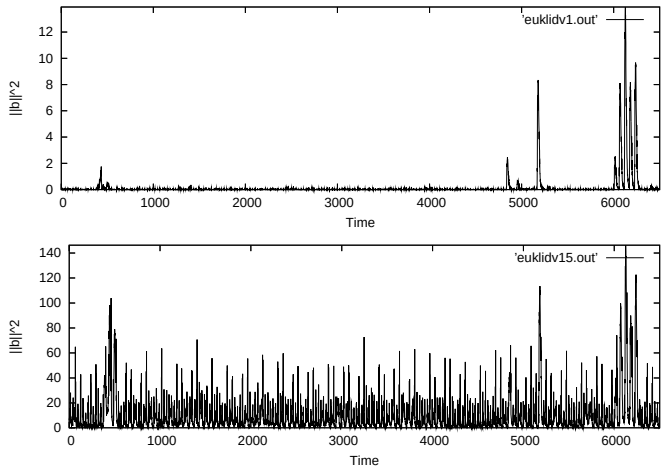


Figure 5.9: Squared Euclidean norm $\|\vec{b}\|^2$ of coefficients $\vec{b} = (\beta^*_{j1}, \ldots, \beta^*_{j,u_j-1})$ by using $\mathcal{F}_2$ (upper plot) and $\mathcal{F}_{16}$ (lower plot) at one measurement series with $N = 2000$.

Table 5.4: Success Rate (SR) that the correct key value is the best candidate as result of (5.17) by using N_3 randomly chosen measurements in different vector subspaces. Algorithm 5.7 was used for instant selection with a vector space depending threshold τ_c.

N_3	SR for $\mathcal{F}_2$ ($\tau_c = 1$)	SR for $\mathcal{F}_5$ ($\tau_c = 8$)	SR for $\mathcal{F}_9$ ($\tau_c = 30$)	SR for $\mathcal{F}_{10}$ ($\tau_c = 30$)	SR for $\mathcal{F}_{16}$ ($\tau_c = 70$)
2	2.59 %	4.22 %	5.57 %	5.18 %	4.81 %
3	4.75 %	9.03 %	12.06 %	11.27 %	9.73 %
5	11.63 %	21.97 %	29.14 %	27.28 %	23.69 %
7	21.66 %	37.61 %	50.39 %	47.66 %	41.04 %
10	37.77 %	62.22 %	75.29 %	72.94 %	65.05 %
15	62.46 %	86.36 %	94.27 %	93.57 %	88.69 %
20	80.36 %	95.71 %	98.27 %	98.41 %	96.17 %
30	96.23 %	99.74 %	99.92 %	99.88 %	99.81 %

during profiling and the dimension of a suitable vector space. In this case study, $\mathcal{F}_9$ (see Table 5.3 and 5.4) seems to be a good choice though there is some space left for optimization, e.g., by using $N = 5000$, $N_3 = 10$,

and $\tau_c = 10$ the success rate of $\mathcal{F}_{10}$ was 80.19 % and superseded the corresponding result for $\mathcal{F}_9$ (77.31 %). Another optimization is to select only contributing functions $g_{i,t}(\cdot,\cdot)$ for the chosen vector subspace at the relevant instants.

Comparison with the DSCA Correlation Method

Herein, the efficiency gain of the minimum principle is compared with the correlation method of [6] on the same pool of measurement data. The correlation method checks for the maximum correlation peak obtained and it does not evaluate joined sets of multiple instants.

The success rate obtained with the correlation method is illustrated in Table 5.5 and can be compared with selection S_3 in Table 5.3 which was restricted to the same instant. In comparison, the correlation method yields worse success rates than the minimum principle. By taking, e.g., $N_3 = 10$ the minimum principle yields an improvement by a factor of 3.0 regarding the Hamming weight prediction and by a factor of 7.1 regarding the best result of one bit prediction of the correlation method. Even, if the estimated coefficients b_{it} of the minimum principle are known an improvement by a factor of 1.8 in favour of the minimum principle is achieved. Note that the relative factor depends on N_3. As the minimum principle uses the adaptation of probability densities it is advantageous if compared to the correlation method that exploits the linear relationship. Moreover, it is worth pointing out that the success rate of the minimum principle increases greatly, if multiple signals are jointly evaluated.

Key Recovery Phase: Maximum Likelihood Principle

For the maximum likelihood principle as described in Section 5.3.4 and equation (5.19) both the estimation of h_j^* and the estimation of the noise is needed. Profiling was done as described in the corresponding parts of Section 5.4.1 and 5.4.2.

The m-dimensional random vector $\vec{Z} = (I_{t_1}(X,k) - \widetilde{h}^*_{t_1}(X,k), \ldots, I_{t_m}(X,k) - \widetilde{h}^*_{t_m}(X,k))$ is assumed to be jointly normally distributed with covariance matrix C. The strategy is to decide for the key hypothesis k' that maximizes equation (5.19) for the multivariate Gaussian distribution using N_3 measurements which is equivalent to find the minimum of the expression $\sum_{i=1}^{N_3} \vec{z_i}^T C^{-1} \vec{z_i}$.

Table 5.5: Success Rate (SR) obtained for the DSCA correlation method using the 8-bit Hamming weight and the least significant bit (lsb-Bit) as the selection function. The last column shows the SR if the weighted estimated coefficients β^*_{jl} using $\mathcal{F}_9$ are used for the correlation.

N_3	SR (Hamming weight)	SR (lsb-Bit)	SR (estimated b_{it})
5	0.82 %	0.51 %	1.12 %
7	1.31 %	0.84 %	2.37 %
10	2.74 %	1.17 %	4.60 %
15	6.04 %	2.11 %	9.33 %
20	9.70 %	3.55 %	16.67 %
30	19.67 %	6.54 %	31.99 %
50	41.27 %	16.53 %	62.84 %
100	82.85 %	45.22 %	96.13 %

The analysis was done by using the vector subspace $\mathcal{F}_9$ with the selections S_2 and S_6 defined at the beginning of Section 5.4.2. Note that for the single instant selection S_3 the maximum likelihood principle reduces to the minimum principle.

Again, the success rate (SR) was computed using ten thousand random choices from one measurement series. As shown in Table 5.6, based on $N_2 = 1000$ a significant improvement was achieved for the selection S_2 regarding Table 5.3, but not for the selection S_6. This decrease by using the maximum likelihood principle if $N_3 < 15$ and $N_2 = 1000$ for S_6 can be explained by the limited profiling process: the estimation error at the profiling of a 7×7 covariance matrix is significantly lower than the error committed for a 21×21 matrix on the base of $N_2 = 1000$. This assessment is confirmed by the corresponding columns in Table 5.6 for $N_2 = 5000$. Both the success rates for S_2 and S_6 were further enhanced. As result, a high value for N_2 can be crucial for the maximum likelihood principle, especially if high dimensions are used for the covariance matrix.

The maximum likelihood method needs typically twice the number of measurements during profiling. Therefore, even though key recovery is less efficient under certain circumstances the 'minimum principle' might be preferred. Given 15 measurements, it can be read out from Table 5.6 that the maximum probability to find the correct key value is

Table 5.6: Success Rate (SR) that the correct key value is the best candidate as result of equation (5.19) by using N_3 randomly chosen single measurements for the analysis. All results are based on F_9 with $N_1 = 1000$. If not explicitly stated it is $N_2 = 1000$.

N_3	SR for S_2	SR for S_6	SR for S_2 (N_2=5000)	SR for S_6 (N_2=5000)
2	6.06 %	4.73 %	7.39 %	6.55 %
3	13.93 %	10.45 %	17.06 %	16.00 %
5	36.30 %	28.04 %	43.70 %	41.43 %
7	61.12 %	51.48 %	70.51 %	68.34 %
10	84.33 %	78.26 %	91.08 %	90.17 %
15	97.97 %	95.86 %	99.14 %	99.25 %
20	99.85 %	99.49 %	99.97 %	99.96 %
30	99.99 %	>99.99 %	>99.99%	>99.99 %

99.25 %. The resulting probability to decide for the correct AES key is $(0.9925)^{16} = 0.8865$.

The number N_3 of measurements can be further reduced if it is tolerated that the correct key value is 'only' among the first candidates as result of DSCA and a plaintext-ciphertext pair is available. For example, if the correct key value is among the first four subkey candidates with high probability, up to 2^{32} tries remain to localize the correct key value. In case of S_2 and $N_3 = 10$ the corresponding success rate that the correct subkey is at least at the fourth position of the subkey ranking was 97.58 % if $N_2 = 1000$, and 99.42 % if $N_2 = 5000$.

5.5 Experimental Analysis of a Masked Implementation

Stochastic methods were also applied at a masked implementation. This section focuses on an application of the most general case for higher order analysis in the presence of masking, i.e., an implementation of boolean masking is considered that is typically the first step of, e.g., a masked AES or DES implementation. At a masked cryptographic implementation it is further necessary to switch the mask at non-linear or

arithmetic operations which is not considered as part of this experimental analysis. Then the situation is even more favorable because of additional leakage signals, e.g., caused by S-Box processing or the use of a restricted form of masking.

The masked implementation was done on an 8-bit microprocessor AT90S8515 that is embedded in a smart card and programmed with a basic variant of SOSSE [34]. It proceeds as shown in Fig. 5.10. Note that there are different implementation choices of masking. Alternatively, it can be assumed that $x \oplus y$ is computed first before adding the key k, as, e.g., done in [93]. The motivation for the choice of this implementation in Fig. 5.10 is based on the fact that neither $x \oplus k$ nor $x \oplus y$ should be observable at a single point in time. Leakage on y, k, x, $y \oplus k$ and $y \oplus k \oplus x$, however, remains observable at single instants.

The physical channel used is the power consumption of the 8-bit microprocessor AT90S8515. By using the estimation of the deterministic part it was assured that first order differential analysis is prevented by verifying that leakage of the intermediate results $k \oplus x$ and $y \oplus x$ at single instants is negligible if any.

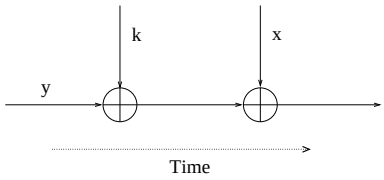


Figure 5.10: Process of boolean masking

5.5.1 Profiling Phase

Concretely, four measurement series were recorded with $N = 10,000$ measurements each using different fixed keys. Further, one additional measurement series includes $N = 20,000$ measurements with varying keys drawn randomly from a uniform distribution. All these series were used for profiling purposes.

For profiling a 'white-box' model is assumed, i.e., x, k, and y are known. Profiling applies the stochastic model at the two intermediate results $y \in \{0,1\}^8$ and $(y \oplus k \oplus x) \in \{0,1\}^8$. More concretely, the vector

subspace is spanned by the constant function 1, the bits of y, and the bits of $x \oplus y \oplus k$ yielding an 17 dimensional vector subspace. In a first try, the deterministic part $h_j^*(x, y, k)$ was approximated by

$$\widetilde{h}_j^*(x, y, k) = \beta_{j0}^* \cdot 1 + \sum_{l=1}^{8} \beta_{jl}^* \cdot g_l(y) + \sum_{l=9}^{16} \beta_{jl}^* \cdot g_{l-8}(x \oplus y \oplus k) \,. \quad (5.30)$$

Herein, the function $g_l : \{0,1\}^8 \to \{0,1\}$ outputs the l-th bit of an 8-bit data item with a bit ordering from the most significant bit ($l = 1$) to the least significant bit ($l = 8$). The coefficients β_{jl}^* are determined by solving (5.11). As result, it turned out that leakage signals of y and $x \oplus y \oplus k$ are well separated in time. Therefore, it is appropriate to reduce the number of dimensions during profiling which helps in suppressing noise in the estimation process. For the refined application of (5.11), the chosen vector subspace depends on the time instant j, i.e., y is profiled if $0 < j \leq 2500$ and $x \oplus y \oplus k$ is profiled if $2500 < j \leq 10000$:

$$\widetilde{h}_j^*(x, y, k) = \begin{cases} \beta_{j0}^* + \sum_{l=1}^{8} \beta_{jl}^* \cdot g_l(y) & \text{if } 0 < j \leq 2500 \\ \beta_{j0}^* + \sum_{l=9}^{16} \beta_{jl}^* \cdot g_{l-8}(x \oplus y \oplus k) & \text{if } 2500 < j \leq 10000 \end{cases} \quad (5.31)$$

Accordingly, the coefficients β_{jl}^* are set to zero if not profiled in the given time frame:

$$\beta_{jl}^* := \begin{cases} 0 & \text{if } (9 \leq l \leq 16) \text{ and } (0 < j \leq 2500) \\ 0 & \text{if } (1 \leq l \leq 8) \text{ and } (2500 < j \leq 10000) \end{cases}$$

Profiling by using the vector subspace given in (5.31) was also applied to four measurement series with different fixed keys. By comparing experimental profiling results it turned out that the EISM property is only partly fulfilled (see Figure 5.12) as the estimated coefficients β_{jl}^* differ significantly for different measurement series at a few instants. There are even leakage contributions that are completely suppressed in the series with varying keys, thus indicating that there is key dependent leakage which averages to zero if considering profiling based on varying keys. For the series with fixed keys two additional signals were found and included in the set of selected instants.

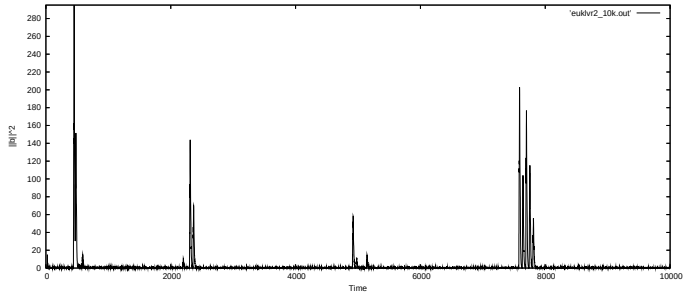


Figure 5.11: Squared Euclidean norm $\|\vec{b}\|^2 = \|(\beta^*_{j1}, \beta^*_{j2}, ..., \beta^*_{j16})\|^2$ of the bit depending coefficients as result of profiling according to equation (5.31) by using the measurement series with varying key data. Note that this computation includes two sets of basis functions in subsequent, but separated time frames. High values for $\|b\|^2$ indicate instants with significant deterministic side channel leakage.

5.5.2 Key Recovery Phase

At key recovery, the adversary knows x and aims at retrieving the fixed key k°. In a special test case an artificial case is considered that masking is completely ineffective at key recovery, i.e., here x and y are known. Key recovery was done at the series with fixed keys provided that the series assigned for profiling was different. When applying the maximum likelihood principle at key recovery, it was $N_1 = N_2 = N/2$ during profiling. For the minimum principle, all N measurements were used for profiling. As far as key recovery according to the maximum likelihood principle is concerned equations (5.27) and (5.28) were used.

Maximum Likelihood Principle

Results for 'varying-key' profiling are based on a ten-dimensional covariance matrix and are summarized in Table 5.7. Note that in case of misses of the correct key value often a closely related key value differing only at one bit is obtained instead, especially at high values of N_3. Such an observation is reasonable, as differential side channel analysis on a boolean operation yields to related key hypotheses (see Chapter 4 and [81]). Results for 'fixed key' profiling can be found in Table 5.8 by using a twelve-dimensional covariance matrix. Table 5.8 applies three

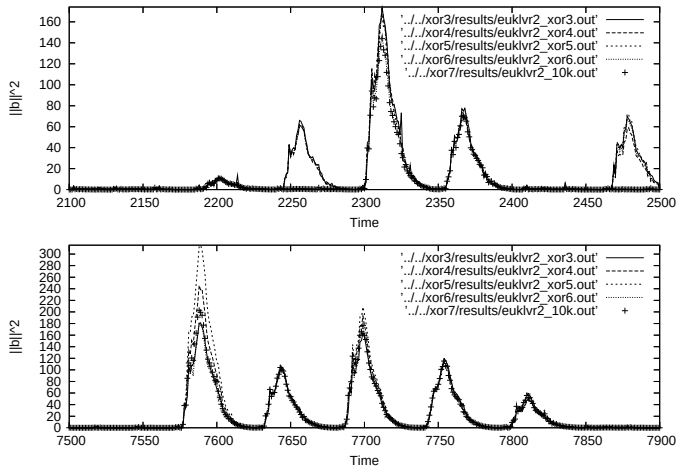


Figure 5.12: Squared Euclidean norm $\|\vec{b}\|^2 = \|(\beta^*_{j1}, \beta^*_{j2}, ..., \beta^*_{j16})\|^2$ of the bit depending coefficients after profiling for different series. In these time frames, e.g., in the upper plot around offset 2260 and 2480, profiling results with fixed keys give a signal part that is wiped out if profiling is done with varying keys. The lower plot shows resulting differences in the estimation for fixed keys. For each series, 10,000 single measurements were used.

different probability densities (obtained from series no. 2, 3, and 4) to series no. 1 yielding results of various quality. This is a clear indicator for a lack of symmetry at some instants. If comparing Table 5.7 with Table 5.8 the average success rate for key recovery is 69 % for 'varying key' profiling while it is 43 % for 'fixed key' profiling at $N_3 = 100$. Trial classifications on the profiling series themselves, however, yield success rates of 97 % at $N_3 = 100$. These results lead to two conclusions. First, it is indicated that profiling for all subkeys will clearly increase success rates and second, the use of a measurement series with varying keys is advantageous if profiling for all subkeys is not feasible, e.g., because of limitations at the profiling stage.

Maximum Likelihood Principle with Known Masking Values

Here, we consider an artificial case that key recovery can be done with known masking values, i.e., masking is completely ineffective. This might

Table 5.7: Success Rate (SR) that the correct key value is the best candidate as result of (5.27) by using N_3 randomly chosen measurements (100 random selections of measurements each). Profiling was done with variable keys with $N_1 = N_2 = 10000$.

N_3	SR Series no. 1	SR Series no. 2	SR Series no. 3	SR Series no. 4
10	8 %	9 %	5 %	6 %
20	20 %	24 %	14 %	13 %
30	26 %	33 %	22 %	24 %
50	53 %	53 %	36 %	34 %
100	82 %	78 %	55 %	62 %
200	92 %	95 %	79 %	82 %
400	98 %	99 %	93 %	96 %

Table 5.8: Profiling with fixed key for series no 2, no. 3, and no. 4 with $N_1 = N_2 = 5000$ and key recovery on series no 1. Success Rate (SR) that the correct key value is the best candidate as result of (5.27) by using N_3 randomly chosen measurements (100 random selections of measurements each).

N_3	SR Series no. 2	SR Series no. 3	SR Series no. 4
10	0 %	4 %	4 %
20	5 %	11 %	12 %
30	11 %	25 %	17 %
50	6 %	21 %	32 %
100	24 %	53 %	53 %
200	42 %	78 %	76 %
400	50 %	96 %	86 %

be a realistic case if the random number generator used for generating masking values is predictable, e.g., as result of physical modification or of special insights in the construction. The procedure for key recovery was modified in such a way that y is known and is equivalent to a first order side channel analysis. Results are presented in Table 5.9. For example, for $N_3 = 10$ the success rate to obtain the correct key value is 62.0 %. Among the key misses, a total amount of 25.5 % aggregates at eight related key values differing only by one bit from the correct key value. The security gain of masking in terms of N_3 can be quantified

if comparing to Table 5.7, series no. 1. If considering success rates of about 90 %, N_3 is enlarged by roughly a factor of ten.

Table 5.9: Summarizing the results of key recovery with knowledge of masking values. Success Rate (SR) that the correct key value is the best candidate (1000 repetitions) on series no. 1.

N_3	SR (Known masking values)
2	8.8 %
3	17.2 %
5	31.3 %
7	46.0 %
10	62.0 %
20	89.1 %
30	97.3 %
50	99.5 %

Table 5.10: Summarizing the results of the application of the minimum principle. Profiling was done on the series with varying keys and it was $N = 20,000$. Success Rate (SR) that the correct key value is the best candidate (100 repetitions) on series no. 1.

N_3	SR (Minimum Principle)
10	5 %
20	11 %
30	12 %
50	14 %
100	42 %
200	65 %
400	89 %

Minimum-Principle

For the application of the minimum principle, the series with varying keys was used for profiling and the choice of time instants was identical

to the application of the maximum likelihood principle. The results give evidence that the minimum principle works in practice. Success rates of 42 % if $N_3 = 100$ and of 89 % if $N_3 = 400$ were obtained by using series no. 1 (see Table 5.10). These results can be compared with the column for series no. 1 in Table 5.7 and reveal a noticeable efficiency loss if compared to the maximum likelihood principle.

Second-Order DSCA

Known-offset single-bit second-order DSCA has been applied according to [93, 143]. It is noted that the use of a selection function based on the 8-bit Hamming weight of $x \oplus k$ [67, 112, 109, 131] turned out to be not appropriate for the present type of leakage at the microcontroller AT90S8515, as it was not feasible to find any combination of instants for that second-order DSCA converges to the correct key value. This observation is due to the fact that the leakage contribution per bit is not a constant portion and in addition a different sign for single bit contributions is observed.

The measurement outcomes were initially standardized with $i_{ij} := (i_{ij} - \overline{i_j})/\sqrt{S_j}$ wherein $\overline{i_j} = \sum_{i=1}^{N} i_{ij}$ is the mean value and $S_j^2 = \frac{1}{N-1}\sum_{i=1}^{N}(i_{ij} - \overline{i_j})^2$ is the empirical variance for each j-th time instant. Second-order DSCA results are clearly significant for the correct key value. However, for the interpretation of results in this setting, one needs to know the sign of contribution for each bit at the two instants[3]. This typically requires also some kind of profiling phase. To get a rough rating for second-order DSCA success, second-order DSCA was applied by using both the multiplication and the absolute difference of the measurands as combining operation (cf. Algorithm 3.6). Results are given in Table 5.11.

For comparison, the stochastic maximum likelihood principle was repeated by restricting to the same two instants and using a three-dimensional vector subspace spanned by the constant function 1 and the bit of y and $x \oplus y \oplus k$ in question. At key recovery, this leads to success rates of 86 % at $N_3 = 200$, 89 % at $N_3 = 400$ and 98 % at

[3] This is different to a non-linear setting, e.g., after a masked S-box in the first round of a block cipher. Then one may be only interested in the highest absolute peak for key recovery.

Table 5.11: Summarizing the results for second order DSCA. Success Rate (SR) that the correct key value is the best candidate (10000 repetitions) on series no. 1. Column 2 shows the results for the multiplication and Column 3 gives the results for the absolute difference as combining operation according to Algorithm 3.6).

N_3	SR (Multiplication)	SR (Difference)
10	56.12 %	52.89 %
20	58.76 %	53.69 %
50	64.22 %	58.08 %
100	70.44 %	61.48 %
200	76.06 %	66.34 %
400	85.48 %	71.77 %
800	93.32 %	79.20 %
2000	99.48 %	91.45 %

$N_3 = 800$ indicating also a significant gain in key recovery efficiency if compared to second-order DSCA.

Chapter 6

Templates vs. Stochastic Methods

6.1 Contribution

This chapter deals with an experimental performance analysis for advanced multivariate methods proposed for side channel cryptanalysis. The methods under consideration are the template attack [40] and the maximum likelihood principle as part of the stochastic model [126]. Both methods include a profiling stage for the estimation of a key dependent multivariate probability density of the physical observable. This contribution aims to answer the question which method performs best under the same measurable parameter setting such as the number of curves during profiling and key recovery.

Parameters with an impact on efficiency in side channel cryptanalysis are manifold. Among them, (i) the quantity of the leakage (chip dependent), (ii) the quality of the measurement equipment (lab dependent), and (iii) the attack's ability to extract information (method dependent) are seen as the most important ones.

This work is driven by the demand for an objective and systematic methodical performance comparison in identical physical conditions since the *quality* of side channel measurements is one of the most crucial factors in terms of attack efficiency. Both methods are applied to measurements from two separate setups using two different microcontrollers running an AES implementation in software.

The contribution of this chapter is two-fold. First, a systematic performance analysis is carried out for both methods as they were originally proposed. Second, the same performance analysis is repeated by applying improvements for each method under consideration. The most crucial improvement deals with the selection of time instants for the multivariate density. The application of the T-Test for the choice of instants yields significant performance gains for both methods under consideration. For the stochastic approach additionally higher-order analysis is applied to improve capturing of key dependent peaks.

By using the originally proposed attacks, it is revealed that towards a low number of profiling measurements stochastic methods are more efficient whereas towards a high number of profiling measurements templates achieve superior performance results.

As the main result of optimizations, T-Test based templates are the method of choice if a high number of measurements is available for profiling. However, in case of a low number of measurements for profiling,

stochastic methods are an alternative and can reach superior efficiency both in terms of profiling and classification. Moreover, stochastic methods remain applicable even if the number of measurements at profiling is less than the number of subkey dependencies. It is shown that the improved variants are indeed practical, even at a low number of profiling measurements. This is of particular importance when applying these attacks to noisy measurements. It was experimentally proved that the T-Test based attacks yield far better results than the original attacks in such a setting.

This work is based on [55] and the results presented here are already published in [56]. This chapter is organized as follows. Section 6.3 describes how templates and the stochastic methods are applied in the concrete setting for the performance analysis of the original attacks. The testing framework used for performance analysis is presented in Section 6.4. Section 6.5 presents results that were obtained by using the original approach for both methods, whereas Section 6.6 introduces and evaluates the optimizations.

6.2 Previous Work

The measure of the minimum number of measurements needed for key recovery is not new and has been already widely applied in literature to demonstrate the (improved) effectiveness of DSCA variants or to evaluate the effectiveness of countermeasures. For example, this argumentation was used at improvements of the signal-to-noise ratio by multi-bit DPA studied in [96], for the introduction of multi-channel attacks in [4], and for the effectiveness analysis of countermeasures in [86] and [89]. Also in Chapter 5 performance gains are provided in terms of measurements, e.g., at comparing variants of stochastic methods.

The problem of identifying relevant instants as part of a template attack was first discussed by [118]. A different solution for enhancing the efficiency of templates has appeared very recently in [11]. Here, principal component analysis (PCA) has been applied. This new approach is called *principle subspace-based template attack* and performs an eigendecomposition of the covariance matrix in order to identify both the principal directions (eigenvectors) and the variance (eigenvalues).

6.3 Application of Templates and Stochastic Methods

As introduced in Section 3.3 side channel cryptanalysis can be distinguished into *one-stage* methods, without any prior knowledge about the expected side channel leakage that are directly used for key recovery, and *two-stage* methods that make use of a profiling stage to obtain an 'a priori' knowledge on the side channel leakage that can be used for extracting keys later on. Both, templates and stochastic methods are two-stage attacks. For profiling, two-stage methods require a cryptographic device which is identical to the device used at key recovery. The task of this section is to provide the overall approach and the concrete methodical usage for both methods under consideration.

6.3.1 Template Attack

Templates are claimed to be the strongest side channel attack possible from an information theoretic point of view [40]. As part of this thesis, templates have been already introduced in Section 3.3.8. Because of this, the focus in this chapter is on the usage of templates at an AES implementation.

While in case of attacks on stream ciphers, a further requirement is that the profiling device must allow to load keys [40], attacks on AES do not require this, which weakens the assumptions on the adversary's power. In [40] an 'expand and prune' strategy is described that is particularly useful when analyzing stream ciphers. Applying this strategy, profiling and classification build a recurring cycle for sieving key candidates which means in particular that the vast effort of the profiling stage cannot be pre-computed. In contrast, if the attacked key is known to be sufficiently small or assailable in such blocks[1], profiling can be done independently before or after obtaining a measurement trace from the target device. For example, to recover an 128-bit AES key one can pre-compute $2^8 \cdot 16$ instead of (infeasible) 2^{128} templates and - after obtaining a measurement trace - immediately start the classification stage, i.e., the key recovery stage which may take only a few seconds.

It is assumed that profiling is done with one fixed AES key. In this

[1] This is true for many block ciphers.

chapter the subkey dependency k, for which templates are generated, is understood as an intermediate result depending both on input data $x_i \in \{0,1\}^d$ and a (fixed) subkey $k \in \{0,1\}^s$ whereby $s = d$. More concretely, for each value of $(x_i \oplus k) \in \{0,1\}^d$ a template is generated. In this setting, measurements with the same values of x_i belong to the same subkey dependency. This approach is already outlined as *advanced template attack* in Section 5.3.3. It leads to 2^d subkey dependencies.

Improvement 1 (Selection of *Relevant* Instants)

For the microprocessor ATM163 the sum of pairwise differences of the key dependent mean vectors, i.e., the resulting difference vector

$$\sum_{i,l=0,l>i}^{2^d-1} \vec{\mu}_i - \vec{\mu}_l$$

turned out to be not an appropriate basis for choosing the *relevant* points in time (cf. Section 3.3.8). This is due to the fact that positive and negative differences of means may zeroize, which is desirable to filter noise but hides as well valuable peaks that derive from significant signal differences with alternating algebraic sign.

Therefore the sum of *squared* pairwise differences of the key dependent mean vectors is introduced. For each sampled instant j, i.e., the j-th scalar component of the resulting difference vector, one computes

$$\mathrm{sosd}_j := \sum_{i,l=1,l>i}^{2^d-1} (\mu_{ij} - \mu_{lj})^2$$

(also referred to as *sosd* in this work), see Algorithm 6.1. Hiding effects do not emerge anymore at the cost of a non-zero noise floor. Further, large differences get amplified.

Remarks 6.1. (i) For the performance analysis, Algorithm 6.1 is refined in such a way that for each clock cycle at maximum one instant is chosen, namely that one achieving the maximum value of τ_j . (ii) In this chapter m denotes the number of selected instants.

Algorithm 6.1 Instant selection based on the sum of squared differences (sosd).

Input: (i) Mean vectors $\vec{\mu}_0, \ldots, \vec{\mu}_{2^d-1}$ for each subkey dependency k and $\vec{\mu}_k \in \mathbb{R}^p$ for $k \in \{0, \ldots, 2^d - 1\}$
(ii) A significance threshold $\tau_c \in \mathbb{R}$.
Output: A set of $m (m \leq p)$ chosen points of interest $\{t_1, \ldots, t_m\}$

```
1: P ← ∅;
2: for j from 1 to p do
3:    τ_j = 0;
4:    for i from 0 to 2^d − 1 do
5:       for l from i + 1 to 2^d − 1 do
6:          τ_j = τ_j + (μ_ij − μ_lj)^2;
7:       end for
8:    end for
9:    if τ_j ≥ τ_c then
10:      P ← P ∪ j;
11:   end if
12: end for
```

Improvement 2 (Multiple Traces for Classification)

The original template attack provides a classification strategy based on one available measurement trace. While this may be a realistic scenario in the context of stream ciphers[2], the situation is probably less tight in the context of block ciphers. Moreover, in case of an implementation with small side channel leakage, one trace may not be sufficient for a reliable classification. For these reasons, a classification strategy that processes one or several measurement traces is applied according to the usage of (5.19) as introduced in Chapter 5. One decides in favour of the subkey hypothesis k' that maximizes

$$\alpha(x_1, \ldots, x_{N_3}; k) := \prod_{i=1}^{N_3} \text{prob}_{\mathbf{C}_{x_i \oplus k}}(\vec{z}_i) \tag{6.1}$$

[2]Reference [118] presents an amplified attack on stream ciphers for the case of several available traces.

among all $k \in \{0,1\}^s$. For practical purposes one makes use of the multivariate Gaussian probability distribution (see (3.1))

$$\text{prob}_{\mathbf{C}_{x_i \oplus k}}(\vec{z_i}) = \frac{1}{\sqrt{(2\pi)^m \det(\mathbf{C}_{x_i \oplus k})}} \exp\left(-\frac{1}{2}\vec{z_i}^T \mathbf{C}_{x_i \oplus k}^{-1} \vec{z_i}\right), \quad \vec{z_i} \in \mathbb{R}^m, \tag{6.2}$$

with $\vec{z_i} := \vec{i}(x_i \oplus k^\circ) - \vec{\mu}_{x_i \oplus k}$ with $\vec{i}(x_i \oplus k^\circ) \in \mathbb{R}^m$ being the measurement vector at the chosen instants depending on the unknown subkey k°, $\det(\mathbf{C}_{x_i \oplus k})$ denotes the determinant of covariance matrix $\mathbf{C}_{x_i \oplus k}$, and $\mathbf{C}_{x_i \oplus k}^{-1}$ its inverse. Note that due to the construction of templates, the value $x_i \oplus k$ is used for addressing a template in (6.1) and (6.2).

If one is only interested in the ranking of probabilities for different subkeys, combining of (6.1) and (6.2) leads to the key candidate k' that minimizes

$$\ln \alpha(x_1, \ldots, x_{N_3}; k) := \sum_{i=1}^{N_3} \left(\ln\left(\det(\mathbf{C}_{x_i \oplus k})\right) + \left(\vec{z_i}^T \mathbf{C}_{x_i \oplus k}^{-1} \vec{z_i}\right)\right) . \tag{6.3}$$

Concretely, equation (6.3) was used for the computation of the ranking of key hypotheses at key recovery.

6.3.2 Stochastic Methods

The stochastic model and its algorithms are already introduced in Chapter 5. For this performance analysis, the maximum likelihood principle is used and the minimum principle is skipped as it is already proven to be less efficient in [126]. Profiling processes $N = N_1 + N_2$ traces representing a known subkey k and known data $x_1, x_2, \ldots, x_N$ and consists of an estimation on the deterministic side channel leakage and an estimation of the multivariate noise (cf. Table 5.1).

The vector subspace was the same as introduced in Section 5.4.1, i.e., the vector subspace $\mathcal{F}_9$ is spanned by 1 and the bitwise coefficients at the outcome of the AES S-box S. The basis vectors $g_l(x_i \oplus k)$ $(0 \leq l \leq 8)$ are

$$g_l(x_i \oplus k) = \left\{ \begin{array}{ll} 1 & \text{if } l = 0 \\ l\text{-th bit of } S(x_i \oplus k) & \text{if } 1 \leq l \leq 8 \end{array} \right\}. \tag{6.4}$$

The choice of relevant time instants is different to Chapter 5. It is based on a modified sosd algorithm. Algorithm 6.1 is modified so

that $\vec{\mu}_0, \ldots, \vec{\mu}_{2^d-1}$ is replaced by the estimators $\widetilde{\vec{h}^*}(0), \ldots, \widetilde{\vec{h}^*}(2^d-1)$. Further, only at most one instant per clock cycle is selected as mentioned in Remarks 6.1. It is noted that the squared Euclidean norm proposed in Chapter 5 and [126] produces very similar results.

Other parameters are kept fixed, e.g., $N_1 = \frac{N}{2}$ measurements are used for estimating the deterministic part and $N_2 = \frac{N}{2}$ measurements are used for the estimation of the m-variate noise throughout this chapter[3]. Key recovery applies equation (5.19) and (5.20).

6.3.3 Compendium of Differences

Table 6.1 summarizes the fundamental differences of both attacks. Following the notation in [126], templates estimate on the true deterministic part h_t itself, whereas stochastic methods approximate the linear part of h_t in the chosen vector subspace (e.g., $\mathcal{F}_9$) and are not capable of including non-linear parts. Templates build a covariance matrix for each key dependency whereas the stochastic maximum likelihood principle generates only one covariance matrix, hereby neglecting possible multivariate key dependent noise terms. A further drawback may be that terms of the covariance matrix are distorted because of non-linear parts of h_t in $\mathcal{F}_9$.

Table 6.1: Fundamental differences between templates and stochastic methods.

	Templates	Stochastic Maximum Likelihood Principle
Mean	estimation of key dependent means: 256 means	linear estimation of key dependent means in $\mathcal{F}_9$: nine coefficients
Noise	key dependent, characterized: 256 covariance matrices	non-key dependent, characterized: one covariance matrix

[3] One may argue that the choice of instants can be done using all N traces. However, this was not done in this chapter, but it would surely further increase the performance of stochastic methods.

6.4 Performance Evaluation

In this contribution, performance aspects for side channel cryptanalysis are elaborated for the template attack and the stochastic maximum likelihood principle. The goal is to provide a systematic performance comparison with respect to resources[4] needed for a successful attack. An adversary is *successful* if the (unknown) key value is correctly identified at classification.

6.4.1 Metrics, Parameters, and Factors to Study

Hence in determining performance of side channel based techniques one has to answer four related questions first: (i) which are the relevant parameters that have an impact on attack performance, (ii) which of these parameters can be controlled, respectively, their influence measured and hence should be in the scope of our experiments, (iii) on which values for the remaining parameters this case study should be based, and (iv) which metrics should be selected in order to best capture performance aspects?

From the standpoint of resources needed for a successful attack, parameters that influence the success rate are manifold ranging from the measurement equipment and its environment, the knowledge about the target implementation, the configuration of the implementation during profiling, and the concrete methodical approach used for analysis to the number of measurements in the profiling and classification stages.

Among them, (i) the methodical approach, (ii) the number of curves for profiling, and (iii) the number of curves in the classification stage are evaluated. The remaining parameters are chosen to be identical for both methods. Because of this, one is able to exclude any measurement or implementation dependent impact on the analysis results for each setup.

Two methodical approaches are evaluated: the template attack and the stochastic maximum likelihood principle. Concrete parameter settings of both methods additionally include the number and composition of time instants chosen for the multivariate probability density. Point selection algorithms are implemented identically operating on sosd with Algorithm 6.1 selecting at most one point per clock cycle. The number of

[4] It is focused on the number of available traces (side channel quality) since computational complexity is of minor importance for the attacks under consideration.

measurements, both during profiling and key recovery, is regarded as the relevant and measurable parameter. Let N be the number of measurements used in the profiling stage and N_3 the number of measurements used at key recovery. For both, the templates and the stochastic method, the concrete parameter values to study are given in Section 6.4.2.

Metrics

Profiling efficiency is measured (i) as efficiency in estimating the data-dependent side channel portion (refers only to N) and (ii) as ability to determine the correct set of points of interests (refers to N and m). Both metrics relate to reference values obtained for maximal N (referred to as N_{max} below) used in the concrete setting. Classification efficiency (refers to N_3, N and m) is measured as success rate to obtain the correct key value in metric 3.

Metric 1 (Approximation of the mean vector): The first efficiency metric for profiling evaluates the empirical correlation coefficient ρ (2.10) of the mean vectors $\vec{\mu}_i^N \in \mathbb{R}^p$ obtained from N measurements and the reference vectors $\vec{\mu}_i^{N_{max}} \in \mathbb{R}^p$ obtained from the maximal number of measurements available:

$$\frac{1}{2^d} \sum_{i=0}^{2^d-1} \rho(\vec{\mu}_i^N, \vec{\mu}_i^{N_{max}}) \ .$$

The mean vector $\vec{\mu}_i^{N_{max}}$ is also assumed to be the best estimator on the true deterministic leakage for each key dependency (cf. Example 5.2) if this metric is applied to the stochastic method:

$$\frac{1}{2^d} \sum_{i=0}^{2^d-1} \rho(\widetilde{\vec{h}^{*N}}(S(i)), \vec{\mu}_i^{N_{max}}) \ .$$

Note that the numbering of templates is done at the entry of the AES S-box S whereas stochastic methods profile at the output of the AES S-box S. Therefore, the estimator $\widetilde{\vec{h}^{*N}}(S(i))$ is selected for computing the correlation coefficient with $\vec{\mu}_i^{N_{max}}$.

Metric 2 (Selection of time instants): The second metric compares the set of selected points based on N measurements to the reference set obtained using N_{max} measurements and returns the percentage

of points that are located in the correct clock cycle. Here, the reference set of instants is generated by the same methodical approach, i.e., the reference set of instants for the stochastic method is given by the set of instants obtained with the stochastic method by using N_{max} measurements.

Metric 3 (Classification efficiency at key recovery): The success rate at key recovery is empirically determined by classifying N_3 randomly chosen measurements out of the key recovery measurement series. This random choice is repeated one thousand times and the success rate is then defined as the percentage of success in determining the correct key value.

6.4.2 Experimental Design

The performance analysis is applied to two experimental units performing AES in software without any countermeasures. The first experimental unit (device A) is an ATM163 microcontroller [55] that is programmed with the same AES implementation already studied in Section 5.4. A set of more than 230,000 power measurements was recorded for profiling purposes with a fixed AES key and randomly chosen plaintexts. For classification purposes, a second set comprising 3000 measurements with a different fixed AES key was produced. The experimental design is full factorial. The second experimental unit is another 8-bit microcontroller from a different manufacturer (device B) [56]. Furthermore, the power measurements of device B stem from a different, low-noise, measurement setup. A set of 50,000 power measurements was obtained for profiling purposes and a classification set of 100 measurements, both with fixed but different AES keys. Table 6.2 shows all concrete parameter values. However, Sections 6.5 and 6.6 only provide the most relevant results.

6.5 Experimental Results for Original Attacks

6.5.1 Comparison of Profiling Efficiency

Profiling metrics 1 and 2 are summarized in Figure 6.1 and Table 6.3. Metric 1 clearly yields enhanced results for templates which is reasonable as the stochastic model uses only half of the measurements for the

Table 6.2: Concrete parameter values to study

Device	Parameter	Parameter Values
A	N	231k, 50k, 40k, 30k, 25k, 20k, 10k, 5k, 2k[5], 1k[5], 200[5]
A	m	3, 6, 9, x[6]
A	N_3	1, 2, 5, 10
B	N	50k[7], 10k, 5k, 500[5], 100[5]
B	m	x[6]
B	N_3	1, 2, 5

determining the deterministic part. Though less efficient in determining the deterministic part, Table 6.3 indicates the superiority of the stochastic model in terms of selecting the relevant points in time also with a reduced number of measurements.

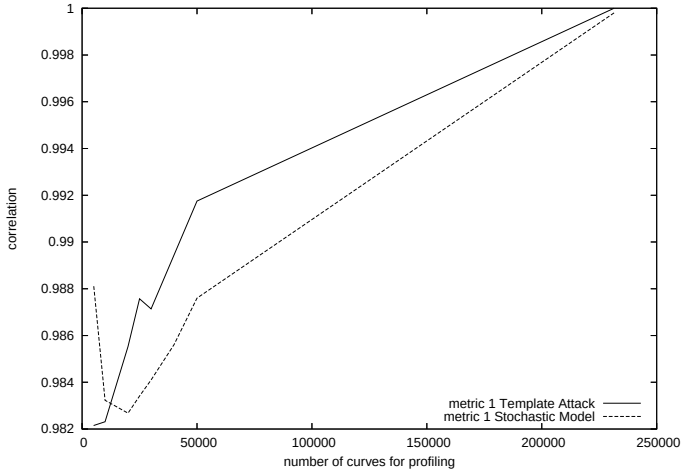


Figure 6.1: Metric 1 for device A for original attacks.

[5]Stochastic Method only.
[6]x = maximum number identified after profiling.
[7]Template Attack only.

Table 6.3: Metric 2 for device A as function of N (Original Attacks).

N	231k	50k	40k	30k	25k	20k	10k	5k
Templates	1	0.89	0.89	0.78	0.67	0.56	0.23	0.23
Stochastic Method	1	1	1	1	1	1	0.67	0.78

6.5.2 Comparison of Classification Efficiency

For the comparison of success rates the parameters N and $N_3 \in \{1, 10\}$ are considered and in every case the optimal number of selected instants that maximizes the success rate. Figure 6.2 shows metric 3 plotted as function of these parameters. One can observe, that each pair of plots intersects at least once. Hence, a general statement on which attack yields better success rates depends on the number of curves that are available in the profiling stage. If a large number of measurements is available (e.g., more than twenty thousand), templates yield higher success rates. If only a small number of measurements is available (e.g., less than twenty thousand), the stochastic method is the better choice.

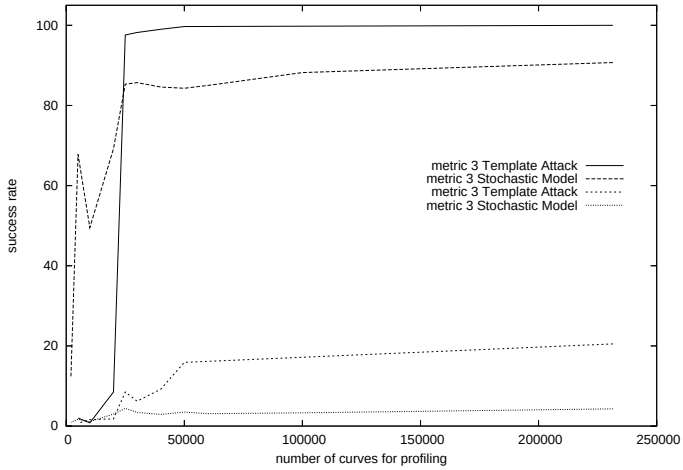


Figure 6.2: Metric 3 for device A for original attacks, $N_3 = 10$ for upper and $N_3 = 1$ for lower curves.

6.5.3 Weaknesses and Strengths

Template Attack: Templates extract by far more information from the measurement traces than the stochastic maximum likelihood principle. Given sufficient traces in the profiling stage, templates are clearly superior to the stochastic method in the classification stage, due to the precise estimation of the average signal and the use of 256 covariance matrices. On the other hand, templates require much more measurements than the stochastic method to reduce the noise and to select the instants contributing to the side channel leakage (see Table 6.3).

Stochastic Method: Stochastic methods learn quickly from a small number of measurements. One weakness lies in the reduced precision due to the linear approximation in a vector subspace. A second weakness is the usage of only a single covariance matrix. If the approximation of the data dependent part is not precise enough, errors in the approximation affect the remaining noise.

6.6 Experimental Results for Optimized Attacks

The maximum efficiency achievable at key recovery for each method is of high importance, so that optimizations were carried out for each method. Particularly, Section 6.5 reveals that the point selection algorithm is crucial for the key recovery efficiency. Both, for templates and the stochastic method, the use of the statistical t-distribution as the basis of instant selection is evaluated in this section. For the stochastic method, additionally, the choice of the vector subspace (single intermediate result vs. two intermediate results) is studied.

Template Attack with T-Test

The template attack's weakness is its relatively poor ability to reduce the noise if the adversary is bounded in the number of measurements in the profiling stage. For small N, the remaining noise distorts the sosd curve, which was used as the basis for the selection of interesting points so far.

The T-Test (cf. Section 2.1.2) is a standard statistical tool to meet the challenge of distinguishing means in noisy signals. When computing the significant difference of two sets (i, j), it does not only consider the distance of their means $\vec{\mu}_i, \vec{\mu}_j$ but their variance $(\vec{\sigma}_i^2, \vec{\sigma}_j^2)$ in relation to the number of measurements (n_i, n_j) as well. Algorithm 6.2 computes the sum of squared pairwise t-differences (also referred to as *sost* in this work) as basis for the point selection instead of sosd. In this algorithm, the T-Test variant for possibly different variances is used.

Algorithm 6.2 Instant selection based on sum of the squared pairwise t-differences (sost).

Input: (i) Mean vectors $\vec{\mu}_0, \ldots, \vec{\mu}_{2^d-1}$ for each subkey dependency k and $\vec{\mu}_k \in \mathbb{R}^p$ for $k \in \{0, \ldots, 2^d - 1\}$
(ii) Variance vectors $\vec{\sigma}_0^2, \ldots, \vec{\sigma}_{2^d-1}^2$ for each subkey dependency k and $\vec{\sigma}_k^2 \in \mathbb{R}^p$ for $k \in \{0, \ldots, 2^d - 1\}$
(iii) Number of measurements n_k belonging to each template $k \in \{0, \ldots, 2^d - 1\}$.
(iv) A significance threshold $\tau_c \in \mathbb{R}$.

Output: A set of $m (m \leq p)$ chosen points of interest $\{t_1, \ldots, t_m\}$

1: $P \leftarrow \emptyset$;
2: **for** j from 1 to p **do**
3: $\quad \tau_j = 0$;
4: $\quad$ **for** i from 0 to $2^d - 1$ **do**
5: $\quad\quad$ **for** l from $i + 1$ to $2^d - 1$ **do**
6: $\quad\quad\quad \tau_j = \tau_j + \left(\frac{(\mu_{ij} - \mu_{lj})}{\sqrt{\frac{\sigma_{ij}^2}{n_i} + \frac{\sigma_{lj}^2}{n_l}}} \right)^2$;
7: $\quad\quad$ **end for**
8: $\quad$ **end for**
9: $\quad$ **if** $\tau_j \geq \tau_c$ **then**
10: $\quad\quad P \leftarrow P \cup j$;
11: $\quad$ **end if**
12: **end for**

Figure 6.3 illustrates the striking difference between sosd and sost for $N = 50,000$ and $N = 10,000$ measurements. The scale of the vertical axis is not the same for all plots, but as one is not interested in compar-

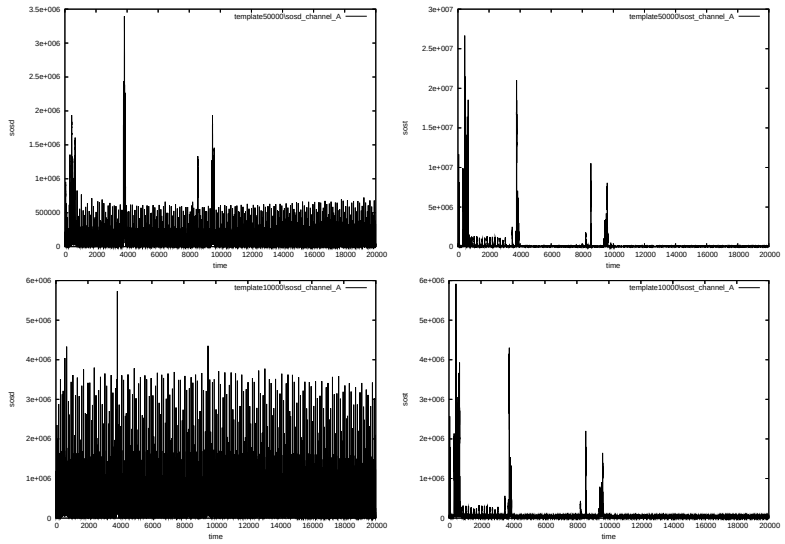


Figure 6.3: Selection of instants with sosd (left) and sost (right) as functions of time, $N = 50000$ (top) and 10000 (bottom).

ing the absolute height of the peaks, this can be disregarded. What is important is the relative distance between the peaks and the noise floor in each curve. While the reduction of N by a factor of five leads to a very distorted sosd signal, the significance of sost in terms of where to find interesting points does not change. Apart from the different scale, the peaks have a virtually identical shape.

High-Order Stochastic Method with F_{17} and T-Test

According to the improvements for templates, Algorithm 6.2 was slightly modified for the use with stochastic methods: Algorithm 6.2 used as input (i) the data dependent estimators $\widetilde{\vec{h}^*}(0), \ldots, \widetilde{\vec{h}^*}(2^d - 1)$, (ii) the empirical variance $\vec{\sigma}^2$ derived from N_1 measurements, and (iii) $n_k = \frac{N_1}{2^d}$. As for templates, this yields a significant improvement of the point selection.

The weakness of the stochastic method with $\mathcal{F}_9$ is the limited precision due to the approximation of the deterministic side channel leakage.

One obvious solution to this problem is to increase the number of dimensions of the vector subspace in order to generate a more precise estimator at the cost of needing more measurements in the profiling stage (trade off problem). But as [126] already analyzed several high-dimensional vector subspaces and concluded that $\mathcal{F}_9$ seems to be most efficient, a different attempt was followed here.

Comparing the sosd curves of the stochastic method and the template attack leads to the optimization done here. Due to the fact that the underlying traces represent only one fixed key, the template attack's sosd curve shows peaks for x_i, $x_i \oplus k$, and $S(x_i \oplus k)$. Since the stochastic method only approximates the data dependent side channel portion at $S(x_i \oplus k)$, it can not track bits through the S-box and hence the point selection algorithm only finds instants contributing to $S(x_i \oplus k)$. The optimization is motivated by the fact that the stochastic method overlooks instants covering the S-box lookup which yields the strongest peaks in the sosd curve of the template attack. The number of dimensions of the vector subspace is increased, but rather than increasing the level of detail at one intermediate result of the AES encryption, a second intermediate result is considered. The selection functions g_l of the 17-dimensional vector subspace $\mathcal{F}_{17}$ are defined as follows:

$$g_l(x_i \oplus k) = \left\{ \begin{array}{ll} 1 & \text{if } l = 0 \\ l\text{-th bit of } S(x_i \oplus k) & \text{if } 1 \leq l \leq 8 \\ (l-8)\text{-th bit of } x_i \oplus k & \text{if } 9 \leq l \leq 16 \end{array} \right\}. \tag{6.5}$$

As desired, additional clear peaks during the S-box lookup $(x_i \oplus k)$ were now found by the point selection algorithm.

6.6.1 Templates vs. T-Test based Templates

When comparing the optimized templates with the original attack is is figured out how the new point selection algorithm contributes to the performance.

Profiling Efficiency

Table 6.4 shows the efficiency of both attacks in the profiling stage using metric 2. The numbers clearly indicate the superiority of the improved

version, the T-Test based template, in terms of selecting the right instants and hence, in the profiling stage. Considering Figure 6.3 again, the improved profiling efficiency obviously derives from the enhanced ability to suppress noise in the physical channel.

Table 6.4: Metric 2 for device A as function of N (Improvements for Templates).

N	231k	50k	40k	30k	20k	10k	5k
Template Attack	1	0.89	0.89	0.78	0.56	0.23	0.23
T-Test Templates	1	1	1	1	1	1	1

Classification Efficiency

Here, classification success rates are given in Figure 6.4. The attention is restricted to variations of N, $N_3 \in \{1, 10\}$ for the sake of clarity, and, each time, the optimal number of selected instants to maximize the success rates. For small N, e.g., N smaller than thirty thousand, the improved profiling of the optimized attack clearly leads to a higher success rate at classification.

6.6.2 First-order Stochastic Method vs. T-Test based High-order Stochastic Method

When comparing the optimized stochastic method with the original attack, the choice of the vector sub-space and the T-Test based point selection is evaluated.

Profiling Efficiency

Table 6.5 shows the profiling efficiency of both attacks in metric 2. The numbers indicate that the optimized method has a significantly advanced ability to select the relevant points in time, in particular when processing only a small number of profiling measurements.

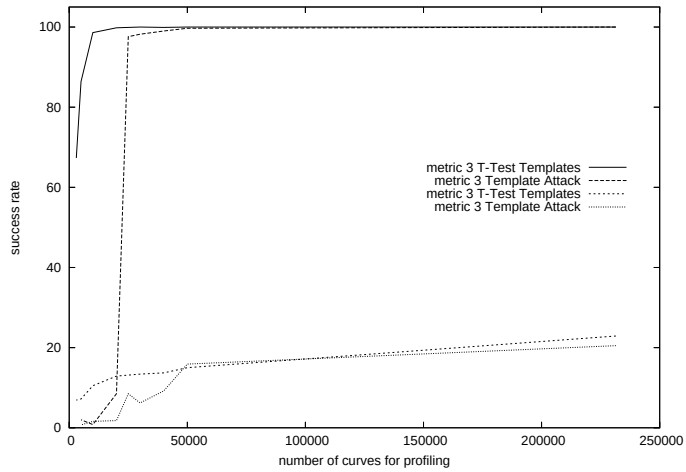


Figure 6.4: Metric 3 for device A for the optimized template attack, $N_3 = 10$ for upper and $N_3 = 1$ for lower curves.

Table 6.5: Metric 2 for device A as function of N (Improvements for Stochastic Method).

N	231k	50k	40k	30k	20k
Stochastic Method	1	1	1	1	1
T-Test Stochastic Method	1	1	1	1	1
N	10k	5k	2k	1k	200
Stochastic Method	0.67	0.78	0.67	-	-
T-Test Stochastic Method	1	0.9	1	1	0.5

Classification Efficiency

Classification success rates of both the original and the optimized method are compared here. Again, it is focused on variations of N, $N_3 \in \{1, 10\}$, and, each time, the optimal number of selected instants to maximize the success rates. Figure 6.5 shows metric 3 plotted as function of these parameters.

The benefit of generating eight additional base vectors with respect to the S-box input and using sost instead of sosd is clearly visible. Following

the profiling efficiency (see Table 6.5), the efficiency in the classification stage is significantly increased. Particularly, for N larger than thirty thousand and $N_3 = 10$, the T-Test based high-order stochastic method clearly exceeds the 90% success rate boundary and finally reaches 100% success.

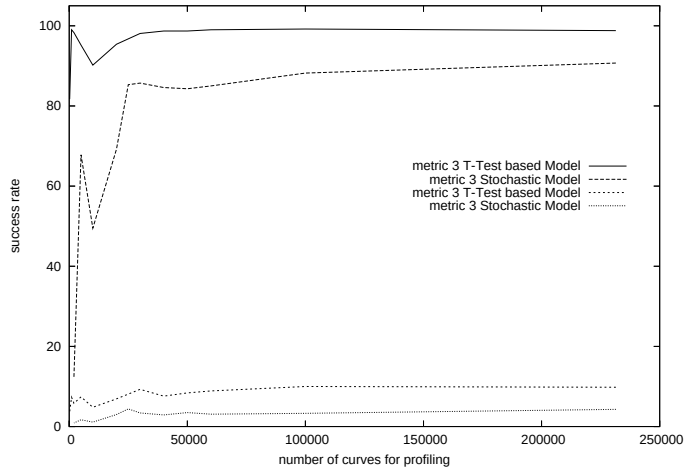


Figure 6.5: Metric 3 for device A for the optimized stochastic methods, $N_3 = 10$ for upper and $N_3 = 1$ for lower curves.

6.6.3 Overall Comparison

In this section the efficiency for the optimized templates and the optimized stochastic maximum likelihood principle is compared in the classification stage. Further, a short summary of the observations is given to serve as an overall survey of this work. Figure 6.6 contrasts the classification efficiency of the optimized attacks using metric 3.

T-Test based templates are the best possible choice in almost all parameter ranges. For small N (e.g., N less than five thousand), however, the T-Test based high-order stochastic method leads to better results. This can be of high importance if an adversary is limited at profiling. Further, it is worth pointing out that the improved version of the stochastic method still operates successfully using extremely small N. For ex-

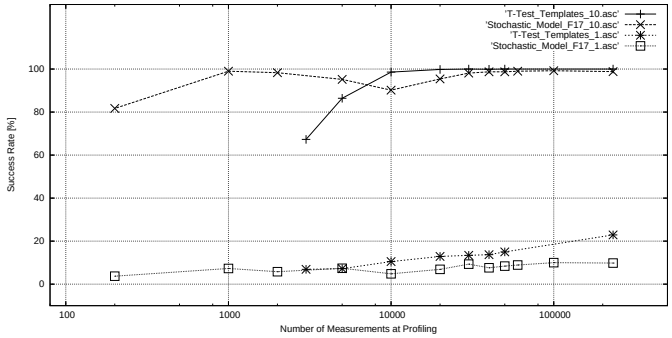


Figure 6.6: Overall comparison: metric 3 for device A on a logarithmic scale, $N_3 = 10$ for upper and $N_3 = 1$ for lower curves.

ample, using $N = 200$ profiling measurements and $N_3 = 10$ curves for classification it still achieves a success rate of 81.7%.

To stress the impact of the factor 'measurement quality' success rates of the optimized methods are also presented on the basis of measurements with device B that stem from the low-noise setup. Table 6.6 provides the attack efficiencies in metric 3 for variations of N, $N_3 \in \{1, 5\}$, and, each time, the optimal number of selected instants to maximize the success rates.

Table 6.6: Metric 3 for device B as function of N. These results are taken from [56].

		50k	10k	5k	500	100
T-Test Templates	$N_3 = 1$	94.8	93.0	88.2	-	-
	$N_3 = 5$	100.0	100.0	100.0	-	-
T-Test based	$N_3 = 1$	-	57.5	60.1	46.8	27.1
Stochastic Method	$N_3 = 5$	-	100.0	99.9	100.0	96.5

Besides the fact that the relation of N to success rate of both attacks is better by orders of magnitude when using low-noise measurements, it is worth noting, that the improved stochastic method still classifies keys successfully, even if the profiling has been done with as little as $N = 100$ curves which is far less than the number of subkey hypotheses.

Chapter 7

A Model on Physical Security Bounds against Tampering

7.1 Contribution

This chapter is dedicated to the modelling of fault channel cryptanalysis in integrated circuits. Its main contribution is to present an adversary model with a strong focus on fault injection techniques based on radiation and particle impact and to define physical security parameters against tampering adversaries. This work is driven by the question of what kind of circuit based security can ever be guaranteed if *all* computations are vulnerable towards fault injection?

If a cryptographic device is used in a hostile environment, special properties for the device are required to ensure a certain level of physical security for the storage and processing of cryptographic keys. For the theoretical perspective refer to the concepts on *read-proof hardware* and *tamper-proof hardware*, as given in [54]. *Read-proof hardware* prevents an adversary from reading internal data and *tamper-proof hardware* prevents the adversary from changing internal data. Moreover, this chapter uses the term *tamper-resistant hardware* as a 'relaxed' term for *tamper-proof hardware*, i.e., the hardware is resistant to tampering to a certain extent. Such bounds are made more precise in this work.

In a tamper-proof implementation, fault injections are not feasible per definition. However, in real life, practical experiments have shown that tamper resistance is hard to achieve. Many contributions (e.g., [74, 12, 137, 14, 121, 138]) have reported that integrated circuits are vulnerable to fault injections. Such findings are related to the development of devices for the use in aerospace and high-energy physics which have to be tolerant on particle radiation impact during operation [83, 84]. In contrast to applications developed for safety and reliability reasons, security applications have to withstand an active malicious adversary. As discussed in Section 7.2, the *Algorithmic Tamper-Proof (ATP)* security model [54] does only partly give a framework for existing attacks.

Implementation security is different from algorithmic security. For the assessment of implementation security, properties of the concrete layout and timing of the circuit are needed. Current fault injection techniques as presented in Section 3.4.2 are reconsidered to build an unified adversary model based on [79, 82] as a first step towards bridging the gap between the theoretical framework of [54] and real-world experiences. Hereby it is assumed that any kind of data memory can be tampered with in a probabilistic sense and that the adversary is able to induce faults at

any internal state and computation of the physical device. By doing so, modelling comprises the manifold nature of faults as well as Differential Fault Analysis (DFA) [24, 114] more adequately for integrated circuits.

Early results of this work are published in [79, 82]. The main contents of this chapter are already published in [80]. In this chapter, a physical model for fault injection based on radiation and particle impact is introduced. It is assumed that fault injection can be both applied prior and during computations of a cryptographic device which is a realistic assumption that should also be included in provable security models. Strategies for countermeasures are evaluated as result of the new physical model. However, it is still an open question whether physical quantities can be formally tied to security notions in a realistic physical model for tampering. It is hoped that this framework is useful to both map concrete impact probabilities of a given circuit as well as to improve the circuits' layout.

7.2 Previous Work

The model of *Algorithmic Tamper-Proof (ATP)* security was introduced in [54]. It assumes that devices are built using two different components, one being tamper-proof but readable, and the other being read-proof yet tamperable. Only data that is considered to be universally known (i.e., public data) is tamper-proof beyond the reach of the tampering adversary. Other data is subject to tampering, i.e., fault induction. ATP Security defines a powerful tampering adversary who is able to initiate three commands: $\mathsf{Run}(\cdot)$, i.e., the cryptographic computation, $\mathsf{Apply}(\cdot)$, i.e., the fault injection, and $\mathsf{Setup}(\cdot)$. The adversary knows all construction details, especially, each bit-position in the device's memory. It is concluded in [54] that a component is needed which is both read-proof and tamper-proof to achieve general Algorithmic Tamper-Proof (ATP) Security.

The main limitation of [54] is caused by the fact that the command $\mathsf{Run}(\cdot)$ itself is assumed to be invulnerable to fault injection. In practice, there is no reason for the adversary not attacking $\mathsf{Run}(\cdot)$ itself. Actually, standard scenarios of Differential Fault Analysis (DFA) induce faults *during* the cryptographic computation [24, 114]. Such a setting becomes especially important in case of tampering with memory-constrained de-

vices as, e.g., a modification prior to $\mathsf{Run}(\cdot)$ can hardly affect *only* the last round of DES. In [54], tamper-proofing a signature (or decryption) scheme is part of the command $\mathsf{Run}(\cdot)$ which first checks the integrity of the storage using a verification algorithm. If so, the signature (or decryption) algorithm is computed yielding an output as result. Otherwise, self-destruction of the device is invoked. In case the verification algorithm is subject to fault injection, too, the tamper-proofing solution of the ATP model does not hold anymore.

Reference [54] also discusses restrictions of the model assuming limited capabilities of the adversary, for instance, a restriction to perform only a probabilistic flipping of bits in the device's memory. The type of DFA discussed in [54] requires the strong assumption that the memory type is significantly asymmetric. For this type of DFA, [54] argues that checking for faults is sufficient for ATP security, even if the device is not equipped with a self-destruct capability. Faults can be very precisely induced, e.g., by optical fault induction, as reported in a recent survey on hardware security analysis [138]. Therein, it is demonstrated that any individual bit of SRAM memory can be changed to a definite state by injection of light. Both the target states '0' and '1' could be set, just by a lateral adjustment of the light spot.

For previous work on fault channel cryptanalysis refer to Section 3.4. A valuable survey on countermeasures can be found in [14].

7.3 Adversary Model

The adversary model presented is an extended version of [79]. By assumption the physical device $\mathcal{D}$ is encapsulated. Especially, it does neither offer a logical nor a physical interface to modify the internal memory or the internal construction of $\mathcal{D}$. The set-up for attacks based on fault analysis consists of i) the physical device $\mathcal{D}$ under test, ii) a reader device for the data communication interface, and iii) a fault injection set-up. Optionally, iv) a monitoring set-up is used by the adversary to analyze the fault induction process and its effects, e.g., by measuring side channel leakage. The set-up as well as the information flow is illustrated in Figure 7.1 and described in more detail below.

The adversary is denoted by $\mathcal{A}$. By assumption $\mathcal{A}$ has physical access to the device $\mathcal{D}$ under attack and can run a high number N of instances

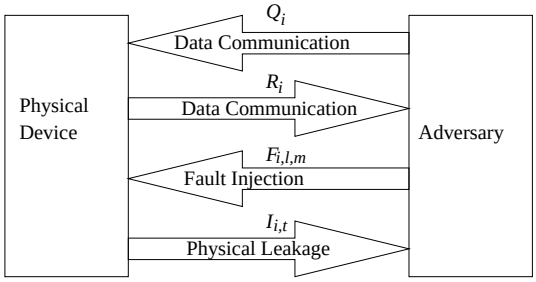


Figure 7.1: Information flow at a fault analysis set-up.

of a security service $\mathcal{S}$. Each instance is initiated by a query Q_i of $\mathcal{A}$ and completed by $\mathcal{D}$ at time T_i returning a response R_i, where $i \in \{1, \ldots, N\}$. $\mathcal{A}$ applies a probabilistic physical interaction process aiming at disturbing the intended computation of $\mathcal{S}$. $\mathcal{A}$ may be able to monitor the effects caused by physical interaction using auxiliary means, e.g., by observing the instantaneous leakage $I_{i,t}$ of the implementation with a monitoring set-up at time t. If necessary, $\mathcal{A}$ applies cryptanalytical methods for a final analysis.

Moreover, it is assumed that $\mathcal{A}$ is able to perform M multiple fault injections with a fault injection set-up and let L be a small number of spatially separated fault injection set-ups that can be operated in parallel. The distinct fault injections during one invocation of $\mathcal{S}$ are numbered with $\mathcal{F}_{i,l,m}$, where $l \in \{1, \ldots, L\}$ and $m \in \{1, \ldots, M\}$, and occur at the times $\{t_{i,1,1}, \ldots, t_{i,L,M}\}$ with $t_{i,1,1} \leq \cdots \leq t_{i,L,M} \leq T_i$.

$\mathcal{A}$ is an active adaptive adversary, i.e., both the queries Q_i as well as the parameters of $\mathcal{F}_{i,l,m}$ can be chosen adaptively. It is worth pointing out that the leakage $I_{i,t}$ is typically not yet available for the configuration of $\mathcal{F}_{i,l,m}$ at the same instantiation of $\mathcal{S}$, unless a more demanding real-time analysis is applied.

For the physical device $\mathcal{D}$ an implementation in circuitry is considered. The target circuit $\mathcal{C}$ that is part of $\mathcal{D}$ consists of interconnected Boolean gates and memory cells[1]. Each spatial position within $\mathcal{C}$ is uniquely represented in three dimensional co-ordinates $\vec{x} = (x, y, z)$. Processing of $\mathcal{C}$ is modelled by the transition states of the circuit at time

[1] In a refined model one may distinguish different types of memory elements such as flip-flops, RAM, flash and EEPROM.

t, i.e., by using four dimensional co-ordinates $(\vec{x}, t)$. The state of the circuit S_t at time t is given by the contents of the memory cells. Introducing glitches or short circuits affects Boolean gates and thereby eventually cause computational faults. Such faults can result in erroneous states stored in memory cells. Faults affecting memory cells directly cause a transition from memory contents S_t to $f(S_t)$ with $S_t \neq f(S_t)$. Fault induction itself is a probabilistic process with a certain success probability that depends on the circuit $\mathcal{C}$, the underlying physical process $\mathcal{P}$ used for fault injection and the configuration of the fault analysis set-up $\mathcal{F}_{i,l,m}$.

Summarizing, the information channels are

1. the *Query Channel* modelling $\mathcal{A}$ sending the query Q_i to $\mathcal{D}$,
2. the *Response Channel* modelling $\mathcal{A}$ receiving the response R_i of $\mathcal{D}$,
3. the *Fault Channel* modelling $\mathcal{A}$ applying physical fault injection processes $\mathcal{F}_{i,l,m}$ targeting $\mathcal{D}$, and
4. the *Monitoring Channel* modelling $\mathcal{A}$ receiving physical leakage $I_{i,t}$ of $\mathcal{D}$.

Informally speaking (a more precise definition for a digital signature scheme is given below), an adversary $\mathcal{A}$ is *successful*, if the insertion of faults either i) yields access to a security service $\mathcal{S}$ without knowledge of the required secret or ii) yields partial information about the secret.

7.3.1 Objectives of the Adversary

As introduced in Section 3.4, manifold attack scenarios for fault analysis have already been proposed. The core of all these scenarios includes a loop of an instantiation of the security service $\mathcal{S}$ and a sequence of fault injection processes $\mathcal{F}_{i,l,m}$. A classification into three main categories, namely Simple Fault Analysis (SFA), Successive Simple Fault Analysis (SSFA) and Differential Fault Analysis (DFA), can be found in [79].

For concreteness, a digital signature scheme is considered that is defined as a triple of algorithms (Gen, Sig, Ver) with key generation algorithm Gen, signing algorithm Sig and verifying algorithm Ver. Let (pk, sk) be public and secret key of the signing algorithm Sig that is implemented as security service $\mathcal{S}$ of $\mathcal{D}$ in the circuit $\mathcal{C}$.

In this model, fault injection can both be done *prior* and *during* the computation of a digital signature. Fault injection may modify the computation of $\mathcal{C}$ (resulting in wrong intermediate data of the computation) as well as the actual memory contents of $\mathcal{C}$. The chosen message m_i, part of Q_i, is used for signature generation and s_i, part of R_i, generated by $\mathcal{D}$ such that $s_i \leftarrow Sig_{sk}(m_i)$. If $Ver_{pk}(m_i, s_i)$= yes, the computation of the signature generation is correct, otherwise it is not.

$H \leftarrow \{\}; I \leftarrow \{\}; State \leftarrow \epsilon;$
$for\ i = 1 \ldots N$
$\quad (State, \mathcal{F}_{i,l,m}, m_i) \leftarrow \mathcal{A}(State, pk, H)$
$\quad I \leftarrow I \cup \{(m_i)\}$
$\quad (s_i) \leftarrow Sig_{sk}(m_i)$
$\quad H \leftarrow H \cup \{(m_i, s_i, Ver_{pk}(m_i, s_i))\}$
$(m, s) \leftarrow \mathcal{A}(pk, H)$
$m \notin I$ and $Ver_{pk}(m, s) = yes$

Figure 7.2: Tampering Attack against a Digital Signature Scheme based on adaptively chosen messages

As shown in Figure 7.2, $\mathcal{A}$ invokes N instantiations of the signature computation. For each run, $\mathcal{A}$ configures $F_{i,l,m}$, chooses m_i and runs the signature computation $Sig_{sk}(m_i)$. Though configuration of $F_{i,l,m}$ may be done before the signature computation, fault injection of $F_{i,l,m}$ may also be effective during signature computation. $\mathcal{A}$ stores $(m_i, s_i, Ver_{pk}(m_i, s_i))$ for the analysis step. $\mathcal{A}$ is successful with N instantiations of $Sig_{sk}(m_i)$ if a valid signature s is generated for a new message m which was not been used before. In practice, fault analysis against digital signature schemes may be even stronger, as result, $\mathcal{A}$ then outputs sk.

7.3.2 Physical Means of the Adversary

In this section it is detailed on the physical modelling of the circuit $\mathcal{C}$ and the physical interaction process $\mathcal{P}$. Let assume a strong adversary $\mathcal{A}$ possessing a map of $\mathcal{C}$, including a behavioral simulation for any time t. $\mathcal{A}$ is then able to configure the setup $F_{i,l,m}$ for fault injection according

to the known circuit layout and processing times.

Interaction Range

According to FIPS 140-2 [105] the concept of the *cryptographic boundary* that encloses all security relevant and security enforcing parts of an implementation is used. Additionally, a second boundary called *interaction boundary* is defined that is specific for each physical interaction process. If the adversary does not pass the interaction boundary, the physical interaction has no effect on the cryptographic device. The interaction boundary can be an outer boundary of the cryptographic boundary, as, e.g., in case of temperature which affects the entire cryptographic module. Interaction with light is only feasible if a non-transparent encapsulation is partially removed, e.g., the chip is depackaged. Because of the limited range of the interaction, interaction processes using particles with non-zero mass may require the removal of the passivation and other layers which breaches the cryptographic boundary.

The means of $\mathcal{A}$ can be manifold. The main limitations are obviously caused by the technical equipment available. Because of this the non-invasive adversary, the semi-invasive adversary, and the invasive adversary are defined according to earlier work (e.g., [137, 79]) on fault induction.

Let $\mathcal{A}$ choose a physical interaction process $\mathcal{P}$. $\mathcal{A}$ uses *non-invasive* means if the interaction boundary of $\mathcal{P}$ is an outer boundary of the cryptographic boundary. The non-invasive adversary is denoted by $\mathcal{A}_{non-inv}$. $\mathcal{A}$ uses *invasive* means if the interaction boundary of $\mathcal{P}$ is an inner boundary of the cryptographic boundary. The invasive adversary is denoted by $\mathcal{A}_{inv}$. A *semi-invasive* adversary $\mathcal{A}_{semi-inv}$ uses light or electromagnetic fields as the interaction process and is a special case of $\mathcal{A}_{non-inv}$. Table 7.1 provides a summary on physical means.

In circuitry, modifications of charges, currents and voltage levels may cause faults of the implementation. Modification of charges can be invoked by injecting charged particles or photons. For example, the underlying physical process for optical fault induction is the photoelectric effect, where injected photons are absorbed by the electronic semiconductor that in turn excites electrons from the valence band to the conduction band. Modification of currents can result from manipulating at the electrical circuit or by electromagnetic fields. Modification of in-

Table 7.1: Physical means according to the interaction range of an adversary

Adversary	Physical Means
$\mathcal{A}_{non-inv}$	glitches at external interfaces, changes of the environmental conditions
$\mathcal{A}_{semi-inv}$	light, electromagnetic radiation
$\mathcal{A}_{inv}$	active probes, charged particle beams

ternal voltage levels within the cryptographic boundary are feasible by microprobing or the use of more sophisticated equipment, as focused ion beams. Note that often cumulative effects are needed to induce a fault, e.g., sufficient free carriers have to be generated or driven to load or unload a capacitance of the circuit. In the general case, multiple fault injections cannot be considered as stochastically independent single fault injections, especially if their effects overlap in time or space.

Spatial Resolution

If a special volume dV of the circuit $\mathcal{C}$ is targeted by the adversary, optimizing the success rate requires that the physical interaction process needs to be applied to the cryptographic device with a good resolution in space. The following considerations are mostly suited for light, electromagnetic fields and charged particles as interaction processes.

$F(\vec{x}, E, t)$ is used to model the spatial, energetic and temporal density[2] of identical physical particles[3] as a function of a three-dimensional position vector $\vec{x} = (x, y, z)$, energy E and time t. Before the impact on $\mathcal{C}$ takes place the movement of the density is given by the three-dimensional velocity vector $\vec{v} = (v_x, v_y, v_z)$. For example, $F(\vec{x}, E, t)$ may describe a mono-energetic[4] light beam of photons that is injected into the circuit during a short amount of time.

Without loss of generality the circuit $\mathcal{C}$ is assumed to be in line with the two-dimensional $x - y$ plane (as seen in Figure 7.3) at $z = 0$.

[2]The number of particles per space unit, per energy unit and per time unit.
[3]Correspondingly, one may consider a movement of a wave.
[4]The energy distribution can be modelled with the δ-function $\delta(E - E_0)$, i.e., all particles have energy E_0.

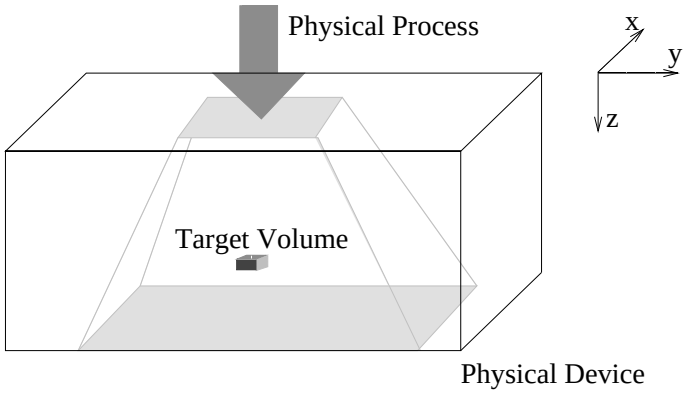


Figure 7.3: Impact of the particle beam into the circuit

The z-axis with $z \geq 0$ quantifies the penetration depth. An interaction process $\mathcal{P}$ of $F(\vec{x}, E, t)$ with a composition of electronic semiconductor material at position $\vec{x}$ is described by a differential cross section $\mathrm{d}\sigma(\vec{x})$, defined as $\mathrm{d}\sigma(\vec{x}) = \frac{dN(\vec{x})}{N(\vec{x})}$, wherein $dN(\vec{x})$ is the number of interacting particles per time unit dT and $N(\vec{x})$ is the number of particles that cross the area dA per time unit dT. Assuming that dA lies in a $x - y$ plane on the surface of $\mathcal{C}$ ($z = 0$), $N(\vec{x})$ is derived by $N(\vec{x}) = \int_0^{v_z \cdot dT} dz \int_{dA} dx\, dy \int_0^{\infty} dE\ F(\vec{x}, E, t)$.

Next, one has to answer on the success probability to hit a target volume dV of $\mathcal{C}$ that is located at depth z with depth extension dz and spanning an area dA. During transfer through the circuit, incident particles are partly absorbed, reflected and transmitted. Interaction processes with matter cause a decrease and spread of the energetic and spatial distribution of $F(\vec{x}, E, t)$ with increasing penetration range in $\mathcal{C}$. The interrelationship of $F(\vec{x}, E, t)$ as a function of the penetration depth z is complex and does typically not solely depend on *one* interaction process. It is assumed that $F(\vec{x}, E, t)$ can be predicted for $z > 0$, e.g., by using a Monte-Carlo simulation of particles' movements by including the most important interaction processes as well as the circuit layout. The spatial spreading of particles due to interactions shall be bounded by $\Delta A(z)$ for any $x-y$ plane within $\mathcal{C}$. Accordingly, the energetic spread shall be bounded by $\Delta E(z)$. The differential cross section also depends

on the energy E, so that it is noted $\mathrm{d}\sigma(\vec{x}, E)$ from now on.

Then, the number of interacting particles in $dV = dz\, dA$ is

$$N_{dV} = \int_z^{z+dz} dz' \int_{dA} dx'\, dy' \int_0^{\Delta E(z')} dE'\, F(\vec{x'}, E', t)\, d\sigma(\vec{x'}, E') \; .$$

Let ΔV be the overall volume of $\mathcal{C}$ that is affected by the physical interaction process. Accordingly, in the volume ΔV one obtains

$$N_{\Delta V} = \int_0^{\Delta z} dz' \int_{\Delta A(z')} dx'\, dy' \int_0^{\Delta E(z')} dE'\, F(\vec{x'}, E', t)\, d\sigma(\vec{x'}, E')$$

with Δz being the thickness of $\mathcal{C}$. The probability to cause an interaction process within the volume dV that is located between depth z to $z + dz$ with area extension dA given the overall affected volume ΔV with $N_{\Delta V} \neq 0$ is

$$p_V = \frac{N_{dV}}{N_{\Delta V}} = \frac{\int_z^{z+dz} dz' \int_{dA} dx'\, dy' \int_0^{\Delta E(z')} dE'\, F(\vec{x'}, E', t)\, d\sigma(\vec{x'}, E')}{\int_0^{\Delta z} dz' \int_{\Delta A(z')} dx'\, dy' \int_0^{\Delta E(z')} dE'\, F(\vec{x'}, E', t)\, d\sigma(\vec{x'}, E')} \tag{7.1}$$

Example 7.1. Mono-energetic beam with exponential attenuation in homogeneous material: $F(\vec{x'}, E', t) = F_0\, \delta(E' - E_0)\, e^{-az'}$ with $a = (10\ \mu m)^{-1}$, $\Delta A(z') = 10\ \mu m^2$, $\Delta z = 100\ \mu m$, $dA = 0.02\ \mu m^2$, $dz = 0.1\ \mu m$, $z = 20\ \mu m$ and $\sigma(\vec{x'}, E') = \sigma_0$. Then, one obtains

$$p_V = \frac{N_{dV}}{N_{\Delta V}} = \frac{dA\, e^{-a\, z}(1 - e^{-a\, dz})}{\Delta A\, (1 - e^{-a\, \Delta z})} \approx 2.69 \cdot 10^{-6} \; .$$

Spatial and Timing Resolution

So far, spatial resolution has been considered. Often additionally knowledge about the timing resolution is required, e.g., the physical interaction process has to be induced during a specific time frame dt of the computation of the implementation, i.e., within the time interval $[t, t + dt]$.

When considering timing resolution in addition to spatial resolution in (7.1) one obtains

$$p_{VT} = \frac{\int_t^{t+dt} dt' \int_z^{z+dz} dz' \int_{dA} dx'\, dy' \int_0^{\Delta E(z')} dE'\, F(\vec{x'}, E', t')\, d\sigma(\vec{x'}, E')}{\int_{-\infty}^{\infty} dt' \int_0^{\Delta z} dz' \int_{\Delta A(z')} dx'\, dy' \int_0^{\Delta E(z')} dE'\, F(\vec{x'}, E', t')\, d\sigma(\vec{x'}, E')} \tag{7.2}$$

as corresponding probability.

Example 7.2. Continuing the previous example with

$$F(\vec{x'}, E', t') = \begin{cases} F_0\, \delta(E' - E_0)\, e^{-az'}, & \text{if } t \le t' \le t + \Delta T \\ 0, & \text{otherwise} \end{cases}$$

with $dt = 10\ ns$ and $\Delta T = 100\ ns \Longrightarrow p_{VT} \approx 2.69 \cdot 10^{-7}$.

Immediate Consequences

- If $F(\vec{x}, E, t')$ does not reach the target area dV it follows that $p_{VT} = 0$.
- If $F(\vec{x}, E, t')$ is uniform in space and time and $\frac{dV}{\Delta V} \ll 1$ and $\frac{dt}{\Delta T} \ll 1$ then $p_{VT} \ll 1$ (e.g., in case of thermal radiation). It follows, that $p_{VT} \ll 1$ for $\mathcal{A}_{non-inv}$.

Sensitive and non-sensitive volumes of a circuit

'Sensitive' and 'non-sensitive' volumes of the circuit $\mathcal{C}$ during computation of $\mathcal{S}$ should be distinguished. A *sensitive volume* of the circuit at time t is composed of Boolean gates and memory cells that are used during computation of the security service $\mathcal{S}$ at the time t. The complementary set of volumes in $\mathcal{C}$ at time t is defined as *non-sensitive volume* of the circuit. As a consequence, physical interaction processes in non-sensitive volumes do not lead to a computational fault of $\mathcal{S}$, whereas physical interaction processes in sensitive volumes can have an impact on the computation of $\mathcal{S}$. In a refined version of (7.2) this fact can be included by neglecting non-sensitive volumes of the circuit at time t.

7.4 Physical Security Bounds

As already outlined, a strong adversary $\mathcal{A}$ is assumed that possesses a map of $\mathcal{C}$ including a behavioral simulation that also indicates sensitive

and non-sensitive volumes of a circuit $\mathcal{C}$ for any time t. Given these means, $\mathcal{A}$ is able to perform a vulnerability analysis of $\mathcal{C}$ and to identify tampering attack paths of $\mathcal{C}$.

For security notions, metrics are needed to quantify physical properties of $\mathcal{C}$. Defining such quantities for a circuit $\mathcal{C}$ is strongly dependent on the concrete layout and has to consider all feasible attack paths, i.e., the set of all admissible events for $\mathcal{A}$. Suitable metrics of $\mathcal{C}$ could be, but are not limited to (i) the size of target gates, (ii) the attacking time frame for target gates, (iii) the smallest Euclidean distance between target gates and the cryptographic boundary of $\mathcal{C}$, and (iv) the smallest Euclidean distance between target gates and other sensitive volumes of $\mathcal{C}$.

A circuit $\mathcal{C}$ implementing a security service S is said to be *statistically secure in the average case against an (N, L, M)-limited tampering adversary* if a negligible function $\text{negl}(\mathcal{C}, N, L, M)$ exists for all physical interaction processes $\mathcal{P}$, such that the success probability of a fault analysis scenario is bounded by $\text{negl}(\mathcal{C}, N, L, M)$. For concreteness, if event E is the fault analysis scenario against a Digital Signature Scheme based on adaptive chosen messages as depicted in Figure 7.2, then $\text{Prob}(E) \leq \text{negl}(\mathcal{C}, N, L, M)$ for the given circuit $\mathcal{C}$. As previously said, the function $\text{negl}(\mathcal{C}, N, L, M)$ depends on the concrete circuit layout. It is still an open question whether physical quantities can be formally tied to security notions in a realistic model for physical tampering.

7.4.1 Evaluation of Countermeasure Strategies

Generic passive and active physical defense strategies are considered that result from physical means as detailed in Section 7.3.2. Passive defense strategies (fault prevention) aim at significantly reducing the success probability for fault injection. Active defense strategies require that $\mathcal{D}$ is capable to detect computational errors resulting from faults (error detection) or the presence of abnormal conditions that may lead to faults (fault detection). In any case, reliable defense strategies should be part of the construction of $\mathcal{D}$. Combinations of these defense strategies are feasible, especially as most strategies have an impact on different parameters in (7.2). The decision whether or not the device shall enter a permanent non-responsive mode in case of error or fault detection depends on the concrete impact probability as well as the concrete security

service. It is a matter of risk evaluation.

Table 7.2: Passive and Active Defense Strategies

Strategy	Impact on Parameter	Security Objective
Shrinking	dA, dz and N	fault prevention
Passive Encapsulation	z and $\mathrm{d}\sigma(\vec{x}, E)$	fault prevention
Timing Modifications	t, dt and N	fault prevention
Error Detection Codes	L and N	error detection
Physical Duplication	L and N	error detection
Repeating Computations	M and N	error detection
Sensors	$\Delta A(z)$ and N	fault detection

Shrinking

Due to the shrinking process, integrated circuits become more and more compact. Shrinking decreases the target volume $dA \cdot dz$. Upcoming chip technology is based on 90 nm structures. For comparison, a focus of a laser beam on the chip surface of 1 μm was reported in [137] for an optical fault injection setup. Due to the limited spatial resolution, multiple faults at neighboring gates are much more likely to occur than single faults at the target, resulting in an increase of N. Note that shrinking may enhance the sensitivity of the circuit so that less free carriers or currents are needed for fault injection.

Passive Encapsulation

Passive encapsulation aims at the absorption or reflection of the interaction process before its effects reach the target area, i.e., $F(\vec{x}, E, t)$ should not reach the target area dV at depth z (resulting in $p_{VT} = 0$ in (7.2)). Such an encapsulation has to be constructed within the cryptographic boundary of the device to prevent it from the reach of $\mathcal{A}_{non-inv}$ and $\mathcal{A}_{semi-inv}$. One approach includes shields that are non-transparent in a

broad light spectrum and prevent throughpassing of photons, i.e., aiming at high values of $\mathrm{d}\sigma(\vec{x}, E)$ within the shield. A simpler design intent is to place security critical parts in center of the chip to prevent both attacks from the front as well as from the back-end side of the chip. If considering different physical interaction processes $\mathcal{P}$, the range of $F(\vec{x}, E, t)$ in semiconductor materials has to be evaluated, i.e., the average value of the depth to which a particle will penetrate in the course of slowing down to rest. This depth is measured along the initial direction of the particle. For high energy particles these data can be found at [103]. However, the effectiveness of passive encapsulation against invasive adversaries is quite limited.

Randomization of the Timing

This strategy can be useful if timing is crucial in a concrete fault analysis scenario. Implementations may include dummy random cycles or random delays of processes [14]. The objective is to randomly embed the relevant time interval dt within a larger time interval which leads to an enhancement of N. If the physical leakage in $\mathcal{C}$ cannot be analyzed in real-time, an adversary is not able to adapt to the randomized timing. Instead, the source of randomness in the circuit may become an attractive target. Similarly, variations of the clock frequency may help to increase N.

Error Detection Codes

Error detection codes of data items are well known for software implementations. For implementations in circuitry, [70] introduced parity based error detection for a substitution-permutation network based block cipher. In [85] error detection techniques based on multiple parity bits and non-linear codes are evaluated. Among them, r-bit non-linear codes are the most promising, but at the cost of an area overhead that is nearly comparable to duplication. As result, error detection codes lead to an enhancement of L which in turn typically increases N.

Physical Duplication

The objective is to duplicate critical target volumes of the circuit for parallel computation and a comparison of correctness [14]. In the context

of asynchronous circuits, [52] has proposed this idea to improve tamper resistance. These circuits make use of redundant data encoding, i.e., each bit is encoded onto two wires. Such dual-rail coding offers the opportunity to create an alarm signal that can be used for error detection by the physical device. For memory cells, a 'dual flip-flop dual-rail' design is proposed. The main idea is that an error state at any gate input always propagates to the gate output. In case of area duplication, the number of locations for fault injections is typically doubled, i.e., L is enhanced and precise control over the fault injection process is needed to prevent an error detection.

Sequential Computation

Sequential computation processes the same operation twice, i.e., the same part of the circuit is re-used and the results are compared [14]. However, this method is not reliable if a permanent fault is present in the circuit. In case of transient errors, repeating leads to an enhancement of M.

Active Sensor Networks

The idea of active sensor networks is proposed in [52]. The authors suggest to include small optical tamper sensors within each standard cell. These sensors consist of one or two transistors and enforce an error state if illuminated. Such a network of short-distance sensors can be spanned at critical parts of the circuit. The mean distance between sensors then constitutes an upper bound to the area $\Delta A(z)$ at which fault injection may not be detected by the sensors. An adversary has to precisely focus only on the target volume dV which establishes a hard problem, for $\mathcal{A}_{non-inv}$ and $\mathcal{A}_{semi-inv}$. Alarm detection may be deployed at an active encapsulation within the cryptographic boundary of the device. Again, this encapsulation should be out of the reach of $\mathcal{A}_{non-inv}$ and $\mathcal{A}_{semi-inv}$.

Chapter 8

Conclusion and Open Problems

8.1 Conclusion

This thesis provides models and algorithms for enhancing physical cryptanalysis. The models of this thesis provide a better understanding and analysis of the nature of physical leakage in integrated circuits, and more efficient tools and algorithms for cryptanalysis. The developed algorithms are useful in order to find critical points of a cryptographic implementation. Improving models and algorithms for physical cryptanalysis thereby helps in developing efficient countermeasures. A summary on the main research contributions is given in Section 1.2.

The best currently known approach in physical cryptanalysis is a thorough experimental verification at a profiling stage, which is included in methods achieving maximum power. The final multivariate algorithms of Chapter 6 can be seen as the most efficient ones in side channel cryptanalysis. T-Test based templates are the method of choice if a high number of measurements is available for profiling. However, in case of a low number of measurements, stochastic methods are an alternative and can reach superior efficiency both in terms of profiling and classification. Moreover, stochastic methods remain applicable even if the number of measurements at profiling is less than the number of subkey dependencies, which may be important for block cipher designs with an enlarged bit size of subkeys and in the presence of masking. While profiling may be expensive in terms of measurements, a lower bound of measurements needed for key recovery can easily be found by applying the maximum likelihood principle to small sets of additional measurements.

For securing cryptographic implementations on physical cryptanalysis, primarily physical leakage must be reduced and fault tolerance in integrated circuits should be improved. Because of that, countermeasures should be planned at the development stage of integrated circuits. Combining new logic styles like SABL [140, 141, 87] with randomized timing [41] and algorithmic randomization constitutes the most promising direction for hiding internal states and thereby securing cryptographic implementations.

8.2 Open Problems

Templates and stochastic methods still require strong assumptions on the capabilities of an adversary in a profiling stage. Especially, for masked implementations it is of interest to develop methods for limited adversaries that, e.g., do not have access to masking values at profiling. A first contribution into this area is the use of Gaussian mixture models that will be published soon [L17].

While many advances in side channel cryptanalysis have been achieved in the last years, the optimization of equipment for electromagnetic emanation has not progressed accordingly. Applications of special interest are pervasive RFID tags where the clock signal dominates and interferes the measurements [37].

Chapter 7 provides a new physical model for circuit based security bounds against tampering. However, a proof-of-concept at a concrete integrated circuit is still missing. Fault channel cryptanalysis deserves definitively more detailed research efforts. For ongoing research an IC with a known layout and an optical fault injection set-up is needed. It would be beneficial if laboratory equipment for optical fault injection would also be made available to research groups at universities. Advances in fault channel cryptanalysis are expected to be forthcoming soon. It appears reasonable that there will be experimental evidence for second-order and multivariate analysis. Open problems are, for instance, how precise faults can be injected and what the success rates for inducing multiple precise faults will be.

Another valuable direction for further research is to develop stochastic methods for fault channel techniques. Here, the stochastic process for each internal state is seen as a physically enforced transition to any new internal state. As internal states and transition rates between internal states are specific for each implementation, a good starting point would be again an IC with a known layout. Related work dealing with a graph-theoretical approach may also be favorable.

The development of efficient and reliable countermeasures against fault analysis deserves more investigations. Critical points of fault injection into an integrated circuit seem to be not very well understood, yet, and thorough investigations are certainly necessary. A forward-looking vision is the integration of fault tolerance into the design flow for integrated circuits. Also, circuit based tests of SABL logic styles regarding

their resistance against fault injection would be good to have.

It is still an interesting challenge whether physical leakage can formally be tied to security notions in a realistic model for physical cryptanalysis. The work of [97, 54, 64, 139] may provide a starting point for ongoing research.

The most promising future vision, however, is the reduction of experimental testing efforts for physical cryptanalysis. This may be achieved by the development of advanced design flow tools that, even more visionary, could guarantee suitable assumptions about the physical leakage of integrated circuits, maybe even in the context of provable security.

Bibliography

[1] The Side Channel Cryptanalysis Lounge. Available from: `http://www.crypto.rub.de/en_sclounge.html`.

[2] Wikipedia, The Free Encyclopedia. Available from: `http://en.wikipedia.org/wiki/Main_Page`.

[3] Onur Aciiçmez, Jean-Pierre Seifert, and Çetin Kaya Koç. Predicting Secret Keys via Branch Prediction. Cryptology ePrint Archive, Report 2006/288, 2006. Available from: `http://eprint.iacr.org/2006/288.pdf`.

[4] Dakshi Agrawal, Josyula R. Rao, and Pankaj Rohatgi. Multi-channel Attacks. In C. Walter, Ç. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003*, volume 2779 of *LNCS*, pages 2–16. Springer-Verlag, 2003.

[5] Dakshi Agrawal, Josyula R. Rao, Pankaj Rohatgi, and Kai Schramm. Templates as Master Keys. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005*, volume 3659 of *LNCS*, pages 15–29. Springer, 2005.

[6] Manfred Aigner and Elisabeth Oswald. Power Analysis Tutorial. Available from: `http://www.iaik.tugraz.at/aboutus/people/oswald/papers/dpa_tutorial.pdf`.

[7] Marcelo Alonso and Edward J. Finn. *Physik*. Addison-Wesley, 1988.

[8] Ross Anderson. *Security Engineering*. John Wiley & Sons, Inc., 2001.

[9] Ross Anderson and Markus Kuhn. Tamper Resistance — A Cautionary Note. In *The Second USENIX Workshop on Electronic Commerce Proocedings*, pages 1–11, 1996. Available from: `http://www.cl.cam.ac.uk/~mgk25/tamper.pdf`.

[10] Ross J. Anderson and Markus G. Kuhn. Low Cost Attacks on Tamper Resistant Devices. In Bruce Christianson, Bruno Crispo, T. Mark A. Lomas, and Michael Roe, editors, *Security Protocols Workshop*, volume 1361 of *LNCS*, pages 125–136. Springer, 1997.

[11] C. Archambeau, E. Peeters, F.-X. Standaert, and J.-J. Quisquater. Template Attacks in Principal Subspaces. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *LNCS*, pages 1–14. Springer, 2006.

[12] Christian Aumüller, Peter Bier, Wieland Fischer, Peter Hofreiter, and Jean-Pierre Seifert. Fault Attacks on RSA with CRT: Concrete Results and Practical Countermeasures. In B. S. Kaliski, Ç Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *LNCS*, pages 260–275. Springer-Verlag, 2003.

[13] D. Babbage, D. Catalano, C. Cid, C. Gehrmann (editor), L. Granboulan, T. Lange, A. Lenstra, M. Näslund (editor), P. Nguyen, C. Paar, J. Pelzl, T. Pornin, B. Preneel, M. Robshaw, A. Rupp, N. Smart, and M. Ward. D.SPA.16, ECRYPT Yearly Report on Algorithms and Keysizes (2005), 2006. Available from: `http://www.ecrypt.eu.org/documents/D.SPA.16-1.0.pdf`.

[14] Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstall, and Claire Whelan. The Sorcerer's Apprenctice's Guide to Fault Attacks. Technical report, 2004. Available from: `http://eprint.iacr.org/2004/100`.

[15] Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstall, and Claire Whelan. The Sorcerer's Apprentice Guide to Fault Attacks. *Proceedings of the IEEE*, 94(2):370–382, 2006.

[16] L. Batina, E. De Mulder, K. Lemke, E. Oswald, G. Piret, and F.-X. Standaert (editor). ECRYT D.VAM.4: Electromagnetic Analysis and Fault Attacks: State of the Art. Technical report, 2005. Available from: `http://www.ecrypt.eu.org/documents/D.VAM.4-3.pdf`.

[17] Raphael Bauduin. Fault attacks, an intuitive approach. Invited Talk at FDTC 2006.

[18] Friedrich Beck. *Integrated Circuit Failure Analysis: A Guide to Preparation Techniques*. John Wiley & Sons, 1998.

[19] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying Hash Functions for Message Authentication. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96*, volume 1109 of *LNCS*, pages 1–15, 1996.

[20] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Message Authentication using Hash Functions – The HMAC Construction. Technical report, RSA Laboratories' CryptoBytes, Vol. 2, No. 1, 1996. Available from: `www-cse.ucsd.edu/~mihir/papers/hmac-cb.ps`.

[21] Daniel J. Bernstein. Cache-timing attacks on AES, 2005. Available from: `http://cr.yp.to/antiforgery/cachetiming-20050414.pdf`.

[22] Eli Biham, Louis Granboulan, and Phong Q. Nguyen. Impossible Fault Analysis of RC4 and Differential Fault Analysis of RC4. In Henri Gilbert and Helena Handschuh, editors, *Fast Software Encryption: 12th International Workshop*, volume 3557 of *LNCS*, pages 359–367. Springer, 2005.

[23] Eli Biham and Adi Shamir. The Next Stage of Differential Fault Analysis: How to break completely unknown cryptosystems, 1996. Available from: `http://jya.com/dfa.htm`.

[24] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In Burton S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO '97*, volume 1294 of *LNCS*, pages 513–525. Springer, 1997.

[25] Eli Biham and Adi Shamir. Power Analysis of the Key Scheduling of the AES Candidates. In *Proceedings of the Second Advanced Encryption Standard (AES) Candidate Conference*, Rome, Italy, 1999. Available from: `http://csrc.nist.gov/CryptoToolkit/aes/round1/conf2/papers/biham3.pdf`.

[26] Johannes Blömer and Volker Krummel. Fault Based Collision Attacks on AES. In *Fault Diagnosis and Tolerance in Cryptography*, volume 4236 of *LNCS*, pages 106–120. Springer, 2006.

[27] Johannes Blömer, Martin Otto, and Jean-Pierre Seifert. A new CRT-RSA algoritm secure against Bellcore attacks. In *Conference on Computer and Communications Security – CCS 2003*, pages 311–320. ACM SIGSAC, ACM Press, 2003.

[28] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults (Extended Abstract). In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97*, volume 1233 of *LNCS*, pages 37–51. Springer, 1997.

[29] Eric Brier, Benoît Chevallier-Mames, Mathieu Ciet, and Christophe Clavier. Why One Should Also Secure RSA Public Key Elements. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *LNCS*, pages 324–338. Springer, 2006.

[30] Eric Brier, Christophe Clavier, and Francis Olivier. Optimal Statistical Power Analysis. Cryptology ePrint Archive, 2003. Available from: `http://eprint.iacr.org/2003/152.pdf`.

[31] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *LNCS*, pages 16–29. Springer-Verlag, 2004.

[32] I. N. Bronstein and K. A. Semendjajew. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 1989.

[33] David Brumley and Dan Boneh. Remote Timing Attacks are Practical. In *Proceedings of the 12th USENIX Security Symposium*, August 2003. Available from: `http://www.cs.cmu.edu/~dbrumley/pubs/openssltiming.pdf`.

[34] Matthias Brüstle. SOSSE: Simple Operating System for Smartcard Education. Available from: `http://www.mbsks.franken.de/sosse`.

[35] BSI, CESG, DCSSI, and NLNCSA. Application for Attack Potential to Smartcards, Version 1.1, July 2002. Available from: `http://www.commoncriteriaportal.org/public/files/2002-08-001.pdf`.

[36] Dario Carluccio. Electromagnetic Side Channel Analysis for Embedded Crypto Devices. Master's thesis, Ruhr-Universität Bochum, 2005.

[37] Dario Carluccio, Kerstin Lemke, and Christof Paar. Electromagnetic Side Channel Analysis of a Contactless Smart Card: First Results. In *ECRYPT Workshop on RFID and Lightweight Crypto, Graz, Austria*, pages 44–51, 2005. Available from: `http://www.iaik.tu-graz.ac.at/research/krypto/events/RFID-SlidesandProc%eedings/Proceedings-WSonRFIDandLWCrypto.zip`.

[38] Suresh Chari, Charanjit Jutla, Josyula R. Rao, and Pankaj Rohatgi. A Cautionary Note Regarding Evaluation of AES Candidates on Smart-Cards. In *Second Advanced Encryption Standard (AES) Candidate Conference*, Rome, Italy, 1999. Available from: `http://csrc.nist.gov/encryption/aes/round1/conf2/aes2conf.htm`.

[39] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankay Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In M. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *LNCS*, pages 398–412. Springer-Verlag, 1999.

[40] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template Attacks. In B. S. Kaliski, Ç Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems*, volume 2523 of *LNCS*, pages 13–28. Springer-Verlag, 2003.

[41] Christophe Clavier, Jean-Sébastian Coron, and Nora Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000*, volume 1965 of *LNCS*, pages 252–263. Springer, 2000.

[42] Jean-Sébastian Coron. Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 1999*, volume 1717 of *LNCS*, pages 292–302. Springer-Verlag, 1999.

[43] Jean-Sébastian Coron and Louis Goubin. On Boolean and Arithmetic Masking against Differential Power Analysis. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000*, volume 1965 of *LNCS*, pages 231–237. Springer-Verlag, 2000.

[44] Jean-Sébastian Coron, Paul Kocher, and David Naccache. Statistics and Secret Leakage. In Y. Frankel, editor, *Financial Cryptography (FC 2000)*, volume 1962 of *LNCS*, pages 157–173. Springer-Verlag, 2001.

[45] Jean-Sébastian Coron and Alexei Tchulkine. A New Algorithm for Switching from Arithmetic to Boolean Masking. In C. Walter, Ç. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003*, volume 2779 of *LNCS*, pages 89–97. Springer-Verlag, 2003.

[46] Bert den Boer, Kerstin Lemke, and Guntram Wicke. A DPA Attack against the Modular Reduction within a CRT Implementation of RSA. In B. S. Kaliski, Ç Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *LNCS*, pages 228–243. Springer-Verlag, 2003.

[47] J.-F. Dhem, F. Koene, P.-A. Leroux, P. Mestré, J.-J. Quisquater, and J.L. Willems. A practical implementation of the timing attack. UCL Crypto Group Technical Report Series CG-1998/1, Universiteé catholique de Louvain (UCL), 1998.

[48] Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. RIPEMD-160: A Strengthened Version of RIPEMD. In Dieter Gollmann, editor, *Fast Software Encryption, Third International Workshop*, volume 1039 of *LNCS*, pages 71–82, 1996. Available from: `http://homes.esat.kuleuven.be/~cosicart/pdf/AB-9601/AB-9601.pdf`.

[49] Paul. N. Fahn and Peter K. Pearson. IPA: A New Class of Power Attacks. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 1999*, volume 1717 of *LNCS*, pages 173–186. Springer-Verlag, 1999.

[50] K. Finkenzeller. *RFID-Handbuch*. Hanser Fachbuchverlag, Third edition, October 2002.

[51] Wieland Fischer and Berndt M. Gammel. Masking at Gate Level in the Presence of Glitches. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005*, volume 3659 of *LNCS*, pages 187–200. Springer, 2005.

[52] Jacques J. A. Fournier, Simon W. Moore, Huiyun Li, Robert D. Mullins, and George S. Taylor. Security Evaluation of Asynchronous Circuits. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003*, volume 2779 of *LNCS*, pages 137–151. Springer, 2003.

[53] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic Analysis: Concrete Results. In Ç Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2001*, volume 2162 of *LNCS*, pages 251–261. Springer-Verlag, 2001.

[54] Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic Tamper-Proof (ATP) Security: Theoretical Foundations for Security against Hardware Tampering. In Moni Naor, editor, *Theory of Cryptography*, volume 2951 of *LNCS*, pages 258–277. Springer, 2004. Available from: `http://www.cs.brown.edu/~anna/papers/glmmr04.pdf`.

[55] Benedikt Gierlichs. Signal Theoretical Methods in Differential Side Channel Cryptanalysis. Master's thesis, Ruhr-Universität Bochum, 2006.

[56] Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. Stochastic Methods. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *LNCS*, pages 15–29. Springer, 2006.

[57] Louis Goubin. A Sound Method for Switching between Boolean and Arithmetic Masking. In Ç Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2001*, volume 2162 of *LNCS*, pages 3–15. Springer-Verlag, 2001.

[58] Louis Goubin and Jacques Patarin. DES and Differential Power Analysis - The "Duplication" Method. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 1999*, volume 1717 of *LNCS*, pages 158–172. Springer-Verlag, 1999.

[59] Alfonso De Gregorio. Cryptographic Key Reliable Lifetimes: Bounding the Risk of Key Exposure in the Presence of Faults. In *Fault Diagnosis and Tolerance in Cryptography*, volume 4236 of *LNCS*, pages 144–158. Springer, 2006.

[60] Shay Gueron and Jean-Pierre Seifert. Is it Wise to Publish Your Public RSA Keys? In *Fault Diagnosis and Tolerance in Cryptography*, volume 4236 of *LNCS*, pages 1–12. Springer, 2006.

[61] Peter Gutmann. Secure Deletion of Data from Magnetic and Solid-State Memory. In *Sixth USENIX Security Symposium Proceedings, San Jose, California, July 22-25, 1996*, pages 77–90, 1996. Available from: `www.usenix.org/publications/library/proceedings/sec96/full_papers/gutma%nn/`.

[62] Jonathan J. Hoch and Adi Shamir. Fault analysis of stream ciphers. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *LNCS*, pages 240–253. Springer, 2004.

[63] Atmel Smart Card ICs, Hitachi Europe Ltd., Infineon Technologies AG, and Philips Semiconductors. Smartcard IC Platform Protection Profile, 1.0. Available from: `www.bsi.bund.de/cc/pplist/ssvgpp01.pdf`.

[64] Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private circuits ii: Keeping secrets in tamperable circuits. In *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 308–327. Springer, 2006.

[65] ISO. ISO 13491-1, Banking – Secure cryptographic devices (retail) – Part 1: Concepts, requirements and evaluation methods, First edition 1998-06-15.

[66] JEDEC. JEDEC STANDARD, Measurement and Reporting of Alpha Particles and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices, 2001. Available from: `http://www.jedec.org/download/search/JESD89.pdf`.

[67] Marc Joye, Pascal Paillier, and Berry Schoenmakers. On Second-Order Differential Power Analysis. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005*, volume 3659 of *LNCS*, pages 293–308. Springer, 2005.

[68] Marc Joye, Jean-Jacques Quisquater, Sung-Ming Yen, and Moti Yung. Observability analysis - detecting when improved cryptosystems fail. In Bart Preneel, editor, *CT-RSA*, volume 2271 of *LNCS*, pages 17–29. Springer, 2002.

[69] Pascal Junod. *Statistical Cryptanalysis of Block Ciphers*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2004.

[70] Ramesh Karri, Grigori Kuznetsov, and Michael Goessel. Parity-Based Concurrent Error Detection of Substitution-Permutation Network Block Ciphers. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003*, volume 2779 of *LNCS*, pages 113–124. Springer, 2003.

[71] M. G. Kendall and A. Stuart. *The Advanced Theory of Statistics I, II*. Macmillan, Inc., New York, 1977.

[72] Paul C. Kocher. Timing Attacks on Implementations of Diffie–Hellman, RSA, DSS, and Other Systems. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO '96*, volume 1109 of *LNCS*, pages 104–113. Springer-Verlag, 1996.

[73] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In M. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *LNCS*, pages 388–397. Springer-Verlag, 1999.

[74] Oliver Kömmerling and Markus G. Kuhn. Design Principles for Tamper-Resistant Smartcard Processors. In *Proceedings of the USENIX Workshop on Smartcard Technology (Smartcard '99)*, pages 9–20, 1999. Available from: `http://www.cl.cam.ac.uk/~mgk25/sc99-tamper.pdf`.

[75] Markus G. Kuhn. Compromising emanations: eavesdropping risks of computer displays. UCAM-CL-TR 577, University of Cambridge, Computer Laboratory, 2003. Available from: `http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-577.pdf`.

[76] Xuejia Lai and James L. Massey. Markov Ciphers and Differential Cryptoanalysis. In Donald W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91*, volume 547 of *LNCS*, pages 17–38, 1991.

[77] Hervé Ledig, Frédéric Muller, and Frédéric Valette. Enhancing Collision Attacks. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *LNCS*, pages 176–190. Springer-Verlag, 2004.

[78] E. L. Lehmann. *Testing Statistical Hypotheses*. Springer, 1986.

[79] Kerstin Lemke and Christof Paar. An Adversarial Model for Fault Analysis against Low-Cost Cryptographic Devices. In *Workshop on Fault Detection and Tolerance in Cryptography*, pages 82–94, 2005.

[80] Kerstin Lemke, Christof Paar, and Ahmad-Reza Sadeghi. Physical Security Bounds Against Tampering. In *Applied Cryptography and Network Security*, volume 3989 of *LNCS*, pages 253–267. Springer, 2006.

[81] Kerstin Lemke, Kai Schramm, and Christof Paar. DPA on n-Bit Sized Boolean and Arithmetic Operations and Its Application to IDEA, RC6, and the HMAC-Construction. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *LNCS*, pages 205–219. Springer-Verlag, 2004.

[82] Kerstin Lemke-Rust and Christof Paar. An Adversarial Model for Fault Analysis against Low-Cost Cryptographic Devices. In *Fault Diagnosis and Tolerance in Cryptography*, volume 4236 of *LNCS*, pages 131–143. Springer, 2006.

[83] Regís Leveugle. Early Analysis of Fault Attack Effects for Cryptographic Hardware. In *Workshop on Fault Detection and Tolerance in Cryptography*, 2004.

[84] P.-Y. Liardet and Y. Teglia. From Reliability to Safety. In *Workshop on Fault Detection and Tolerance in Cryptography*, 2004.

[85] Tal G. Malkin, François-Xavier Standaert, and Moti Yung. A Comparative Cost/Security Analysis of Fault Attack Countermeasures. In *Workshop on Fault Detection and Tolerance in Cryptography*, pages 109–123, 2005.

[86] Stefan Mangard. Hardware Countermeasures against DPA - A Statistical Analysis of Their Effectiveness. In Tatsuaki Okamoto, editor, *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004*, volume 2964 of *LNCS*, pages 222–235. Springer-Verlag, 2004.

[87] Stefan Mangard. *Securing Implementations of Block Ciphers against Side-Channel Attacks*. PhD thesis, Institute for Applied Information Processing and Communications (IAIK), Graz University of Technology, Graz, Austria, 2004.

[88] Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-Channel Leakage of Masked CMOS Gates. In *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005*, volume 3376 of *LNCS*, pages 351–365. Springer, 2005.

[89] Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully Attacking Masked AES Hardware Implementations. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005*, volume 3659 of *LNCS*, pages 157–171. Springer, 2005.

[90] Rita Mayer-Sommer. Smartly Analyzing the Simplicity and the Power of Simple Power Analysis on Smartcards. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000*, volume 1965 of *LNCS*, pages 78–92. Springer-Verlag, 2000.

[91] Joel McNamara. The Complete, Unofficial TEMPEST Information Page. Available from: `http://www.eskimo.com/~joelm/tempest.html`.

[92] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[93] Thomas S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000*, volume 1965 of *LNCS*, pages 238–251. Springer-Verlag, 2000.

[94] Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *Proceedings of USENIX Workshop on Smartcard Technology*, pages 151–162, 1999. Available from: `http://www.usenix.org/events/smartcard99/full_papers/messerges/messerge%s.pdf`.

[95] Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Power Analysis Attacks of Modular Exponentiation in Smartcards. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 1999*, volume 1717 of *LNCS*, pages 144–157. Springer-Verlag, 1999.

[96] Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers*, 51(5):541–552, May 2002.

[97] Silvio Micali and Leonid Reyzin. Physically Observable Cryptography. Cryptology ePrint Archive, 2003. Available from: `http://eprint.iacr.org/2003/120.pdf`.

[98] David Naccache. Finding Faults. *IEEE Security & Privacy*, 3(5):61–65, 2005.

[99] David Naccache, Phong Q. Nguyen, Michael Tunstall, and Claire Whelan. Experimenting with Faults, Lattices and the DSA. In Serge Vaudenay, editor, *Public Key Cryptography*, volume 3386 of *LNCS*, pages 16–28. Springer, 2005.

[100] Michael Neve, Eric Peeters, David Samyde, and Jean-Jacques Quisquater. Memories: a Survey of their Secure Uses in Smart Cards. Technical report. Available from: `http://www.dice.ucl.ac.be/~mneve/document/Publications/sisw03.pdf`.

[101] U.S. Department of Commerce / National Institue of Standards and Technology. DATA ENCRYPTION STANDARD (DES). FIPS PUB 46-3, 1999. Available from: `http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf`.

[102] National Institue of Standards and Technology. Announcing the ADVANCED ENCRYPTION STANDARD (AES). FIPS PUB 197, 2001. Available from: `http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`.

[103] National Institute of Standards and Technology (NIST). Physical Reference Data, available at http://physics.nist.gov/PhysRefData/contents.html.

[104] National Institute of Standards and Technology. Security Requirements for Cryptographic Modules, 1994. Available from: `csrc.nist.gov/cryptval`.

[105] National Institute of Standards and Technology. FIPS PUB 140-2, Security Requirements for Cryptographic Modules, 2002. Available from: `csrc.nist.gov/cryptval`.

[106] National Institute of Standards and Technology. Secure Hash Standard, 2002. Available from: `csrc.nist.gov/cryptval`.

[107] Elisabeth Oswald. *On Side-Channel Attacks and the Application of Algorithmic Countermeasures*. PhD thesis, Institute for Applied Information Processing and Communications (IAIK), Graz University of Technology, Graz, Austria, 2003.

[108] Elisabeth Oswald and Stefan Mangard. Template Attacks on Masking – Resistance is Futile. In Masayuki Abe, editor, *Topics in Cryptology – CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007*, volume 4377 of *LNCS*, pages 243–256. Springer, 2006.

[109] Elisabeth Oswald, Stefan Mangard, Christoph Herbst, and Stefan Tillich. Practical Second-Order DPA Attacks for Masked Smart Card Implementations of Block Ciphers. In David Pointcheval, editor, *Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006*, volume 3860 of *LNCS*, pages 192–207. Springer, 2006.

[110] Elisabeth Oswald and Bart Preneel. A Theoretical Evaluation of some NESSIE Candidates regarding their Susceptibility towards Power Analysis Attacks. Technical report, Katholieke Universiteit Leuven, Dept. ESAT, B-3001 Leuven-Heverlee, Belgium, 2002. Available from: `http://www.iaik.tugraz.at/aboutus/people/oswald/papers/NESSIE_BC_main.p%df`.

[111] D. Page. Theoretical Use of Cache Memory as a Cryptanalytic Side-Channel. Cryptology ePrint Archive, 2002. Available from: `http://eprint.iacr.org/2002/169.pdf`.

[112] Eric Peeters, François-Xavier Standaert, Nicolas Donckers, and Jean-Jacques Quisquater. Improved Higher-Order Side-Channel Attacks with FPGA Experiments. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware*

and Embedded Systems - CHES 2005, volume 3659 of *LNCS*, pages 309–323. Springer, 2005.

[113] Wiebe R. Pestman. *Mathematical Statistics*. Walter de Gruyter, 1998.

[114] Gilles Piret and Jean-Jacques Quisquater. A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003*, volume 2779 of *LNCS*, pages 77–88. Springer, 2003.

[115] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1995.

[116] Jean-Jacques Quisquater and David Samyde. ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smard Cards. In I. Attali and T. Jensen, editors, *Proceedings of Smart Card Programming and Security (E-smart 2001)*, volume 2140 of *LNCS*, pages 200–210. Springer-Verlag, 2001.

[117] Wolfgang Rankl and Wolfgang Effing. *Handbuch der Chipkarten*. Carl Hanser Verlag, 2002.

[118] Christian Rechberger. Side Channel Analysis of Stream Ciphers. Master's thesis, Institute for Applied Information Processing and Communications (IAIK), Graz, Austria, 2004. Available from: `http://www.iaik.tugraz.at/aboutus/people/rechberger/written/mthesis_SCA%_on_Stream_Ciphers.pdf`.

[119] Horst Rinne. *Taschenbuch der Statistik*. Verlag Harri Deutsch, 2003.

[120] Ronald L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L. Yin. The RC6 Block Cipher. Version 1.1, August 20, 1998. Available from: `http://theory.lcs.mit.edu/~rivest/rc6.pdf`.

[121] David Samyde and Jean-Jacques Quisquater. Eddy Current for Magnetic Analysis with Active Sensor. In *Proceedings of ESmart 2002*, pages 185–194, 2002.

[122] David Samyde, Sergei Skorobogatov, Ross Anderson, and Jean-Jacques Quisquater. On a New Way to Read Data from Memory. Technical report. Available from: `http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/SISW02.pdf`.

[123] Werner Schindler. A Timing Attack against RSA with the Chinese Remainder Theorem. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000*, volume 1965 of *LNCS*, pages 109–124. Springer-Verlag, 2000.

[124] Werner Schindler. On the Optimization of Side-Channel Attacks by Advanced Stochastic Methods. In Serge Vaudenay, editor, *Public Key Cryptography*, volume 3386 of *LNCS*, pages 85–103. Springer, 2005.

[125] Werner Schindler, François Koene, and Jean-Jacques Quisquater. Unleashing the full power of timing attack. UCL Crypto Group Technical Report Series CG-2001/3, Universiteé catholique de Louvain (UCL), 2001.

[126] Werner Schindler, Kerstin Lemke, and Christof Paar. A Stochastic Model for Differential Side Channel Cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005*, volume 3659 of *LNCS*, pages 30–46. Springer, 2005.

[127] Bruce Schneier. *Applied Cryptography*. John Wiley & Sons, Inc., 1998.

[128] Bruce Schneier. *Beyond Fear*. Copernicus Books, 2003.

[129] Kai Schramm. *Advanced Methods in Side Channel Cryptanalysis*. PhD thesis, Department for Electrical Engineering and Information Technology, Ruhr-Universität Bochum, Germany, 2006.

[130] Kai Schramm, Gregor Leander, Patrick Felke, and Christof Paar. A Collision-Attack on AES Combining Side Channel- and Differential-Attack. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *LNCS*, pages 163–175. Springer-Verlag, 2004.

[131] Kai Schramm and Christof Paar. Higher Order Masking of the AES. In David Pointcheval, editor, *Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006*, volume 3860 of *LNCS*, pages 208–225. Springer, 2006.

[132] Kai Schramm, Thomas Wollinger, and Christof Paar. A New Class of Collision Attacks and Its Application to DES. In Thomas Johansson, editor, *Fast Software Encryption, 10th International Workshop, FSE 2003*, volume 2887 of *LNCS*, pages 206–222. Springer-Verlag, 2003.

[133] Motorola Semiconductors, Philips Semiconductors, Siemens Semiconductors, STMicroelectronics, and Texas-Instruments Semiconductors. PP/9806: Smartcard Integrated Circuit Protection Profile v2.0. Available from: `www.ssi.gouv.fr/site_documents/PP/PP9806.pdf`.

[134] Adi Shamir. Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000*, volume 1965 of *LNCS*, pages 71–77. Springer-Verlag, 2000.

[135] C. E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27:379–423, 1948. Available from: `http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf`.

[136] Sergei Skorobogatov. Low temperature data remanence in static RAM. Technical Report UCAM-CL-TR-536, 2002. Available from: `http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-536.pdf`.

[137] Sergei P. Skorobogatov and Ross J. Anderson. Optical Fault Induction Attacks. In B. S. Kaliski, Ç Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *LNCS*, pages 2–12. Springer-Verlag, 2003.

[138] Sergei S. Skorobogatov. Semi-invasive attacks — A new approach to hardware security analysis, available at http://www.cl.cam.ac.uk/techreports/ucam-cl-tr-630.pdf. Technical report, 2005. Available from: `http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-630.pdf`.

[139] Francois-Xavier Standaert, Tal G. Malkin, and Moti Yung. A Formal Practice-Oriented Model For The Analysis of Side-Channel Attacks, Version 1.5, January 3, 2007. Cryptology ePrint Archive. Available from: `http://eprint.iacr.org/2006/139.pdf`.

[140] Kris Tiri and Ingrid Verbauwhede. Securing Encryption Algorithms against DPA at the Logic Level: Next Generation Smart Card Technology. In C. Walter, Ç. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003*, volume 2779 of *LNCS*, pages 125–136. Springer-Verlag, 2003.

[141] Kris Tiri and Ingrid Verbauwhede. Charge Recycling Sense Amplifier Based Logic: Securing Low Power Security IC's against Differential Power Analysis. Cryptology ePrint Archive, 2004. Available from: `http://eprint.iacr.org/2004/067.pdf`.

[142] Pim Tuyls, Geert-Jan Schrijen, Boris Škorić, Jan van Geloven, Nynke Verhaegh, and Rob Wolters. Read-Proof Hardware from Protective Coatings. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop*, volume 4249 of *LNCS*, pages 369–383. Springer, 2006.

[143] Jason Waddle and David Wagner. Towards Efficient Second-Order Power Analysis. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *LNCS*, pages 1–15. Springer-Verlag, 2004.

[144] Steve H. Weingart. Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000*, volume 1965 of *LNCS*, pages 302–317. Springer, 2000.

[145] Neil H. E. Weste and Kamran Eshraghian. *Principles of CMOS VLSI Design*. Addison-Wesley, 1992.

[146] Andreas Wiemers. Partial collision search by side channel analysis. Workshop "Smartcards and Side Channel Attacks", Horst Görtz Institute for IT-Security, Ruhr-University Bochum, Germany, 30-31 January 2003, 2003. Available from: `http://ca.itsc.ruhr-uni-bochum.de/hgi/smaca/wiemers.pdf`.

List of Figures

2.1 CMOS inverter 27

3.1 Cryptographic boundary of a computer system. 31
3.2 Passive Implementation Attacks 40
3.3 Active Implementation Attacks 40
3.4 Measurement set-up for a timing attack at a network connection 48
3.5 Measurement set-up for both power consumption and EM emanation 52
3.6 Power consumption measurement trace of an AES implementation 53
3.7 Zoom into Figure 3.6 53
3.8 Last round of DES and DES inner function f in the last round 58
3.9 DSCA results of an AES implementation with the correlation method. 63
3.10 Measurement set-up for a glitch attack and an active probe 73

4.1 Correlation coefficient versus all key hypotheses for an XOR operation by applying Algorithm 3.5 91
4.2 Correlation coefficient versus all key hypotheses for an addition modulo 2^n 92
4.3 Correlation coefficient versus all key hypotheses for the IDEA multiplication (16-bit Hamming weight) 94
4.4 Correlation coefficient versus all key hypotheses for the IDEA multiplication (8-bit Hamming weight) 95

4.5 Experimental results for addition modulo 2^{16} at an 8051 microcontroller . . . 106
4.6 Experimental results for addition modulo 2^{16} at an AVR microcontroller . . . 108
4.7 Experimental results using the XOR selection function at an AVR microcontroller . . . 109
4.8 Experimental results for the IDEA multiplication at an AVR microcontroller . . . 110

5.1 Intermediate result of the AES used for DSCA in the first round . . . 141
5.2 Bit-wise coefficients β^*_{j3} and β^*_{j4} . . . 142
5.3 Bit-wise coefficients β^*_{j7} and β^*_{j8} . . . 142
5.4 Coefficient β^*_{j8} for all four measurement series as a function of time. . . . 143
5.5 Squared Euclidean norm $\|\vec{b}\|^2$ for three different subkeys as result of profiling. . . . 144
5.6 Squared Euclidean norm $\|\vec{b}\|^2$ of coefficients in $\mathcal{F}_9$ . . . 146
5.7 Squared Euclidean norm $\|\vec{b}\|^2$ of coefficients and empirical variance in $\mathcal{F}_9$ . . . 148
5.8 Squared sum of pairwise $t-$differences in $\mathcal{F}_9$ . . . 148
5.9 Squared Euclidean norm of coefficients $\|\vec{b}\|^2$ and empirical variance in $\mathcal{F}_2$ and $\mathcal{F}_{16}$ . . . 152
5.10 Process of boolean masking . . . 156
5.11 Squared Euclidean norm of bit-wise coefficients in two vector subspaces . . . 158
5.12 Profiling results do not fulfill EISM at all instants. . . . 159

6.1 Metric 1 for device A for original attacks. . . . 176
6.2 Metric 3 for device A for original attacks. . . . 177
6.3 Selection of instants with sosd and sost. . . . 180
6.4 Metric 3 for device A for the optimized template attack. . 183
6.5 Metric 3 for device A for the optimized stochastic methods. 184
6.6 Overall comparison: metric 3 for device A. . . . 185

7.1 Information flow at a fault analysis set-up. . . . 191
7.2 Tampering attack against a Digital Signature Scheme . . 193
7.3 Impact of the particle beam into the circuit . . . 196

List of Tables

2.1 T-Test of two populations $N(\mu_X, \sigma^2)$ and $N(\mu_Y, \sigma^2)$ with unknown σ. The abbreviation $\Delta_\mu := \mu_Y - \mu_X$ is used below. 16
2.2 F-Test of two populations $N(\mu_x, \sigma_x^2)$ and $N(\mu_y, \sigma_y^2)$. . . . 18

3.1 Possible states of bit r and $r \oplus b$ and the mean leakage portion 67

4.1 DSCA result of bit $b = (x \oplus k^\circ)_j$ 90

5.1 Tasks for the minimum principle and the maximum likelihood principle. 122
5.2 Ranking list for determining the key at profiling. 144
5.3 Success rate for applying the minimum principle 151
5.4 Success rate for minimum principle in different vector subspaces 152
5.5 Success rate when comparing with the DSCA correlation method 154
5.6 Success rate for the maximum likelihood principle 155
5.7 Success rate on different series with variable-key profiling 160
5.8 Success rate on different series with fixed-key profiling . . 160
5.9 Success rate with knowledge of masking values 161
5.10 Success rate for the minimum principle 161
5.11 Success rate for second order DSCA. 163

6.1 Fundamental differences between templates and stochastic methods. 172

6.2 Concrete parameter values to study 176
6.3 Metric 2 for device A as function of N (Original Attacks). 177
6.4 Metric 2 for device A as function of N (Improvements for Templates). 182
6.5 Metric 2 for device A as function of N (Improvements for Stochastic Method). 183
6.6 Metric 3 for device B as function of N 185

7.1 Physical means according to the interaction range of an adversary 195
7.2 Passive and Active Defense Strategies 200

List of Algorithms

3.1 Square and multiply algorithm 47
3.2 A generic (single-bit) DSCA algorithm 61
3.3 A simple indexing algorithm (sorting for maximum absolute scalars) 62
3.4 Vectorial T-Test 62
3.5 (Multi-bit) DSCA algorithm using the correlation method 64
3.6 Pre-processing for known-offset 2DSCA 68
3.7 A generic Profiling Stage of the Template attack 70
3.8 Key Recovery at the Template attack 71
3.9 RSA-CRT algorithm 78
4.1 IDEA Encryption 96
4.2 IDEA Key Schedule (Encryption) 97
4.3 RC6-$w/r/b$ Encryption 98
4.4 RC6-$w/r/b$ Key Schedule 98
4.5 RIPEMD-160 100
4.6 SHA-1 103
5.1 Estimation of the deterministic part 124
5.2 Estimation of the multivariate Noise 128
5.3 Minimum Principle 130
5.4 Maximum Likelihood Principle 132
5.5 Minimum Principle in the presence of masking 136
5.6 Maximum Likelihood Principle in the presence of masking 138
5.7 Instant selection based on the squared Euclidean norm. . 145
5.8 Instant selection based on the squared Euclidean norm and empirical variance. 146
5.9 Instant selection based on the sum of squared pairwise t-differences. 147

6.1 Instant selection based on the sum of squared differences (sosd). 170
6.2 Instant selection based on sum of the squared pairwise t-differences (sost). 179

List of Abbreviations

2DSCA	Second-order DSCA
AES	Advanced Encryption Standard
ASIC	Application Specific IC
ATP	Algorithmic Tamper Proof
CMOS	Complementary Metal Oxide Silicon
CPA	Correlation Power Analysis
CRT	Chinese Remainder Theorem
DEMA	Differential Electromagnetic Analysis
DES	Data Encryption Standard
DPA	Differential Power Analysis
DSA	Digital Signature Algorithm
DSCA	Differential Side Channel Analysis
ECC	Elliptic Curve Cryptosystem
EEPROM	Electronically Erasable Programmable Read Only Memory
EM	electromagnetic
FIB	Focused Ion Beam
FIPS	Federal Information Processing Standards
FPGA	Field Programmable Gate Array
IC	Integrated Circuit
IPA	Inferential Power Analysis
ISO	International Organization for Standardization
LFSR	Linear Feedback Shift Register
MESD	Multiple Exponent, Single Data
MRED	Modular Reduction on Equidistant Data

NIST	National Institute of Standards and Technology
PIN	Personal Identification Number
RAM	Random Access Memory
RFID	Radio Frequency Identification
RNG	Random Number Generator
RSA	Rivest Shamir Adleman
RTC	Real Time Clock
SABL	Sense Amplifier Based Logic
SCA	Side Channel Analysis
SEMA	Simple Electromagnetic Analysis
SEMD	Single Exponent, Multiple Data
SFA	Simple Fault Analysis
sosd	sum of squared (pairwise) differences
sost	sum of squared (pairwise) t-differences
SPA	Simple Power Analysis
SR	Success Rate
SRAM	static RAM
SSCA	Simple Side Channel Analysis
VLSI	Very Large Scale Integration
XOR	exclusive-or
ZEMD	Zero Exponent, Multiple Data

$\exp(x)$	exponential function e^x
$\lg x$	Logarithm of x to the base 2
$\ln x$	Logarithm of x to the base e (Natural Logarithm)

Curriculum Vitae

Kerstin Lemke-Rust

I was born in Hannover, Germany, in 1970. I grew up in this area and finished school with the final examination 'Abitur' at the Kaiser Wilhelm Gymnasium in Hannover in 1989. Afterwards I studied physics at the University of Hannover. My main research interests were dedicated to plasma physics, biophysics, and computer science. I wrote my diploma thesis on a low-pressure Penning glow discharge at the Institute of Plasma Physics under the supervision of Prof. Manfred Kock. The diploma was finished in March 1995 and I worked a few further months as a Ph.D. student in Hannover. Since September 1995 I have been engaged as an information security analyst at debis Systemhaus GEI in Bonn, Germany. In these years, I have primarily worked on the security evaluation of products, e.g., smart cards. Since 1998, new findings on side channel analysis in the public domain have attracted me and side channel testing evolved to my major field of activity. In November 2003 I reduced to a part-time employment at T-Systems GEI (the legal successor of debis Systemhaus GEI) and started Ph.D. research. My thesis advisor is Prof. Christof Paar, Chair for Communication Security at the Ruhr University Bochum, Germany.

Publications

Kerstin Lemke-Rust

Books

[L1] Kerstin Lemke, Christof Paar, and Marko Wolf, editors. *Embedded Security in Cars*. Springer, 2005.

Chapters in Books

[L2] Kerstin Lemke, Ahmad-Reza Sadeghi, and Christian Stüble. *Embedded Security in Cars*, chapter: Anti-theft Protection: Electronic Immobilizers, pages 51–67. In Lemke et al. [L1], 2005.

[L3] Igor Furgel and Kerstin Lemke. *Embedded Security in Cars*, chapter: A Review of the Digital Tachograph System, pages 69–94. In Lemke et al. [L1], 2005.

[L4] Kai Schramm, Kerstin Lemke, and Christof Paar. *Embedded Security in Cars*, chapter: Embedded Cryptography: Side Channel Attacks, pages 185–204. In Lemke et al. [L1], 2005.

[L5] Kerstin Lemke. *Embedded Security in Cars*, chapter: Embedded Security: Physical Protection against Tampering Attacks, pages 205–217. In Lemke et al. [L1], 2005.

[L6] Kerstin Lemke and Christof Paar. *Encyclopedia of Cryptography and Security*, chapter: Physical Security. Springer, 2005.

[L7] Kerstin Lemke and Christof Paar. *D.A.CH Security 2006*, chapter: Seitenkanal-Analysen: Stand der Forschung in der Methodik, pages 280–291. syssec, 2006.

Articles

[L8] Claas Heise, Kerstin Lemke, and Manfred Kock. Full-dimensional Monte Carlo simulation of glow discharges with superposed magnetic fields. *Contrib. Plasma Phys.*, 37(5):431–450, 1997.

Conferences with Springer LNCS Proceedings

[L9] Bert den Boer, Kerstin Lemke, and Guntram Wicke. A DPA Attack against the Modular Reduction within a CRT Implementation of RSA. In B. S. Kaliski, Ç Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *LNCS*, pages 228–243. Springer-Verlag, 2003.

[L10] Kerstin Lemke, Kai Schramm, and Christof Paar. DPA on n-Bit Sized Boolean and Arithmetic Operations and Its Application to IDEA, RC6, and the HMAC-Construction. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *LNCS*, pages 205–219. Springer-Verlag, 2004.

[L11] Kerstin Lemke, Ahmad-Reza Sadeghi, and Christian Stüble. An Open Approach for Designing Secure Electronic Immobilizers. In Robert H. Deng, Feng Bao, HweeHwa Pang, and Jianying Zhou, editors, *Information Security Practice and Experience – ISPEC 2005*, volume 3439 of *LNCS*, pages 230–242. Springer, 2005.

[L12] Werner Schindler, Kerstin Lemke, and Christof Paar. A Stochastic Model for Differential Side Channel Cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2005*, volume 3659 of *LNCS*, pages 30–46. Springer, 2005.

[L13] Kerstin Lemke, Christof Paar, and Ahmad-Reza Sadeghi. Physical Security Bounds Against Tampering. In *Applied Cryptography and Network Security – ACNS 2006*, volume 3989 of *Lecture Notes in Computer Science*, pages 253–267. Springer, 2006.

[L14] Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. Stochastic Methods. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 15–29. Springer, 2006.

[L15] Kerstin Lemke-Rust and Christof Paar. An Adversarial Model for Fault Analysis against Low-Cost Cryptographic Devices. In *Fault Diagnosis and Tolerance in Cryptography – FDTC 2006*, volume 4236 of *Lecture Notes in Computer Science*, pages 131–143. Springer, 2006.

[L16] Dario Carluccio, Kerstin Lemke-Rust, Christof Paar, and Ahmad-Reza Sadeghi. E-Passport: The Global Traceability or How to Feel Like an UPS Package. In *Workshop on Information Security Applications – WISA 2006*, volume 4298 of *Lecture Notes in Computer Science*, pages 391–404. Springer, 2007.

[L17] Kerstin Lemke-Rust and Christof Paar. Gaussian Mixture Models for Higher-Order Side Channel Analysis. In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 14–27. Springer, 2007.

[L18] Kerstin Lemke-Rust and Christof Paar. Analyzing Side Channel Leakage of Masked Implementations with Stochastic Methods. In J. Biskup and J. Lopez, editors, *ESORICS 2007*, volume 4734 of *Lecture Notes in Computer Science*, pages 454–468. Springer, 2007.

Conferences with Proceedings

[L19] Sandeep Kumar, Kerstin Lemke, and Christof Paar. Some Thoughts about Implementation Properties of Stream Ciphers. In *SASC – The State of the Art of Stream Ciphers*, pages 311–319, 2004.

[L20] Dario Carluccio, Kerstin Lemke, and Christof Paar. Electromagnetic Side Channel Analysis of a Contactless Smart Card: First Results. In *Workshop on RFID and Lightweight Crypto*, pages 44–51, 2005.

[L21] Kerstin Lemke and Christof Paar. An Adversarial Model for Fault Analysis against Low-Cost Cryptographic Devices. In *Fault Diagnosis and Tolerance in Cryptography – FDTC 2005*, pages 82–94, 2005.

[L22] Lejla Batina, Sandeep Kumar, Joe Lano, Kerstin Lemke, Nele Mentens, Christof Paar, Bart Preneel, Kazuo Sakiyama, and Ingrid Verbauwhede. Testing Framework for eSTREAM Profile II Candidates. In *SASC 2006 - Stream Ciphers Revisited*, pages 104–112, 2006.

[L23] Dario Carluccio, Kerstin Lemke-Rust, Christof Paar, and Ahmad-Reza Sadeghi. E-Passport: The Global Traceability or How to Feel Like an UPS Package. In *Workshop on RFID Security 2006*, pages 167–179, 2006.

[L24] Gordon Meiser, Thomas Eisenbarth, Kerstin Lemke-Rust, and Christof Paar. Software Implementation of eSTREAM Profile I Ciphers on 8-bit AVR Microcontrollers. In *SASC 2007 – The State of the Art of Stream Ciphers*, pages 117–128, 2007.

[L25] Gordon Meiser, Thomas Eisenbarth, Kerstin Lemke-Rust, and Christof Paar. Efficient Assembly Implementations of Dragon, LEX, Salsa20 and Sosemanuk on 8-bit AVR Microcontrollers. In *WEWoRC 2007 – Western European Workshop on Research in Cryptology, Conference Record Draft, June 10, 2007*, pages 65–69, 2007.

[L26] Yifei Liu, Timo Kasper, Kerstin Lemke-Rust, and Christof Paar. E-Passport: Cracking Basic Access Control Keys with COPACOBANA. Accepted for *SHARCS'07 – Special-purpose Hardware for Attacking Cryptographic Systems*.

Conferences without Proceedings

[L27] Igor Furgel and Kerstin Lemke. A Review of the Digital Tachograph System. escar 2004, Embedded Security in Cars, 10.-11. November 2004, Bochum, Germany, 2004.

[L28] Kerstin Lemke, Ahmad-Reza Sadeghi, and Christian Stüble. Improving Electronic Immobilizers. escar 2004, Embedded Security in Cars, 10.-11. November 2004, Bochum, Germany, 2004.

Technical Reports

[L29] Martin Feldhofer, Kerstin Lemke, Elisabeth Oswald (Editor), François-Xavier Standaert, Thomas Wollinger, and Johannes Wolkerstorfer. State of the Art in Hardware Architectures. Technical Report D.VAM2, ECRYPT – European Network of Excellence in Cryptology, 2005.

[L30] Lejla Batina, Elke De Mulder, Kerstin Lemke, Stefan Mangard, Elisabeth Oswald, Gilles Piret, and François-Xavier Standaert (Editor). Electromagnetic Analysis and Fault Attacks. Technical Report D.VAM.4, ECRYPT – European Network of Excellence in Cryptology, 2005.

[L31] Lejla Batina, Elke De Mulder, Kerstin Lemke, Nele Mentens, Elisabeth Oswald, Eric Peeters, and François-Xavier Standaert (Editor). Report on DPA and EMA Attacks on FPGAs. Technical Report D.VAM.5, ECRYPT – European Network of Excellence in Cryptology, 2005.

[L32] Roberto Avanzi, Lejla Batina, Gerhard Frey, Pierrick Gaudry, Marc Joye, Tanja Lange (Editor), Kerstin Lemke, Elisabeth Oswald, Christof Paar, Dan Page, Christine Priplata, Nigel Smart, Colin Stahlke, and Ingrid Verbauwhede. Open Problems in Implementation and Application. Technical Report D.VAM.6, ECRYPT – European Network of Excellence in Cryptology, 2006.

Electronic Archives

[L33] Markus Kasper, Sandeep Kumar, Kerstin Lemke-Rust, and Christof Paar. A Compact Implementation of Edon80. eSTREAM, ECRYPT Stream Cipher Project, Report 2006/057, 2006. Available from: `http://www.ecrypt.eu.org/stream/papersdir/2006/057.pdf`.