Physical Security for Next Generation CMOS ICs

Dissertation

---

zur Erlangung des Grades eines Doktor-Ingenieurs (bzw. Ph.D.)
der Fakultät für Elektrotechnik und Informationstechnik
an der Ruhr-Universität Bochum

---

by Thorben Moos
Bochum, 2022

Copyright © 2022 by Thorben Moos. All rights reserved.
Printed in Germany.

PHYSICAL SECURITY FOR NEXT GENERATION CMOS ICS

DISSERTATION

zur Erlangung des Grades eines Doktor-Ingenieurs (bzw. Ph.D.)
der Fakultät für Elektrotechnik und Informationstechnik
an der Ruhr-Universität Bochum

Author:	**Thorben Moos**
Place of Birth:	Unna, Germany
Year of Publication:	2022
Thesis Advisors:	**Prof. Dr. Amir Moradi** Ruhr-Universität Bochum, Germany **Prof. Dr.-Ing. Christof Paar** Max-Planck-Institut für Sicherheit und Privatsphäre, Germany Ruhr-Universität Bochum, Germany **Prof. Dr. François-Xavier Standaert** Université catholique de Louvain, Belgium
Thesis Submitted:	November 16, 2021
Thesis Defense:	December 14, 2021
Last Revision:	September 2, 2022

To Ronja, the love of my life,
and to my parents for their endless love and support.

Abstract

The deployment of cryptography is no distinctive feature of military communication or global financial transactions anymore. Strong and efficient cryptographic algorithms are required for a surprisingly large number of applications in our daily life and get implemented into many devices with computing capabilities. Such devices range from the tiniest microcontrollers in the Internet of Things (IoT) to the most powerful high-end processors found in desktop computers or server farms. Despite the fact that microcontrollers and Central Processing Units (CPUs) posses the natural ability to execute cryptographic algorithms as software programs using general-purpose instructions, this approach usually lacks the desired efficiency for secured real-time communication. Thus, realizing cryptographic algorithms as dedicated hardware circuits is a necessary requirement for a wide variety of applications. The technical parameters for cryptographic hardware implementations can differ vastly depending on the designated application scenario. High-end processors used in smartphones, desktop computers or servers will require utmost speed and throughput, whereas tiny microcontrollers used in chip cards, wearables or even medical devices have to remain small, cheap and energy-efficient. The physical security of devices is another aspect that becomes relevant in this regard due to the varying adversary models. While modern cryptographic algorithms usually come with a set of mathematical security guarantees which attest their resistance to cryptanalytic and brute-force attacks, it is clear that such guarantees are only valid considering computationally-bounded adversaries in the black-box model. In embedded contexts, like the IoT, this model is not respected due to the constant physical exposure of the hardware to potential adversaries, including even the legitimate users. A black-box adversary might have a negligible chance of determining the secret key processed by a cryptographic algorithm when merely given access to the inputs and outputs. A gray-box adversary on the other hand, with additional access to the physical environment of the executing hardware, is often capable of reducing a device's security exponentially in the number of observations conducted. The collection and analysis of such additional information about the secret internals of cryptographic devices through the observation of their physical characteristics is commonly known as Side-Channel Analysis (SCA) and widely recognized as a severe threat to embedded security systems.

Since SCA attacks exploit vulnerabilities in concrete physical realizations of algorithms and not in the algorithms themselves, it is obvious that the underlying device technology plays a significant role in the assessment of the physical security of cryptographic hardware. As famously predicted by Gordon Moore in 1965 already, transistors and logic gates, the essential building blocks of computing technology today, have faced an aggressive scaling process over the past decades in order to fit continually more components onto a chip of the same size. It is no surprise that such an evolution is accompanied by notable changes in device behavior and even consequences for the security of cryptographic hardware. This thesis analyzes how the physical security of cryptographic hardware has been, and will be, affected by the technological advancement in the manufacturing process of Integrated Circuits (ICs). Additionally, it is a focus to provide first solutions for the uncovered challenges. The first and main part of this thesis deals with possibly the most prominent change in device behavior caused by the continuous shrinking of semiconductor technology, namely the idle or standby power consumption, also known as the static power consumption of the hardware. While the dynamic power consumed per logic unit declines in smaller process technologies, the static power consumption intensifies and becomes a new attack surface for physical adversaries. For this part of the thesis, multiple IC prototypes in continuously smaller feature sizes are developed, manufactured and analyzed to gain a better understanding of the evolution of this security threat. Developing this understanding is one of the core contributions of this thesis and of utmost importance for the protection of future security products. The second part of the thesis focuses on countermeasures against SCA attacks. Multiple hardware variants of the so-called masking countermeasure are analyzed with respect to their ability to provide the promised protection level in theory and practice. Additionally, improved evaluation strategies for masked implementations are suggested and examined. Both contributions are essential for the correct instantiation and evaluation of protected cryptographic hardware. The final part of this thesis deals with symmetric cryptographic algorithms deliberately optimized for high execution speed and low latency. The first of the two introduced block ciphers is a redesign of an existing primitive with the intention of increasing its security level without sacrificing performance. The second block cipher is an entirely new design engineered on gate level to offer maximum performance and security when realized in modern semiconductor technology. This primitive enables execution speeds of secure cryptography that were previously unattainable. At the same time, its enormous speed and parallelism help to prevent physical attacks.

Keywords.

Cryptography, Cryptographic Hardware, Integrated Circuits, Physical Security, Side-Channel Attacks, Static Power Consumption, Static Power Side-Channel Analysis, Masking, Leakage Assessment, Deep Learning, Low-Latency Block Ciphers

Kurzfassung

Physikalische Sicherheit von integrierten CMOS Schaltkreisen der nächsten Generation

Der Einsatz von Kryptographie ist mittlerweile kein Unterscheidungsmerkmal von militärischer Kommunikation oder globalen Finanztransaktionen mehr. Starke und effiziente kryptographische Algorithmen werden für überraschend viele Anwendungszwecke in unserem täglichen Leben benötigt und finden sich in allerlei Geräten die Rechenleistung besitzen. Solche Geräte reichen von kleinsten Mikrocontrollern im Internet der Dinge bis hin zu den leistungsstärksten Prozessoren, die man in Desktop-Rechnern oder Server-Farmen findet. Obwohl Mikrocontroller und andere Prozessoren naturgemäß im Stande sind kryptographische Algorithmen als Softwareprogramme mit Hilfe ihrer Universal-Instruktionen auszuführen, so ist dieser Ansatz oft nicht ausreichend leistungsfähig, um eine gesicherte Echtzeit-Kommunikation zu gewährleisten. Daher ist es eine notwendige Voraussetzung für eine Vielzahl von Anwendungen, dass kryptographische Algorithmen als spezialisierte Hardware Schaltkreise realisiert werden. Die technischen Parameter für kryptographische Hardware-Implementierungen können sich je nach Anwendungsszenario erheblich unterscheiden. High-End Prozessoren für Smartphones, Desktop-Rechner oder Server erfordern höchste Geschwindigkeit und Datendurchsatz, während Mikrocontroller, die in Chipkarten, tragbaren Kleinstgeräten oder sogar medizintechnischen Produkten eingesetzt werden, klein, günstig und energieeffizient bleiben müssen. Die physikalische Sicherheit der Geräte ist ein weiterer Aspekt, der in diesem Zusammenhang, aufgrund der verschiedenen Angreifermodelle, an Relevanz gewinnt. Obwohl moderne kryptographische Algorithmen üblicherweise eine Reihe mathematischer Garantien mit sich bringen, die ihre Resistenz gegenüber kryptanalytischen und Brute-Force-Angriffen attestieren, so ist klar, dass diese Garantien nur hinsichtlich Angreifern mit polynomial-begrenzter Rechenleistung im Black-Box Modell gelten. Im Kontext von eingebetteten Geräten, wie zum Beispiel dem Internet der Dinge, ist dieses Modell allerdings nicht zutreffend, da die Hardware dauerhaft potenziellen Angreifern ausgesetzt ist, einschließlich der legitimen Nutzer. Ein Black-Box Angreifer mag eine vernachlässigbare Erfolgsaussicht haben den geheimen Schlüssel zu bestimmen, welcher vom kryptographischen Algorithmus verwendet wird, wenn lediglich Zugang zu Eingaben und Ausgaben besteht. Ein Gray-Box Angreifer mit zusätzlichem Zugriff auf das physikalische Umfeld der Hardware kann jedoch oft die zugrundeliegende Sicherheit von Geräten exponentiell in der Anzahl der vorgenommenen Beobachtungen reduzieren. Die Sammlung von solch zusätzlichen Informationen über die geheimen Interna der kryptographischen Geräte durch die Beobachtung ihrer physikalischen Eigenschaften ist allgemeinhin als Seitenkanalanalyse bekannt und wird allseits als ernsthafte Bedrohung für eingebettete Geräte eingestuft.

Da Seitenkanalangriffe Schwachstellen in konkreten physikalischen Instanzen von Algorithmen und nicht in den Algorithmen selbst ausnutzen, ist es offensichtlich, dass die zugrundeliegende Halbleitertechnologie eine wichtige Rolle in der Bewertung der physikalischen Sicherheit kryptographischer Hardware spielt. Wie bekanntermaßen von Gordon Moore bereits im Jahre 1965 prognostiziert wurde, haben Transistoren und Logik-Gatter, die elementaren Bausteine für heutige Computer-Technologie, in den letzten Jahrzehnten einen aggressiven Verkleinerungs-Prozess durchlebt, um eine stetig wachsende Menge an Komponenten auf einem Chip gleichbleibender Größe unterzubringen. Es ist keine Überraschung, dass eine solche Evolution mit bedeutenden Veränderungen des Geräte-Verhaltens einhergeht und sogar Konsequenzen für die physikalische Sicherheit kryptographischer Hardware mit sich bringt. Diese Dissertation analysiert wie die physikalische Sicherheit kryptographischer Hardware vom technologischen Fortschritt in der Fertigungstechnologie integrierter Schaltkreise beeinflusst wurde und noch werden wird. Zusätzlich wird ein Fokus darauf gesetzt erste Lösungsansätze für die aufgedeckten Herausforderungen zu liefern. Der Hauptteil dieser Dissertation befasst sich mit der womöglich prominentesten Veränderung im Geräte-Verhalten, die durch die stetige Verkleinerung der Halbleitertechnologie ausgelöst wurde, nämlich dem Stromverbrauch im Leerlauf oder Bereitschaftszustand der Geräte, auch bekannt als statischer Stromverbrauch. Während der dynamische Stromverbrauch pro Logikeinheit in kleineren Prozesstechnologien abnimmt, so wächst der statische Stromverbrauch und wird zu einer neuen Angriffsfläche für physikalische Widersacher. Für diesen Teil der Dissertation wurden mehrere integrierte Schaltkreis-Prototypen in stetig kleiner werdenden Strukturgrößen entwickelt, gefertigt und analysiert, um ein tieferes Verständnis der Evolution dieser Sicherheitsbedrohung zu erlangen. Die Entwicklung dieses Verständnisses ist einer der wichtigsten Beiträge dieser Dissertation und von immenser Bedeutung für den Schutz zukünftiger Sicherheits-Produkte. Der zweite Teil dieser Dissertation befasst sich mit Gegenmaßnahmen gegen Seitenkanalangriffe. Mehrere Hardware-Varianten der sogenannten Maskierungs-Gegenmaßnahme werden auf ihre Fähigkeit hin überprüft das versprochene Sicherheitsniveau sowohl in Theorie als auch Praxis einzuhalten. Außerdem werden verbesserte Evaluationsmethoden für maskierte Implementierungen vorgeschlagen und analysiert. Beide Beiträge sind essenziell für die korrekte Instanziierung und Evaluation von geschützter kryptographischer Hardware. Der letzte Teil dieser Dissertation beschäftigt sich mit symmetrischen kryptographischen Algorithmen die bewusst für eine hohe Ausführungsgeschwindigkeit und geringe Latenz optimiert wurden. Die erste der beiden eingeführten Blockchiffren ist eine Neugestaltung einer bereits existierenden Primitive mit der Intention das Sicherheitsniveau zu erhöhen ohne Leistung einzubüßen. Die zweite Blockchiffre beruht auf einem völlig neuen Entwurf, der auf Gatter-Ebene entwickelt wurde, um maximale Performanz und Sicherheit in modernen Halbleitertechnologien zu bieten. Dadurch werden Ausführungsgeschwindigkeiten von sicherer Kryptographie ermöglicht, die zuvor nicht erreichbar waren. Gleichzeitig helfen die enorme Geschwindigkeit und Parallelität physikalische Angriffe zu unterbinden.

Schlagworte.

Kryptographie, kryptographische Hardware, Integrierte Schaltkreise, Physikalische Sicherheit, Seitenkanalangriffe, statischer Stromverbrauch, Seitenkanalanalyse des statischen Stroms, Maskierung, Beurteilung des Informationsverlustes, maschinelles Lernen, Hochgeschwindigkeits-Blockchiffren

Acknowledgements

This thesis is the result of around five years of research at the Chair for Embedded Security (Emsec) headed by Prof. Dr.-Ing. Christof Paar and the Chair for Security Engineering (Seceng) headed by Prof. Dr.-Ing. Tim Güneysu, both at the Ruhr-Universität Bochum under supervision of Prof. Dr. Amir Moradi.

First of all, I would like to thank my parents, my brother and my friends who always supported me and gave me the energy to pursue a PhD. A special thank you goes to my wonderful girlfriend Ronja. Without your patience, love and support none of this would have been possible. I will forever be grateful for keeping me sane during the more demanding stretches of this journey.

I would like to thank my advisor Amir Moradi for accepting me as a doctoral student and for introducing me to the beautiful world of hardware security. You have been a terrific mentor and I am deeply grateful to you for supporting me, pushing me to new heights and for explaining the academic world to me. Your drive and energy in work and research are truly inspiring. I also want to thank you for providing me with the necessary freedom and funding to pursue my own research. Only after years of complaints from other PhD students I started to realize how lucky I was to work on a DFG-funded project all by myself.

I also owe thanks to Christof Paar and Tim Güneysu for initially igniting my passion for cryptography from an engineering perspective while I was still an undergraduate. Oliver Hartkopp, Hendrik Decke, Georg Becker and Bastian Richter deserve recognition for making my bachelor's and master's theses enjoyable and memorable. Ekrem Aydin and Christoph Herbold deserve a thank you for being a terrific study group during my masters. Falk Schellenberg and David Oswald gave me my first student assistant position at the Chair for Embedded Security which indubitably influenced my future as I immediately liked the friendly and serene atmosphere in the group.

I also want to thank François-Xavier Standaert, who often inspired me with his research from afar, for agreeing to be my external referee and for taking the time to read this thesis.

Further thanks goes to all my co-authors (in alphabetical order): Dušan Božilov, Maria Eichlseder, Bijan Fadaeinia, Samaneh Ghandali, Naghmeh Karimi, Miroslav Knezevic, David Knichel, Baptiste Lambin, Gregor Leander, Oleksiy Lisovets, Amir Moradi, Nicolai Müller, Ventzislav Nikov, Christof Paar, Bastian Richter, Tobias Schneider, François-Xavier Standaert, Yosuke Todo, Felix Wegener and Friedrich Wiemer.

I had a fantastic time at Emsec and Seceng and enjoyed many fruitful discussions with almost everyone about work and non-work related topics, resulting in joint projects, but also new friendships. For this reason I want to thank all my current and former colleagues at the Ruhr-Universität Bochum, but especially (in no particular order): Falk Schellenberg and Bastian Richter for guiding me at the start of my doctorate, for the many amazing travels to sunny and not-so-sunny conference locations and for always taking the time to answer my silly questions about electronics and engineering, Steffen Becker and Shahram Rasoolzadeh for becoming great friends and for the many tremendous evenings we have spent together, Felix Wegener for all the projects we discussed but never had the time to bring to life, Anita Aghaie for a lot of fun times and for sharing the quirks of living in one of Bochum's loudest placest, David Knichel and Aein Rezaei Shahmirzadi for sharing an office with me for some time, having great fun

even when working with 3 people in a small room in the summer, and for always being great office neighbors, Bijan Fadaeinia for being my latest office partner although we could not spend much time together due to COVID-19, Tobias Schneider and Pascal Sasdrich for their honest consulting about the pros and cons of becoming a postdoc, Irmgard Kühn, Horst Edelmann and Jennifer Knothe for always taking so much work off of all our shoulders and for wonderful Glühwein-evenings at the christmas market, Pierre Mayr for always being helpful with all things related to Europractice, the whole Ožujsko group (Steffen Becker, Maik Ender, Max Hoffmann, Shahram Rasoolzadeh, Felix Wegener) who contributed to one of the most special and memorable business trips of my time, all CASA PhDs for having lots of fun during colloquia and retreats and finally everyone who ever participated in our regular soccer meetings or helped to win trophies at the annual HGI tournament.

Thank you.

Table of Contents

Imprint iii
Abstract vii
Kurzfassung ix
Acknowledgements xi

I Preliminaries 1

1 Introduction 3
1.1 Motivation 3
1.2 Structure of this Thesis 8
1.3 Summary of Research Contributions 8
1.3.1 Static Power Side-Channel Analysis 8
1.3.2 Evaluation of Masked Implementations 13
1.3.3 Low-Latency Block Ciphers 16

2 Background 21
2.1 Cryptography 21
2.1.1 Symmetric Cryptography 22
2.1.2 Asymmetric Cryptography 22
2.2 Cryptographic Hardware 23
2.2.1 Digital CMOS Integrated Circuits 23
2.2.2 Application-Specific Integrated Circuits (ASICs) 24
2.2.3 Field Programmable Gate Arrays (FPGAs) 25
2.2.4 Printed Circuit Boards (PCBs) 26
2.3 Physical Security of Cryptographic Devices 27
2.3.1 Side-Channel Analysis 27
2.3.2 Fundamental Limits and Information Leakage of Computation 27
2.3.3 Power Consumption of CMOS Devices 28
2.3.4 Power Analysis Attacks 31
2.3.5 Countermeasures 33
2.3.6 Leakage Assessment 38
2.3.7 Deep Learning in Side-Channel Analysis 40
2.4 Measurement Techniques and Equipment 41
2.4.1 Dynamic Power Measurements 41
2.4.2 Static Power Measurements 42
2.4.3 List of Equipment used in this Thesis 46

II Publications 49

3 Static Power Side-Channel Analysis 51
3.1 Static Power Side-Channel Analysis - An Investigation of Measurement Factors . 51
3.2 Static Power SCA of Sub-100 nm CMOS ASICs 67
3.3 Exploring the Effect of Device Aging on Static Power Analysis Attacks 101
3.4 Unrolled Cryptography on Silicon 127
3.5 Countermeasures against Static Power Attacks 157

4 Evaluation of Masked Implementations 185
4.1 On the Easiness of Turning Higher-Order Leakages into First-Order 185
4.2 Glitch-Resistant Masking Revisited 205
4.3 Deep Learning Leakage Assessment 245

5 Low-Latency Block Ciphers 295
5.1 PRINCEv2 295
5.2 The SPEEDY Family of Block Ciphers 327

III Conclusion 365

6 Conclusion and Open Problems 367

IV Appendix 369

Bibliography 371

List of Figures 383

List of Tables 385

About the Author 387

Publications and Academic Activities 388

Part I

Preliminaries

Chapter 1

Introduction

This chapter contains a brief introduction to cryptographic hardware and physical attacks and motivates the continued need for research and novel solutions in both areas as the underlying circuit technology advances into smaller and smaller geometrical dimensions. Afterwards, the structure of the remainder of this document is detailed and the scientific contributions of the publications accumulated in this thesis are summarized.

Contents of this Chapter

1.1 Motivation . . . 3
1.2 Structure of this Thesis . . . 8
1.3 Summary of Research Contributions . . . 8

1.1 Motivation

The rapid transition over the past decades to the current era of connectivity and pervasive computing was fueled by two primary factors. First, the technical ability to mass-produce integrated circuits in advanced semiconductor technology nodes at a moderate price per unit and second, the availability of strong and robust cryptography. The first one allows to equip all kinds of electronic devices, far beyond regular computers or servers, with sufficient computing power to process significant amounts of data, execute relatively complex algorithms and exchange information between each other via networks. The second one is vital to ensure the confidentiality, authenticity, integrity and privacy of the communicated data. While the first factor may be fairly obvious, the second one is often hidden from the ordinary user despite its indispensability for many features that are taken for granted nowadays. It is truly unimaginable how wireless communication for example would look like without the ability to verify that the received content was indeed sent by the expected communication partner or the assurance that the transferred information can only be read by the intended parties. Electronic payment or withdrawal of money with credit and debit cards are further applications that would not exist without strong electronic authentication mechanisms. Modern cryptographic primitives are capable of providing the desired assurances and verifiability based on a sound mathematical foundation and years of public scrutiny. Hence, they are often the most trusted components in today's security solutions. Public research in cryptography has come a long way in the last couple of decades and designers of security systems have a broad spectrum of cryptographic primitives at their disposal, many of which are deliberately tailored for specific purposes, applications or optimization goals. Yet, mathematically secure and efficient algorithms are only one building block for secure communication in an ubiquitous world.

Threat Models for Cryptographic Hardware. At the beginning of the digital revolution in the latter half of the twentieth century, cryptography was mostly used in the traditionally envisioned setting, namely for encrypting messages to transfer them between two distant parties over an insecure channel. In this setting, quite clearly, only the communication channel is insecure. The cryptographic operations are performed in a private and safe environment without any exposure to adversarial entities. Thus, mathematical security of the applied primitives in a *black-box* model was the only important criterion, besides some efficiency considerations. However, when pervasive computing technology started to get more prevalent in daily life, becoming part of portable devices and everyday objects, the threat model for secure communication changed noticeably. Suddenly, unauthorized figures had physical access to the machines that were executing the cryptographic algorithms. In this case, the mathematical operations for encryption, decryption, signature or verification are not carried out in a private and safe environment anymore. Instead, a physical realization of the cryptographic primitive has to perform the calculations whenever and wherever the device is used, even in a hostile environment under adversarial exposure. Additionally, manufacturers of electronic authorization devices like smart cards were forced to engrave the cryptographic key material permanently into the hardware. Once deployed in the field, such physical tokens are generally not connected to a network and therefore can hardly be updated or establish new keys. These developments led to the inconvenient situation that unauthorized entities can obtain possession of devices that permanently store sensitive key material without the possibility to be updated and that willingly perform calculations on this data whenever the operator demands. Potential attackers, a group which includes even the legitimate owners of the devices, may exploit this opportunity to extract the secret material to use it to their personal advantage. Common examples include the malicious replication of a pay TV card or an electronic key of a rented car, or increasing the available balance on a prepaid card for digital payments. Evidently, there are clear incentives for all kinds of actors to tamper with cryptographic devices, especially the embedded ones, in order to circumvent the security features installed by the manufacturer. As one may suspect, this situation led to the advent of a new class of attacks.

Implementation Attacks. Although opening up a device and imaging its surface or probing specific wires may be the most intuitive option for searching and extracting a secret key engraved in the hardware, another class of techniques proved to be much more effective and significantly harder to defend against. In detail, the physical accessibility allows adversaries to observe the global electrical characteristics and emissions of devices during the execution of cryptographic algorithms. Interestingly, it was found that the physical dissipation of common hardware platforms depends measurably on the internally processed data. Without the need to open up a device, an adversary may simply monitor a global physical quantity like the power consumption of the hardware during normal operation and can use the recorded information to learn portions of the internally processed values. In this scenario, the cryptographic algorithms are not operating in a black-box model anymore, but rather in a *gray-box* model, due to the additional information that becomes available to the adversary through so-called side channels.

The malicious exploitation of the information leaked through these side channels, to extract the secret key for example, is commonly known as a Side-Channel Analysis (SCA) attack. The repetitive processing of a fixed symmetric encryption key by a block cipher implementation on a smart card is a prime example of a potentially vulnerable target. Within certain limits it is even possible to influence the execution of the cryptographic algorithms by affecting the physical environment of the devices. When a controlled miscalculation can be provoked by actively tampering with the device during cryptographic operations, it is also possible to extract information about the intermediate values of the computed function. These techniques are called Fault Injection (FI) attacks, and in combination with SCA they form the field of implementation attacks. Cryptographic primitives developed for security in the black-box model cannot automatically withstand attacks that make use of their intermediate results. In fact, it is still an open problem how to best perform cryptographic operations in the gray-box model without revealing secret information. Plenty of countermeasures have been developed to thwart implementation attacks, but none of them provides optimal security. Usually, the designer of a cryptographic hardware product has to make an informed decision regarding the trade-off between the security level provided by a protection mechanism, which should also depend on the value of the protected assets, and the costs for its implementation.

Countermeasures against Physical Attacks. The cost of a countermeasure against SCA or FI attacks is usually measured as the overhead of the protected implementation of the cryptographic algorithm compared to its raw and unprotected realization. With respect to hardware implementations such an overhead can manifest itself as a size overhead, which means a larger number of digital elements or standard cells are required to implement the circuit, or it can be an execution time overhead, which means a larger number of clock cycles are required to execute the algorithm or, alternatively, the clock frequency for its execution is reduced. There are further criteria which may play a role, like the energy consumption or the required number of fresh random bits for example, but most of these are either strongly correlated with the size and speed parameters or they are specific to certain types of protection mechanisms. FI countermeasures are typically based on some form of redundancy. This may include area-, time- and information-redundancy, as well as any possible hybrid combination between them. In the field of SCA countermeasures on the other hand it is generally distinguished between *hiding* and *masking* schemes. Hiding schemes aim to reduce the Signal-to-Noise Ratio (SNR) in the measurements that a physical adversary is able to collect. This is either achieved by creating additional noise to bury the signal in, or by reducing the amplitude of the exploitable signal to sink it in the existing noise. The primary shortcoming of hiding schemes is that their implementation overhead is often more or less proportional to the increase of the physical security level of the resulting circuits. Thus, simply put, achieving a very high protection level with a hiding scheme is also very costly. Masking schemes on the other hand apply a general method known as secret sharing to force SCA adversaries to combine the leakages of multiple circuit parts or operations in order to learn information by recombining the shared secrets. This concept shifts the observable data dependencies to higher statistical moments and conceptually parallels an amplification of the noise level by a factor that grows exponentially in the chosen security order (often related to the number of shares). For this reason, masking schemes, when correctly implemented, have the remarkable property of increasing the complexity of SCA attacks exponentially in the security order while the designer has to spend (approximately) a quadratic implementation overhead [PR13, FGP$^+$18]. Clearly, this ratio between overhead spent and security obtained is preferable over the proportional relation of hiding schemes. Thus, it is no surprise that masking has gained high popularity in industry and academia as a countermeasures against side-channel attacks. Yet, these schemes are not a perfect cure for the side-channel problem either and come with their own share of complications. For instance, they typically have to make certain assumptions about the environment like a sufficiently high noise level, the independence of the leakages of different circuit parts or operations, and the availability of unpredictable, fresh and uniformly distributed random values. It has become clear throughout the past decades of research in this field that implementing the masking countermeasure properly while enforcing the independence assumption is anything but trivial, especially with respect to hardware circuits where the designer needs to take the additional impact of physical defaults such as glitches into account. The public knowledge about leakage effects in hardware was clearly insufficient in the early stages of hardware masking research which often led to the unexpected and undesired recombination of shared secrets in supposedly secure implementations. This negative example has taught the research community that theoretical and practical evaluations have to go hand in hand when developing protection mechanisms against implementation attacks. It is important to develop accurate models based on the insights gained by practical experiments and to subsequently proof the security of schemes in these models. Afterwards, it has to be verified whether the theory holds in practice and if this is not the case, adjust the developed models. Clearly, robust theoretical as well as practical evaluation tools and methods are required to perform this process in a reliable and efficient manner. While formal security proofs are already the state of the art in the field of software-based masking, hardware masking research is still lagging behind at the time of writing this thesis since many proposed schemes rather rely on engineering intuition. Hence, the unification of accurate models, formal security proofs and practical evaluation methodologies is crucial for the development of provably secure hardware masking schemes which are freely scalable to arbitrary protection orders and may still be implemented efficiently.

Evolution of the Device Technology. The vast majority of computing hardware today is fabricated based on Complementary Metal-Oxide-Semiconductor (CMOS) technology. To put it briefly, CMOS technology combines complementary and symmetrical pairs of field-effect transistors (p-type and n-type) to form digital logic elements. This technological innovation outclassed all competitors on its path to worldwide supremacy in integrated circuit production during the 1980s, in part due to one important electrical property, namely the negligible energy consumption in stable states. Traditionally, CMOS logic gates are constructed in a specific way to consume only a minimal amount of energy when all input signals are kept at a stable voltage level. However, the technological progress in semiconductor manufacturing, in particular the continuous down-scaling of device geometries, has left its mark on the physical behavior of CMOS-based circuits. Leakage currents, which are defined as the undesired transfer of electrical energy across a boundary that is technically viewed as insulating, kept growing in advanced technologies and contradicted the notion of a negligible energy wastage in stable states. While the scaling process faced many tough engineering challenges as it moved towards and beyond the deep sub-micron range, few were as demanding as keeping the leakage currents in nanometer-scaled field-effect transistors under control. The difficulty of this problem arises from the fact that typical scaling parameters like the channel length, the supply voltage, the threshold voltage or the gate oxide thickness directly affect the static power consumption of transistors [Hel09]. Nowadays, as a consequence of the leakage currents in individual transistors, CMOS logic cells manufactured in advanced fabrication processes conduct a measurable static current during idle times whose magnitude depends on the voltage levels at the cells' respective inputs and outputs. This is particularly inconvenient for the development of low-power designs. It also creates a new attack surface for physical adversaries, as in digital computing the voltage levels of signal lines are synonymous with the transported logical values. In fact, this unintended relation between internally processed values and externally observable device behavior forms a new side channel which may endanger the security of cryptographic hardware. The feasibility of attacks based on this concept has been continuously favored over the past decades by the technological progress, as the static power consumption is condemned to increase as a natural consequence of the down-scaling of CMOS technology. Hence, it has become a crucial and urgent task to thoroughly examine how the physical security of modern devices is affected by the evolution of the manufacturing process of integrated circuits, and which countermeasures against physical attacks are still able to provide the desired protection in cutting-edge semiconductor technology.

New Possibilities for High-Performance Cryptography. The motivation behind the continuous down-scaling of CMOS technology is primarily of economic nature. Fitting more transistors onto a chip of the same size will decrease the price per transistor in the long run, even if the manufacturing process is more costly in the beginning. The area or size required to implement an algorithm or a functionality on a silicon chip, typically measured in Gate Equivalents (GE), is generally the primary cost factor. Due to the lower price per transistor, the cost per logic unit or per fixed area dimension is also reduced with each passing year. Secure cryptographic algorithms on the other hand, especially the symmetric ones, do not require a larger implementation size today than they did twenty years ago. In fact, the opposite is usually the case as cryptographic primitives have become more efficient and hardware-oriented and because many more lightweight primitives have been developed and introduced by the research community since then. Additionally, it has been explored how to design cryptographic algorithms in such a way that the overhead required to equip them with countermeasures against physical attacks is reduced. Given these trends, it is at least doubtful that the area or size of cryptographic hardware implementations remains the prohibitive factor for their integration into computing devices. In fact, the reduced economic price tag for the realization of cryptographic hardware opens up new possibilities. As soon as the implementation size is not the focal point anymore, other optimization goals can be prioritized. One of these goals is low latency. For many applications inside of modern processing units it can be valuable to have cryptographic implementations available which offer extremely high performance, or in other words an extremely low execution time. Indeed, current trends are pointing towards more encrypted communication inside of processors in the form of memory encryption, secure cache architectures, pointer authentication and similar protection mechanisms. This development has been spurred in part by the myriad of microarchitectural attacks discovered in recent years (e.g., [LSG$^+$18, KHF$^+$19]). Cryptographic operations in these environments typically need to process information as quickly as possible, ideally in just a single clock cycle at a high operating frequency, while still delivering appropriate levels of security. Developing symmetric cryptographic primitives with optimal performance for these applications requires adapting the algorithms as much as possible to the latency properties of the underlying semiconductor technology. Thus, it is crucial for the designers of such specialized primitives to keep up with the industrial progress in integrated circuit fabrication in order to sustain and advance the security and performance of future device generations.

1.2 Structure of this Thesis

This thesis consists of three main parts divided into a total of six chapters. Part I *Preliminaries* contains two chapters, Chapter 1 *Introduction* and Chapter 2 *Background* (this paragraph is part of Chapter 1). The *Introduction* chapter starts with motivating the research questions this thesis attempts to answer, then details the structure of the thesis and finally presents a summary of the relevant contributions to the scientific field. The *Background* chapter recalls the scientific concepts that are crucial for the understanding of this work, in particular considering the subjects cryptography, cryptographic hardware, the physical security of cryptographic hardware and finally the measurement techniques required to perform experimental evaluations in this field. Part II *Publications* covers all papers published in peer-reviewed journals and conference proceedings accumulated in this thesis. The publications are divided into the three sub-topics corresponding to Chapter 3 *Static Power Side-Channel Analysis*, Chapter 4 *Evaluation of Masked Implementations* and Chapter 5 *Low-Latency Block Ciphers*. Finally, Part III *Conclusion* is composed of Chapter 6 *Conclusion and Open Problems* which summarizes the results of this thesis and provides an outlook for future research in the considered fields.

1.3 Summary of Research Contributions

In this section we describe the fundamental contributions in areas of interest to the cryptographic research community made by the publications accumulated in this thesis. The common topic is the physical security of cryptographic hardware implementations in current and future device technologies. This subject is approached in three different parts, corresponding to Chapters 3, 4 and 5. In the following we summarize the research contributions for each part individually.

1.3.1 Static Power Side-Channel Analysis

One of the primary concerns with respect to the security of cryptographic hardware in the future, and in particular the way it might be affected by the continuous shrinking of CMOS technology, is the advent of Static Power Side-Channel Analysis (SPSCA) attacks. The down-scaling of chip technology over the past decades has led to a decline of the dynamic power consumption per logic unit, which makes it noticeably more difficult to target the dissipation of small parts of a circuit in typical divide and conquer based power analysis attacks. However, while this evolution is a welcome one from a security point of view, the static power consumption takes the opposite direction and intensifies in newer technology generations to reach a significant magnitude in sub-100 nm CMOS technology – a developmental stage that has long become the state of the art for many device families. These contrary trends actually beg the question whether the static power consumption is not becoming, or has already become, the more attractive target for adversaries against cryptographic hardware in advanced technology nodes. Before the research presented in this thesis had been conducted, only a very limited amount of empirical results had been made publicly available in this area. The main reason for that is simply the difficulty of performing a scientific and comprehensive study in this field, as suitable cryptographic test chips for such purposes are not commercially available. Thus, custom chip prototypes need to be developed, fabricated and analyzed. Prototyping custom chips in cutting-edge semiconductor technology is not only quite expensive, but also very time consuming. For this thesis, however, we have accepted this challenge and investigated the evolution of the static power side channel across multiple CMOS generations. In the following we present the relevant research results we have obtained and published based on our analysis of four Application-Specific Integrated Circuits (ASICs) that have been developed, manufactured and analyzed specifically for this thesis. Our main strategy to gain a better understanding of the principles and mechanisms behind static power side-channel leakage was to concentrate on the fundamental differences to its counterpart, the dynamic power side channel. Pointing out the differences between the two side channels also helps to answer the question in which scenarios the static power is actually the preferable attack vector for an adversary. It was discovered that several characteristics of the static power consumption make it particularly dangerous as a source of information leakage. Establishing this understanding is clearly helpful for the development of effective countermeasures. Thus, in the list below we describe the most relevant peculiarities of the static power consumption of CMOS devices that differentiate it as a side channel from the dynamic power consumption.

(1) The static power consumption and its potency as a side channel increases significantly when down-scaling the physical feature size of the underlying CMOS technology.

  - *Our experiments, which we performed at different operating conditions, show consistently that the ASIC technology with the smaller minimum feature size indeed exhibits substantially more informative leakages than the one manufactured in the larger technology, even though all targeted instances have been derived from identical RTL code and were implemented using an identical design procedure* [Moo19].
  - *As a result of this comparison, we conclude that the data-dependent currents increase drastically when moving towards smaller CMOS technology nodes* [Moo19].
  - *The leakage exhibited by the 65 nm ASIC is roughly* 10× *as informative as the one on the 90 nm chip* [Moo19].

(2) The static power consumption and its potency as a side channel increases significantly when raising the supply voltage and/or the temperature of the device under test.

  - *By raising the temperature from* 20 ℃ *to* 90 ℃ *and the core voltage from* 1.2 V *to* 1.6 V *the difference of means between two leakage distributions can be amplified by a factor of approximately 12 on the 65 nm chip* [Moo19].
  - *Since the amount of leakage current is increased in higher temperatures and for higher supply voltages, static power analysis attacks are usually conducted while the device is operated at a high temperature, e.g.,* 90 ℃*, and an increased supply voltage, which leads to more easily exploitable leakages, i.e., lower number of traces for successful attacks* [KMM19].
  - *For the higher temperatures it becomes obvious that the number of measurements that are required to extract the same amount of information is reduced in an exponential manner* [MMR20].

- *Therefore, we are able to confirm that the manipulation of operating conditions is a viable method to enhance the magnitude of the leakage currents and to improve the overall quality of the measurement results* [MM21].

(3) Since the static power is typically measured over a longer time period (common are 10 ms to 1000 ms), it is possible to average out several sources of noise and obtain low-noise measurements. This is a danger to masking schemes and complicates the use of certain evaluation methodologies commonly applied in SCA contexts.

- *When we extended the measurement interval from 10 ms to 500 ms and averaged over 500 000 time samples, the standard deviation of the measured static power values decreased by a factor of over 2.5* [MMR17].
- *By means of a masking scheme, it is possible to achieve a security level, in terms of required number of side-channel observations for a successful attack, that grows exponentially in the masking order. However, masking can only deliver such a security guarantee in case the leakage of the individual shares is sufficiently independent and the traces that an adversary can acquire are sufficiently noisy. [...]. We investigate the susceptibility of masked implementations to SPSCA and conclude that due to noise reduction techniques (i.e., averaging over time) adversaries can obtain measurements with such a low noise influence that masking is essentially ineffective* [Moo19].
- *We performed a detailed study on how the reduction of the noise level in static leakage measurements affects the security provided by masked implementations. As a result of this study, we do not only find out that the threat for masking schemes is indeed real, but also that common leakage assessment techniques, such as the Welch's t-test, together with essentially any moment-based analysis of the leakage traces, is simply not sufficient in low-noise contexts* [Moo19].
- *We argue that state-of-the-art leakage assessment techniques like the Welch's t-test are not suitable when analyzing masked implementations in very low noise environments as they cause false negatives* [Moo19].
- *Our experiments on a 150 nm CMOS ASIC reveal that with respect to the signal-to-noise ratio in static power side-channel analyses, stretching the measurement interval decreases the noise exponentially (up to a certain point)* [MMR20].

(4) Data-dependent static power is consumed at any moment in time and therefore can be exploited even at times when no sensitive data is actively processed.

- *If cryptographic co-processors do not clean all their internal memory elements (e.g. registers) after each key-involving cryptographic operation, there is a high risk that sensitive data may remain in the circuit and can be extracted long after the cryptographic operation is finished* [Moo19].
- *In case a cryptographic implementation does not ensure that any sensitive intermediate information is present for at most a few clock cycles in the circuit, this implementation can be susceptible to a static power analysis without the adversary having control over the clock signal* [Moo19].
- *It is very well possible to measure the static currents associated with an intermediate value, even when other computations are performed at time of measurement. It is*

just required that the value remains long enough unchanged in order to measure it precisely [Moo19].

- *The nature of static power analysis is entirely different from dynamic power analysis, since the former does not exploit a momentary transitional effect that can be observed for a finite period of time only. Instead, it is based on observing a static phenomenon that can be quantified for as long as no transition occurs in the targeted circuit part* [Moo20].
- *It was discovered that static power SCA attacks can be performed without obtaining control over the clock signal of the device under test (DUT) when sensitive intermediates remain in the circuit after cryptographic operations and are not subject to an immediate modification* [Moo20].
- *Unrolled circuits which are instantiated without any considerations of this issue, are a prime example of implementations where the full state, containing all sensitive intermediates, remains in the circuit between any two consecutive encryptions* [Moo20].
- *Sensitive information is often left behind by cryptographic co-processors after their operation, which allows the extraction of secret data even without any clock control abilities* [MM21].

(5) Univariate resistance with respect to dynamic power attacks does not equal univariate resistance with respect to static power attacks.

- *Masking schemes with a sequential manipulation of the shares (typical in software) might be in danger when an exploitation of the leakage currents is possible, since the shares may be leaked in a univariate fashion through the static power, making multivariate attacks unnecessary and potentially reducing the effective noise level* [Moo19].
- *Univariate leakage with respect to static power adversaries is much more inclusive than univariate leakage with respect to dynamic power adversaries. A static power adversary can virtually see the cumulative leakage of any gate in a circuit in a single snapshot and not only the leakage of gates that switch simultaneously* [MM21].

All those peculiarities of the static power consumption as a side channel may actually lead to, or can be abused to create, scenarios where this source of information leakage becomes the weak point of a cryptographic hardware implementation with respect to its physical security. Therefore, it is crucial to be aware of this threat when designing cryptographic hardware that should be resilient against physical adversaries. To summarize, the impact of the static power as a security threat is naturally amplified by the technological advancement itself and it can be artificially amplified by adversaries via controlling the environmental conditions. Additionally, the static power side-channel leakage can often be used to circumvent established dynamic power countermeasures (e.g. masking due to noise reduction) or extract data that is not even processed during the time of attack. All these discoveries highlight the potential impact and potency of static power attacks and emphasize the urgent need for countermeasures against this threat. However, to design effective protection mechanisms it is necessary to first develop a deeper understanding of the principles and mechanics that show the potential to mitigate static power attacks. In order to explore this subject we have implemented multiple combined masking and hiding countermeasures on a custom 28 nm CMOS ASIC and evaluate their ability to prevent

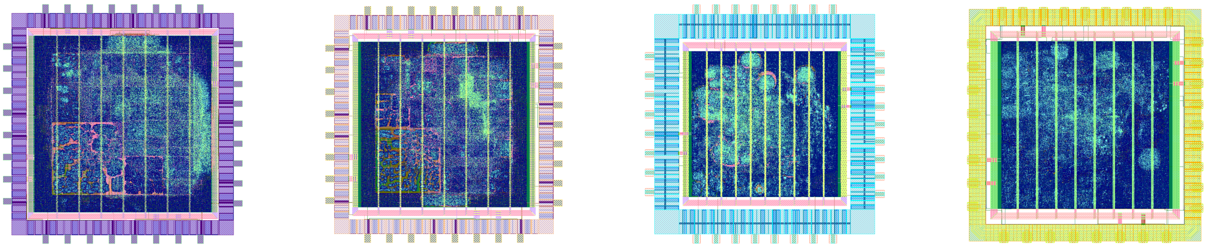

(a) 90 nm (b) 65 nm (c) 40 nm (d) 28 nm

Figure 1.1: This figure shows the four different ASIC prototypes which have been developed, manufactured and analyzed specifically for this thesis.

the extraction of secret information through the static power consumption when manufactured in advanced nanometer technologies [MM21]. In detail, the ASIC contains eleven different cryptographic co-processors based on the PRESENT block cipher [BKL$^+$07] with different levels of SCA protection applied. We also propose the first ever standard-cell-based balancing scheme that provides perfect data independence under the assumption that multiple instances of the same standard cell on the same chip have the same exact leakage characteristics. Of course, in reality this assumption will not hold due to the existence of intra-die process variations and aging-related degradation effects [KMM19]. Yet, this scheme, called Exhaustive Logic Balancing (ELB), is likely as close to a data-independent leakage current as one may get. As a result of the experiments presented in [MM21], it was found that the strongest protection was achieved by a combination of ELB and Threshold Implementation (TI), a provably secure hardware masking scheme. However, this combined countermeasure increased the circuit size by a factor of about 23, the critical path by a factor of about 4, the energy consumption by a factor of about 14 and it was still susceptible to attacks, requiring the equivalent of about 3.75 days of non-stop measurements. This result emphasizes the limits of balancing techniques in general, since even exhaustively balanced circuits are not sufficiently secure to avoid key extraction, not even when paired with first-order masking. Purely algorithmic solutions, like a combination of shuffling and TI achieve a better cost efficiency, but exhibit a much higher leakage in a detection scenario which may become problematic for device certification. In summary, it seems that hiding countermeasures, even rather expensive ones, do not create a sufficient noise level for their pairing with first-order masking to deliver a sufficiently high security level to fully prevent realistic attacks. It is either necessary to pair them with higher-order masking or to develop more effective hiding countermeasures altogether. In general, the quest for better solutions has to continue in this area. The results obtained from this work can certainly be helpful for the design of high-security cryptographic hardware in nanometer semiconductor technologies in the future. For the sake of completeness, we have also explored the impact of device aging on static power analysis attacks [KMM19]. While this is an aspect that adversaries need to be aware of, as it may change the leakage patterns of a circuit part for different inputs over time, it is not sufficiently effective to qualify as an SCA countermeasure [KMM19].

IC Prototyping. It is not only costly and time consuming to develop custom IC prototypes for public research purposes, but also difficult to obtain access to cutting-edge semiconductor

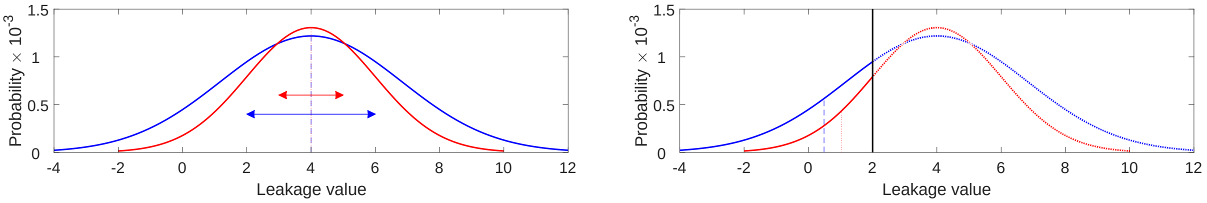
Table 1.1: This table contains implementation details about the four ASIC prototypes and references to the corresponding publications.

	90 nm	65 nm	40 nm	28 nm
IO voltage	2.5 V	2.5 V	2.5 V	1.8 V
Core voltage	1.2 V	1.2 V	1.1 V	0.9 V
Logic IOs	17	17	17	18
Power IOs	16	16	16	16
Metal layers	9	9	8	8
Cipher cores	27	27	57	54
Max. freq.	28.6 MHz	35.1 MHz	83.3 MHz	100 MHz
On-sil. size	1958 × 1958 µm	1942 × 1942 µm	1681 × 1681 µm	1379 × 1379 µm
Publications	[Moo19]	[KMM19, Moo19]	[Moo20, MWM21]	[MM21]

technologies. Usually, foundries will only give access to their leading and most innovative CMOS generations to customers that can afford a volume production in such technologies. Research institutes who require only prototypes in small quantities rarely get access to the most advanced CMOS technologies and often have to sign limiting Non-Disclosure Agreements (NDAs) to even work with the libraries related to newer process generations. Despite this fact, we have been able to develop, manufacture and analyze four different IC prototypes in 90 nm, 65 nm, 40 nm and 28 nm technology specifically for the investigations presented in this thesis. The layouts of these four devices are shown in Figure 1.1 and some implementation details are summarized in Table 1.3.1, together with references to the corresponding publications. For our investigation it was desirable to prototype integrated circuits in the very newest semiconductor manufacturing processes available and a great effort has been invested to achieve this. Yet, the access given to public research institutes is limited, and prototyping ASICs in technologies below 20 nm was not possible for the time being, despite the fact that commercial devices manufactured in 7 nm or even 5 nm nodes are already available on the consumer market (mostly high-end processors). However, according to industry sources, the majority of devices for the Internet of Things (IoT) is still manufactured in 65 nm or even larger technologies, while a slow transition to the 45 nm/40 nm node can be observed (at the time of writing this thesis). Thus, especially the 28 nm node still qualifies as a future generation for IoT devices, which is the device family most susceptible to physical attacks. Furthermore, to the best of our knowledge, no publication outside of the ones accumulated in this thesis has performed static power side-channel analysis on ASICs manufactured in technologies below the 65 nm node. Thus, our contributions to this field of research are unique and valuable to form a better understanding of this security threat.

1.3.2 Evaluation of Masked Implementations

In the second part of this thesis multiple contributions to the theoretical and practical evaluation of masked implementations are made. Masking is undoubtedly the most popular countermeasure to thwart SCA attacks and allows to prove the security of an implementation up to a chosen protection order in theoretical leakage security models. The technique is based on secret sharing, and splits each sensitive variable of an algorithm into a discrete number of shares



(a) Distinguishing the variances

(b) Distinguishing the means of slices

Figure 1.2: This figure illustrates the difference between (a) moment-based second-order analysis of two distributions and (b) first-order analysis of a chosen slice of distributions.

in such a way that only the combination of all of the shares contains information about the sensitive values. When each calculation or circuit part operates on a proper, i.e. incomplete, subset of the shares only and leaks information about its own intermediate values exclusively, independent of other calculations or circuit parts, higher-order SCA attacks are required to extract information. If, in addition to that, the operations on proper subsets of the shares take place at different points in time instead of in parallel, multivariate higher-order SCA attacks are required. To perform higher-order SCA attacks, adversaries need to be able to distinguish noisy leakage distributions based on tiny differences in their higher-order statistical moments. This procedure typically requires a remarkable amount of measurements to estimate the distributions in sufficient quality. Accordingly, the attack complexity tends to be high and increases exponentially in the security order until, ideally, a point is reached where the number of observations required becomes prohibitive for an adversary.

The publications condensed in Chapter 4 of this thesis deal with the evaluation of the security guarantees of masking schemes and masked implementations, which is known to be non-trivial, especially as the number of shares and the claimed security order increase. In the first publication an alternative method to analyze leakage measurements of masked implementations is proposed which does not rely on the estimation of statistical moments and therefore is not restricted to the analysis of a single statistical moment at a time [MM17]. The concept is fairly simple and illustrated in Figure 1.2. It is based on the observation that, instead of distinguishing full leakage distributions by their lowest informative statistical moment (here the variance, i.e., the second-order centered moment), it is possible to distinguish chosen slices of the distributions by their mean, i.e., the first-order statistical moment. It is demonstrated in [MM17] by simulations and practical experiments, that distinguishers based on the first-order moments of slices of the distributions can outperform distinguishers based on the higher-order moments of full distributions in certain situations. This is especially true for scenarios in which the estimation of higher-order moments is known to become suboptimal compared to other distinguishers, as for example when the noise level is too low or when leakages are distributed over multiple statistical moments [Sta18, MRSS18, Moo19, MWM21]. This technique has also been successfully applied in one of the publications from Chapter 3 to demonstrate that in static power attacks the noise level can be reduced to such an extent that moment-based distinguishers become suboptimal and lead to false negatives [Moo19].

The second publication summarized in Chapter 4 contains an in-depth analysis of the robust probing security of existing hardware-oriented masking schemes [MMSS19]. As a result of this analysis it is revealed that not a single multiplication gadget proposed in literature, which comes without a proof in the robust probing model, a variation of the traditional probing model aimed to capture physical defaults such as glitches, actually delivers local and compositional security in presence of glitches for arbitrary protection orders. In fact, a number of local and compositional flaws are exhibited which prevent the secure instantiation of the analyzed schemes at higher orders. These flaws are not only exhibited based on a theoretical analysis, it is also confirmed by empirical investigations that they indeed lead to a detectable security order reduction in real-world power measurements [MMSS19]. While this does not invalidate the innovative ideas behind these schemes, it does show that the engineering intuition which led to the successful design of gadgets secure at low orders benefits from a more formal analysis when moving to higher security orders. Thus, it is argued that proofs in the robust probing model are needed for novel masking gadgets to inspire trust in their local and compositional security, regardless of the protection order or the number of shares as the security parameters. Finally, an example is presented in order to answer the question whether solving probing security and composability independently is formally sufficient to solve them jointly. In fact, it is demonstrated that the combination of a glitch-resistant TI gadget with a composable (i.e., strong non-interfering) refresh gadget does not result in a circuit that is probing secure and composable in the presence of glitches. This result provides another argument for the abstractions considered in robust probing security and the need for proofs in this model. The Conference on Cryptographic Hardware and Embedded Systems (CHES) 2019 awarded this publication with its *best paper award.*

The third and last work covered in Chapter 4 introduces a novel leakage assessment methodology for SCA evaluations [MWM21]. Although technically the proposed method is not specific to masking or masked implementations in particular, it is clear that its runtime overhead compared to traditional approaches makes it primarily attractive for the detection of complex leakage patterns, like multivariate higher-order leakage for example. In detail, the work evaluates the use of deep learning as an eligible strategy for leakage assessment [MWM21]. Leakage assessment, also known as leakage detection, is a technique that has been introduced in order to answer the question whether input-dependent information can be detected in side-channel measurements collected from a device under test. In case this question is answered negatively (and no false negative occurs) the device is assumed to be sufficiently secure. Leakage assessment techniques are often employed by third-party evaluation labs to test the security of hardware devices in order to award security certificates. The general procedure is mostly based on distinguishing leakage distributions for two different input classes. Most commonly, either a group of traces acquired for a fixed input is compared to a group collected for random inputs (fixed-vs-random), or two groups gathered for different fixed inputs are used (fixed-vs-fixed). The typical mathematical tools used in this procedure are statistical hypothesis tests like the Welch's t-test [GJJR11, SM15] and the Pearson's χ^2-test [MRSS18]. These classical tests proved to work particularly well in the SCA context. Yet, they also come with some drawbacks. The main disadvantage is that the tests normally get applied to each point in a leakage trace individually and independently. Therefore, the procedure is mostly unable to detect leakages which are spread over multiple time samples, like multivariate or horizontal leakages. Also, as already mentioned for the moment-based analysis techniques before, the separation of statistical orders in the t-test has been shown to lead to false negatives in corner cases [Sta18, MRSS18, Moo19, MWM21]. Finally, the risk of false positives usually depends on the number of sample points in a trace unless specific corrections are applied. Since leakage assessment is primarily a statistical classification problem, it was tempting to investigate whether machine-learning-based approaches might be able to overcome some of the previously noted drawbacks. In order to evaluate this hypothesis, a leakage assessment methodology called Deep Learning Leakage Assessment (DL-LA) is introduced in [MWM21]. Simply put, a neural network is trained on a randomly interleaved sequence of labeled side-channel measurements corresponding to two groups collected for two distinct fixed inputs (fixed-vs-fixed). Then the trained network is supposed to classify further measurements from both groups without knowing the true labels. In case the classifier succeeds with a higher percentage of correct classifications than it could be achieved by a randomly guessing binary classifier, it is concluded that the side-channel measurements seen by the network during the training phase have revealed enough generalizable input-dependencies to confidently distinguish the two groups. It is demonstrated based on a total of nine different case studies from three different implementation platforms (FPGA, ASIC, μC) that this approach is able to detect first-order, higher-order, univariate, multivariate and horizontal leakages without requiring any trace-specific pre-processing or prior knowledge about the underlying implementation. The most outstanding advantage of the technique is clearly that the underlying network is free to combine as many points for the classification of the two groups as necessary. Thus, even in complex scenarios of purely multivariate or horizontal leakages, the traces can simply be fed as training data into the network without any pre-processing or manual selection of points. Despite the fact that this technique requires a significantly larger runtime than the conventional statistical hypothesis tests, it can be extremely valuable for the detection and evaluation of higher-order multivariate leakages of masked implementations (or other complex leakage forms), which normally require either an exhaustive search over all time offsets (becomes infeasible quickly), or expert-level knowledge and significant manual effort.

1.3.3 Low-Latency Block Ciphers

While the implementation size or area of block ciphers has received the lion's share of attention when attempting to design efficient symmetric cryptography over the past fifteen years, there are further optimization goals which can be at least as relevant for future applications. In Chapter 5 of this thesis we consider the minimum latency or execution time of hardware implementations of block ciphers as the focal point. With each new CMOS technology generation the price for manufacturing an integrated circuit consisting of the same number of transistors is decreasing, together with its size. Consequently, implementing a hardware co-processor on a chip for encrypting data with the Advanced Encryption Standard (AES) [oST01] is much more affordable today than it was twenty years ago. At the same time the propagation delay or latency of CMOS logic elements has decreased significantly. Therefore, processors can either operate at a higher frequency or the gate depth of arithmetic and logic operations that can be evaluated in a single clock cycle increases. Typically, a mixture of both is true. The decreasing relevance of area as a cost factor together with the increasing performance of semiconductor logic open the door to a new era of high-performance cryptography. Given that the trend in modern processor design is pointing towards more encrypted communication to thwart microarchitectural and other software-based attacks, the need for high-performance cryptographic primitives is stronger than ever. One common application is memory encryption where load and store instructions should encrypt and decrypt data on the fly without causing an overhead of many cycles per call. In order to execute cryptographic algorithms in one or a few clock cycles, the hardware implementations need to be unrolled. When unrolling a cipher implementation, the entire encryption (or decryption) function including all rounds is realized as one large combinatorial logic circuit without any memory elements incorporated. While this is trivial to achieve for any cryptographic algorithm, the challenge is to keep the resulting unrolled circuit sufficiently performant to enable its execution at a high clock frequency. The two publications covered in Chapter 5 deal with this problem.

The very first block cipher in public literature specifically designed for low-latency purposes was introduced at ASIACRYPT 2012 and is called `PRINCE` [BCG+12]. `PRINCE` is a 64-bit lightweight block cipher with a 128-bit key and 12 cipher rounds. Its latency is low compared to other block ciphers due to the use of a small, yet cryptographically strong, 4-bit S-box and a linear layer that keeps the number of rounds minimal. However, the main innovation in its design is a reflection property based on its symmetric construction around the middle which allows to encrypt and decrypt data with essentially the same circuit. This is a desirable feature for memory encryption applications on IoT microcontrollers, since only one instance has to be implemented to cover both encryption and decryption [BCG+12]. Attacks on `PRINCE` are claimed to require at least a time complexity of 2^{127-n} in case the data complexity is limited to 2^n. Years of public scrutiny by the cryptographic community have not found attacks with a lower complexity on full round `PRINCE`. The National Institute of Standards and Technology (NIST), however, demands more from lightweight primitives, namely a security level of 112 bits when the data complexity is below 2^{50} bytes (although this demand has been formulated with respect to Authenticated Encryption with Associated Data (AEAD) schemes) [NIS18]. Thus, the goal of the first publication covered in Chapter 5 was to investigate whether minor tweaks to the `PRINCE` block cipher can increase its security beyond the desired level without sacrificing too much of its performance [BEK+20]. Indeed, a carefully revised key-schedule proved to be sufficient to provide the required security goal while keeping (almost) all of the remaining design untouched. Thus, without changing the number of rounds or the particular round operations, a substantially higher security level is achieved at an extremely low overhead in all key categories, such as area, latency and energy. The resulting block cipher is called `PRINCEv2` [BEK+20]. In a detailed comparison of its resource consumption and performance to other (potential) low-latency block ciphers, including `MANTIS` [BJK+16], `QARMA` [Ava17] and `Midori` [BBI+15], `PRINCEv2` stands its ground (almost) as well as `PRINCE`. For legacy support it is also evaluated how a so-called `PRINCE+v2` version, which consolidates `PRINCE` and `PRINCEv2` in one circuit, stacks up against the competition.

Although `PRINCE` and `PRINCEv2` achieve impressive performance numbers when implemented as unrolled circuits in hardware, they are still designed for microcontrollers in resource-constrained environments and therefore have to remain small and energy efficient in addition to their low latency. In the second publication presented in Chapter 5 we attempt to go a step further and build a cipher design that focuses on high speed and security only [LMMR21]. As a result, we introduce `SPEEDY`, a family of ultra low-latency block ciphers dedicated to semi-custom, i.e., standard-cell-based, integrated circuit design. In order to construct such a cipher we first

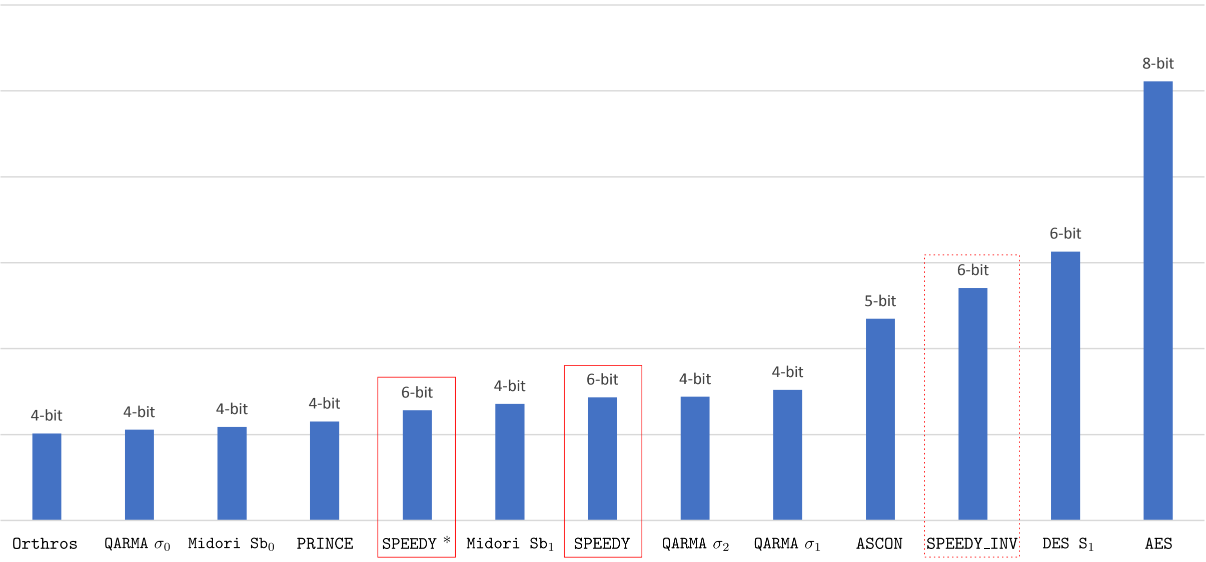


Figure 1.3: This bar graph compares the average normalized latency of different cryptographic S-boxes in hardware across 6 CMOS standard cell libraries.

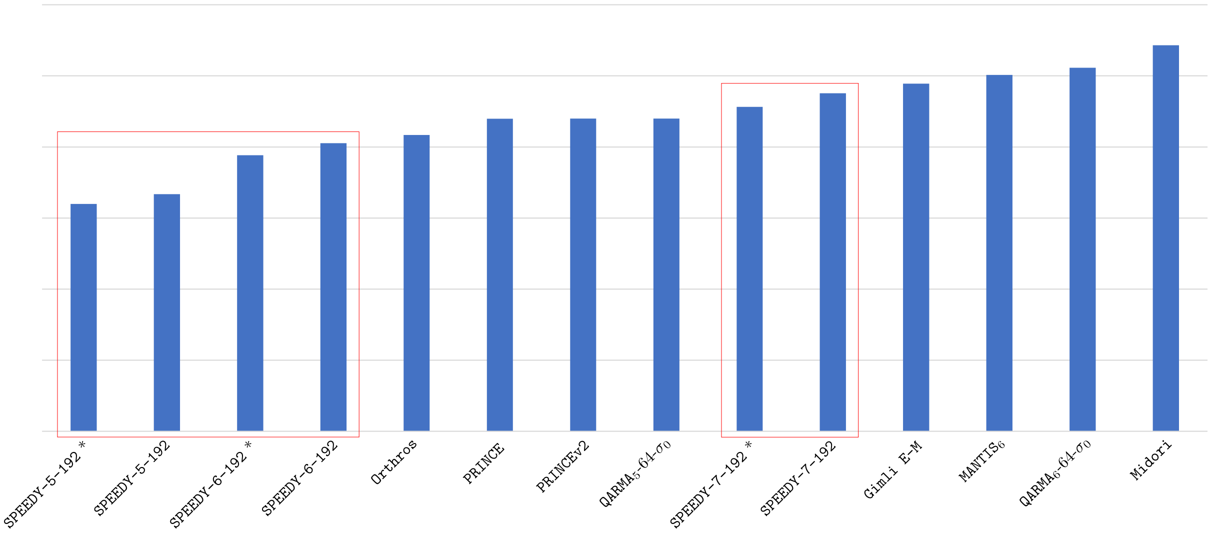


Figure 1.4: This bar graph compares the average normalized latency of different encryption functions in hardware across 6 CMOS standard cell libraries.

analyzed which type of CMOS logic gates, based on their transistor-level layout, is particularly suited for ultra low-latency encryption. Afterwards, we examined topologies for combinatorial circuits that benefit a low latency the most. Based on these insights we engineered a high-speed, cryptographically strong, 6-bit substitution box whose coordinate functions are realized as two-level NAND-gate trees. As shown in Figure 1.3, the latency of `SPEEDY`'s S-box compares favorably to other (low-latency) S-boxes from the literature as it even outperforms multiple 4-bit S-boxes. Figure 1.4 compares the latency of full hardware implementations of low-latency encryption functions across 6 different standard cell libraries. Here, `SPEEDY-r-192` denotes a `SPEEDY` variant with a block size of 192 bits and `r` rounds. Clearly, both `SPEEDY-5-192` and `SPEEDY-6-192` achieve a lower latency in hardware than any other encryption primitive while providing a significantly higher security level than competitors like `PRINCE` or `PRINCEv2`. In fact, attacks on `SPEEDY-5-192` are expected to require at least a time complexity of 2^{128} when the data complexity is limited to 2^{64}, while `SPEEDY-6-192` and `SPEEDY-7-192` are expected to offer 128-bit and full 192-bit security respectively without any restrictions to the data complexity. While `SPEEDY` can be instantiated with different block and key sizes, the default is 192 bits as it constitutes the least common multiple of 6 (the S-box's width) and 64 (the common data width in modern CPUs). We believe that `SPEEDY` is a great choice for all applications where encryption speed and security are the primary goals. As previously introduced, the expected target applications are found in high-end processor designs to enable secure cache architectures, memory encryption, pointer authentication and other protection mechanisms required in future generations of CPUs. Finally, with respect to the physical security of unrolled cryptographic cipher implementations we have shown in [Moo20] that the nature of unrolling intrinsically delivers impressive resistance against passive side-channel attacks when combined with simple random reset (or pre-charge) strategies, while being comparably cheap and simple to implement. In this regard it is clear that low latency cryptography not only enables high performance applications, it also benefits the security of devices against physical adversaries, especially in advanced semiconductor technologies.

Chapter 2

Background

This background chapter introduces the fundamental scientific concepts that are necessary to understand the results presented in this thesis and revisits the relevant definitions. The focus of this chapter is on cryptography in general, cryptographic hardware in particular and the physical security of said cryptographic hardware devices specifically. Additionally, the measurement tools and techniques to be used in experimental investigations of this kind are described.

Contents of this Chapter

2.1 Cryptography 21
2.2 Cryptographic Hardware 23
2.3 Physical Security of Cryptographic Devices 27
2.4 Measurement Techniques and Equipment 41

2.1 Cryptography

Cryptography is the art of secure communication in the presence of adversaries [Riv90]. The most elementary use case of cryptography is the secret transmission of messages between two distant parties over an insecure communication channel. While both the sender and the receiver should be able to read the communicated messages, all other entities who might gain possession of the transferred data during its transportation should not be able to decode the content. This can be achieved with the help of cryptographic algorithms. A cryptographic algorithm in its most simple form is a mathematical function that receives two input parameters, first a message to be encoded, also called the plaintext, and second a secret cryptographic key [MOP07]. In a process that is called encryption, these two parameters are then mapped to an output, also known as the ciphertext. The decryption process performs the inverse direction, namely mapping the ciphertext and the key to the plaintext. The most crucial requirement for a cryptographic algorithm is that, without knowledge of the secret key, this mapping cannot be performed, not even partially. According to Kerckhoffs's principle all details about the cryptographic algorithms should be public knowledge and only the key has to remain secret in order to protect the confidentiality of encrypted information. As cryptography matured over the years, it has assumed many further goals than just the confidentiality of messages. Entity and data authentication, integrity, privacy and non-repudiation are other common security goals today. Nowadays it is also required to distinguish between symmetric cryptography, which is also known as secret key cryptography, and asymmetric cryptography, which is also called public key cryptography.

2.1.1 Symmetric Cryptography

Symmetric cryptography is the most traditional form of cryptography and uses identical keys for encryption and decryption of data. Thus, two communicating parties have to be in possession of a common secret key. To distribute such a shared key, a secure channel is required. The most popular type of cryptographic algorithm for encrypting messages between two parties is a so-called block cipher. Block ciphers encrypt data in blocks of a fixed length. A well-known example is the Advanced Encryption Standard (AES) [oST01], also known as Rijndael algorithm which has been standardized by the National Institute of Standards and Technology (NIST) in 2001. AES can be used with a 128-bit, 192-bit or 256-bit key. A block cipher is supposed to be indistinguishable from a family of random permutations for computationally-bounded adversaries. If an attacker succeeds in distinguishing the cipher from a set of random permutations with a lower complexity than that of an exhaustive key search, the cipher can be considered broken (if the claim is full key length security). The exhaustive search through all possible keys, also called a brute-force attack, is an adversarial strategy that can be applied in virtually any scenario. However, this type of attack is computationally heavy. More concretely, its complexity increases exponentially in the effective key length. For a 128-bit key for example, 2^{128} possible candidates exist. If an adversary is able to test around 1 million keys per second, it would take more than 10^{25} years to search through all possibilities. If the adversary can test 1 million keys each *picosecond*, the attack would still take more than 10^{13} years. In comparison, according to current estimations by the National Aeronautics and Space Administration (NASA) the universe is only about 10^{10} years old. Thus, with current computing technologies it is hardly feasible to perform a brute-force attack on a 128-bit key, much less for key sizes of 192 and 256 bits.

2.1.2 Asymmetric Cryptography

In asymmetric or public key cryptography each user possesses a pair of keys consisting of a private key and a public key. As the name suggests, the public key is openly distributed to any potential communication partner, whereas the private key remains a secret. The general concept requires that data which has been encrypted with a user's public key can only be decrypted with that user's private key. In consequence, everyone can use the public key of a particular entity or person to encrypt messages specifically for this one individual who is in possession of the matching private key. The enormous benefit of asymmetric cryptography is that it does not require the communication partners to already have established a shared secret key. Instead, information can be exchanged confidentially without being in possession of identical keys. In public key cryptography it is common practice to base the security of a cryptosystem on the assumption that a certain problem is hard to solve. Under this hardness assumption the security properties of a scheme may be formally proven. Yet, the validity of the assumption itself may not be proven, it can merely be disproven. Consequently, the hardness of the underlying problem may only be assumed for as long as no algorithm is known which efficiently solves it. The most well-known example of an asymmetric cryptosystem is the Rivest-Shamir-Adleman (RSA) cryptosystem [RSA78] which relies on the hardness of the RSA problem. Yet, the RSA problem may be solved efficiently when an algorithm for polynomial-time factorization of large numbers is found. Up to now, such an algorithm is not known in classical computing, only in quantum computing. Today, there exist several asymmetric cryptosystems which rely on problems that are currently believed to be hard even in the realm of quantum computing. Time will tell whether this belief is justified. The clear disadvantage of public key cryptography is that the required algorithms are not nearly as efficient as the ones used in secret key cryptography. Hence, it is common practice to use a combination between both forms of cryptography, for example to exchange a shared key via a public key algorithm and then communicate using that shared key via secret key cryptography.

2.2 Cryptographic Hardware[1]

Implementing cryptography on computing devices is either done in software or in hardware. Although the term *cryptographic hardware* is sometimes used to describe any kind of hardware device that is running cryptographic algorithms, including microcontrollers that execute a cipher as a software program, this chapter is only concerned with cryptographic primitives that are actually implemented as hardware circuits. In contrast to a software implementation consisting of a series of instructions to be executed on a general-purpose processor, a hardware implementation is a dedicated logic circuit built specifically for evaluating one explicit algorithm on the input data. Typically, this form of dedicated circuitry can outperform a corresponding software implementation by multiple orders of magnitude regarding its execution speed due to its specialization and the higher degree of parallelism. Hence, secure and efficient cryptographic hardware has gained more and more importance over the last decades.

2.2.1 Digital CMOS Integrated Circuits

The vast majority of computer chips deployed over the past decades has been manufactured using Complementary Metal–Oxide–Semiconductor (CMOS) technology. To manufacture an Integrated Circuit (IC) in CMOS technology, a silicon wafer is subjected to multiple cycles of chemical and photolithographic treatments in order to form the digital elements and interconnecting wires that constitute the created hardware design [RCN04]. Photolithography uses light to transfer a geometrical pattern from an optical mask to a substrate. In this way the hardware design is transcribed to the silicon fabric. The optical masks used in this process form the central interface between the design created with an Electronic Design Automation (EDA) software and the specifics of the manufacturing process [RCN04]. The continuous demand for down-scaling the minimum feature size in integrated circuits is accompanied by significant challenges for the manufacturing process. When the dimensions of elements to be transferred from an optical mask to the semiconductor material fall below the wavelength of the optical light used in the process it becomes increasingly difficult to achieve the required resolution and accuracy. Yet, the semiconductor industry, as the driving force behind the general electronics industry, puts a large amount of resources into the research and development of new and innovative manufacturing technologies to keep the amount of transistors that can be fabricated onto a chip of the same dimensions steadily growing.

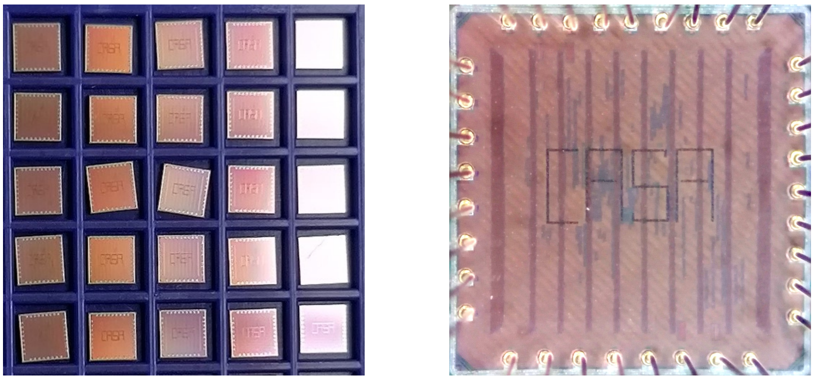
---

[1] This section contains excerpts of our publications [Moo20] and [LMMR21].

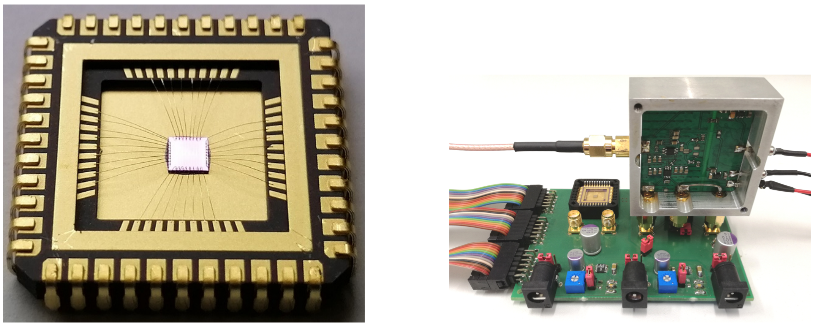
CMOS technology requires that both, n-channel (NMOS) and p-channel (PMOS) transistors, can be manufactured in the same semiconductor material. Using these devices it is then possible to built elementary logic and memory cells. In detail, a static CMOS gate is constructed by combining a pull-up with a pull-down network. The pull-up network, as the name suggests, is responsible for pulling the output of the gate up to the supply voltage `VDD` whenever the Boolean function should result in a logical '1'. The pull-down network, analogously, is responsible for pulling the output down to `GND` whenever the Boolean function should output a logical '0'. The networks are built in a mutually exclusive manner such that only one of them is conductive for each combination of input signals [RCN04]. The pull-up networks are built from PMOS devices whereas the pull-down networks are built from NMOS devices. While PMOS devices can be understood as switches that conduct current between their drain and source terminals whenever their gate voltage is low, NMOS devices conduct current between the terminals whenever their gate voltage is high. For the opposite gate voltages the transistors are in a high-resistance state. The assignment of PMOS transistors to pull-up networks and NMOS to pull-down networks originates from the fact that PMOS devices cannot produce so-called *strong zeros*, while NMOS devices cannot produce *strong ones* [RCN04]. In consequence, static CMOS gates with a single stage are naturally inverting by design. Non-inverting Boolean functions require at least two stages of pull-up and pull-down networks. Thus, logic cells like a NOT (inverter), NAND or NOR gate can be realized very naturally in CMOS technology. Since this group of gates is functionally complete (actually a NAND or NOR gate alone would already be), CMOS logic is able to express any logic function or truth table and therefore can be used to manufacture integrated circuits for any purpose and application.

2.2.2 Application-Specific Integrated Circuits (ASICs)

Application-Specific Integrated Circuits (ASICs) are the opposite of a general-purpose processor. An ASIC realizes a highly customized functionality and is heavily optimized for a specific problem instead of focusing on general-purpose versatility. Therefore, ASICs are typically the most powerful and performance-driven option to implement a function or an algorithm in hardware. However, due to the static circuit structure there is no possibility to change, update or repair the implemented function or application retrospectively. Once the chip is manufactured it cannot easily be altered anymore. Digital ASICs are typically developed using Electronic Design Automation (EDA) tools in a semi-custom design process. Semi-custom means that pre-defined building blocks like standard cells and memories are instantiated and connected to map the functionality of the high-level design to the low-level hardware. Full-custom ASIC designs are created when analog functionality needs to be integrated into the chips or when transistor-level structures need to be realized for which no pre-defined building blocks exist. A cryptographic primitive can unfold its full potential with respect to performance and efficiency when realized in an advanced IC technology node as a semi-custom design. Full-custom design is rarely required for cryptographic implementations. While full ASIC designs developed exclusively for cryptographic algorithms aside from research purposes and bitcoin mining might be rare, it is common practice nowadays to integrate dedicated hardware co-processors for established cryptographic algorithms like the AES into CPUs and Systems-on-Chips (SoCs). The reasoning behind this is the same as detailed before, namely the interest to trade some versatility for more performance and efficiency. Since experimental results involving ASIC prototypes



(a) Tray of naked dies (b) Single bonded die



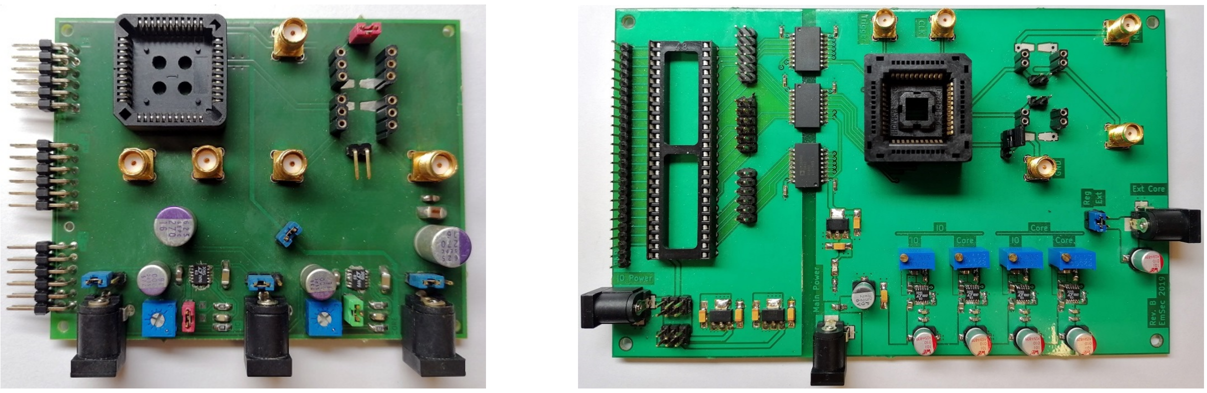
(c) Packaged die (d) Packaged die on a PCB

Figure 2.1: These photos show some of the ASICs we developed and manufactured for this thesis from the naked dies to the packaged and mounted chips.

are a fundamental part of this thesis we provide some exemplary photos of our prototypes in Figure 2.1 from the naked dies to the packaged and mounted chips.

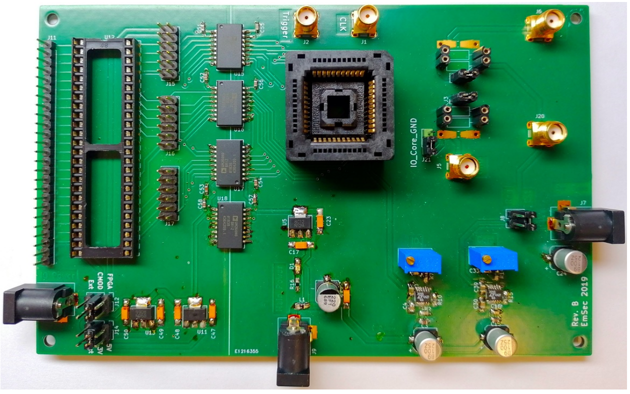
2.2.3 Field Programmable Gate Arrays (FPGAs)

Field-Programmable Gate Arrays (FPGAs) are integrated circuits, which in contrast to ASICs, can be reconfigured. In this sense FPGAs close the gap between ASICs and general-purpose processors. FPGAs consist of an array of programmable logic blocks and programmable interconnects which can be used to realize essentially any hardware function. Also, it is possible to change the configuration of the programmable fabric whenever required. Thus, FPGAs combine some of the advantages of ASICs, like high performance, structural flexibility and parallelism and those of general-purpose processors, namely versatility and reusability. Yet, as a result of the reconfigurability, FPGAs cannot provide the same performance and efficiency as ASICs. To illustrate the discrepancy between the two hardware platforms in more detail, we refer to the so-called *cost of programmability* [KR07]. According to the seminal work by Kuon *et al.* [KR07], a fully combinatorial representation of a function requires about 35 times as much area on an FPGA as on a standard-cell-based ASIC, due to the structure of the programmable fabric. Clearly, such a significant increase in the number of gates involved in the computation leads to



(a) Board for 90 nm and 65 nm ASICs

(b) Board for 40 nm ASIC



(c) Board for 28 nm ASIC

Figure 2.2: These photos show the PCBs we have developed as measurement boards for the different ASIC prototypes.

a much higher power consumption and delay as well. In particular, the authors observed that regular logic designs are more than 4 times slower on an FPGA, while consuming 14 times as much dynamic power as an equivalent ASIC design in the 90 nm technology considered as a reference point [KR07].

2.2.4 Printed Circuit Boards (PCBs)

Printed Circuit Boards (PCBs) are used to mechanically support electronic components by providing soldering footprints to fix them to designated locations. They also provide reliable electrical connections between the components using traces, planes and other features etched from copper sheets laminated onto a non-conductive substrate. PCBs are used in almost all electronic products today in order to connect ICs and to provide peripherals to interact with them. Figure 2.2 shows the different PCBs that have been developed to perform side-channel measurements on the ASIC prototypes developed in this thesis. In some of our works, commercially offered side-channel evaluation boards, like the SAKURA-G [Sak] or the SASEBO-R have been used.

2.3 Physical Security of Cryptographic Devices[2]

Physical security becomes a concern whenever cryptography is deployed in a field that puts the hardware responsible for executing cryptographic algorithms in a potentially hostile environment. Years of academic and industrial research have revealed the unpleasant truth that no universal solution exists to protect cryptographic devices from key recovery attacks when they are forced to operate under permanent physical exposure to untrusted parties. Although significant advances have been made in developing dedicated protection mechanisms against this threat, there is still neither one guaranteeing full resistance, nor any that is universally applicable to all hardware and software implementations alike (without significant adjustments). Thus, the physical security of cryptographic hardware is still a fundamental concern for many security-critical applications and infrastructures and a growing area of research.

2.3.1 Side-Channel Analysis

Side-channel analysis is the comprehensive term for techniques that are used to extract sensitive information from computing devices by exploiting the leakage exhibited through so-called side channels. The most common targets for side-channel analysis attacks are cryptographic devices. Side-channel analysis is categorized as a passive and non-invasive attack, which means that the device under test is operated according to its specifications without any disturbance of the computation or modification of the target. Thus, any exploitable information is solely learned by observing the physical characteristics or dissipation during the execution of sensitive algorithms. It is important to note that side-channel attacks do not target a cryptographic algorithm itself, but rather a concrete physical instance of that algorithm. Many different sources of side-channel leakage exist in common hardware devices. Besides timing attacks [Koc96], primarily the exploitation of the power consumption [KJJ99] and the electromagnetic radiation [GMO01] proved to be effective sources of information for adversaries. However, also thermal [HS14], acoustic [GST14] and optical [SNK$^+$12] attacks have been demonstrated in literature. Clearly, any adversary, who is capable of measuring the physical characteristics of a cryptographic device during the execution of an algorithm, has access to substantially more information than just the inputs and outputs, and therefore does not operate in the black-box model. In fact, those physical characteristics can be correlated to intermediate values of the computation, and cryptographic algorithms are usually not developed to withstand attacks which make use of their intermediate results. An adversary is successful, if the analysis yields a sufficient entropy loss of the secret key used in the cryptographic device to efficiently determine the full key through exhaustive testing (brute force) of the remaining candidates. Obviously, side-channel attacks which rely on measuring the physical emissions of an implementation, in contrast to, for example, its often remotely available execution time, are primarily a concern for devices that an adversary can obtain physical access to.

2.3.2 Fundamental Limits and Information Leakage of Computation

The fundamental physical limits of computation dictate what can and what cannot be achieved by computing machines [BL85]. It has been shown many years ago, for example, that the

---

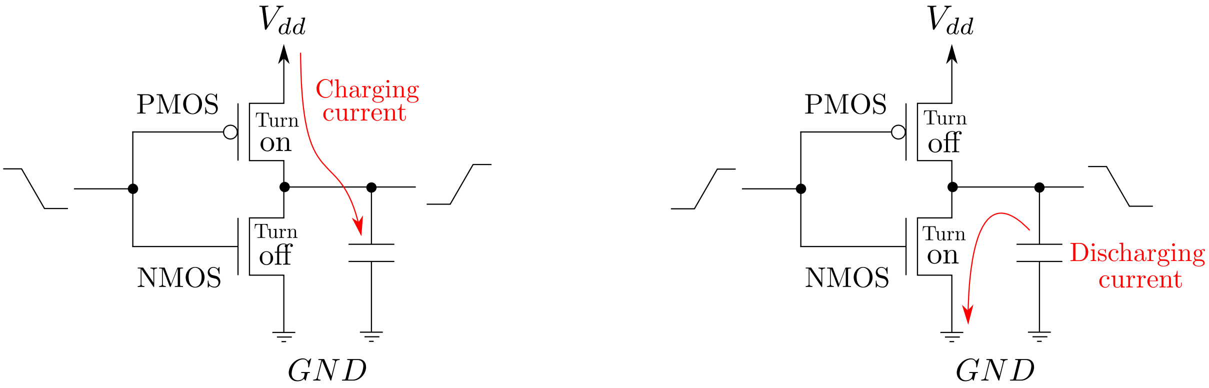
[2] This section contains excerpts of our publications [Moo19], [Moo20] and [MWM21].

majority of classical logic gates, being the essential building blocks of computing technology, cannot be evaluated without a certain amount of dissipation [Lan61, BL85]. This statement holds, regardless of the underlying device technology. In particular, state transitions performed by conventional logic operations are often of an irreversible nature, which means that information is discarded because two or more distinct logical states have a single successor [Ben03]. In a digital two-input AND gate, for example, the input combinations (0,0), (0,1), (1,0) are all mapped to output (0) and thus cannot be reversed. Such transitions *must* be accompanied by a loss of energy to the environment. This has been manifested in Landauer's principle [Lan61] and is a direct implication of the second law of thermodynamics [Llo00]. Whether information is discarded by a logic operation (i.e., an irreversible transition takes place) or not and therefore whether it is dissipated to the environment, depends on the processed data [BL85]. Hence, as a matter of fact, computation, as it is currently carried out, does not only imply energy dissipation, but also leakage of information through physical side channels – entirely independent of any technological details. Logical reversibility can indeed be achieved by specialized and more complex logic gates, bearing the potential to eventually evade the lower bound of Landau [BL85, Llo00]. However, a suitable device technology for nearly physical reversibility needs yet to be developed. In practice, any computing device will dissipate at least some amount of data-dependent energy [Llo00].

This discussion focuses on transitional leakages occurring during an active computation process exclusively. From a thermodynamic standpoint this is sufficient, since there is no necessity for dissipation without a transition of states. In other words, it should be possible to pause a physical computation process and to hold a stable state, keeping sensitive intermediates enclosed in the circuit, without being doomed to an undesired disclosure of information. This is in fact exactly what is described by the famous *only computation leaks* paradigm, introduced in [MR04]. The authors formulate the assumption that *"computation, and only computation, leaks information"*, implying that *"there is no information leakage in the absence of computation"*. Yet, as previous works regarding the information leakage of CMOS devices in stable states have shown [Mor14, PSKM15], this assumption does no longer approximate the behavior of current semiconductor technologies to a sufficient degree.

2.3.3 Power Consumption of CMOS Devices

Modern circuit technologies need to achieve many different objectives in parallel, with energy efficiency being only one of them. High performance, reliability, manufacturability and cost effectiveness are fundamental concerns, besides a number of further considerations depending on the desired area of application. Thus, not all effort can be dedicated to the reduction of the energy consumption and it can be observed that technologies suitable for Very-Large-Scale Integration (VLSI) in practice usually dissipate significantly more energy than what is demanded by the fundamental physical limits. CMOS logic gates consume a relatively large data-dependent current during the state transition from one output value to another, due to the associated charging and discharging of output capacitances. This is illustrated in Figure 2.3 exemparily for a CMOS inverter gate. CMOS gates also consume a (less data-dependent) short-circuit current during any output transition due to the short period of time where both, the pull-up and the pull-down network, are conducting. This effect is depicted in Figure 2.4.



(a) 1 → 0 transition (b) 0 → 1 transition

Figure 2.3: This figure illustrates the charging and discharging currents in a CMOS inverter gate.

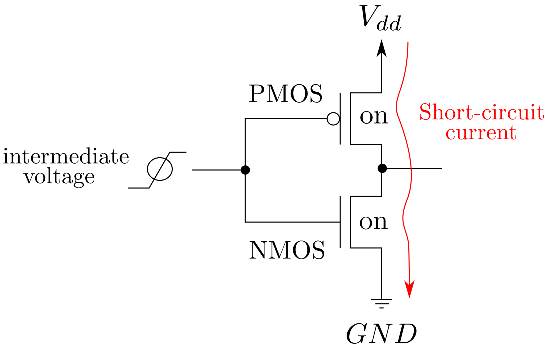
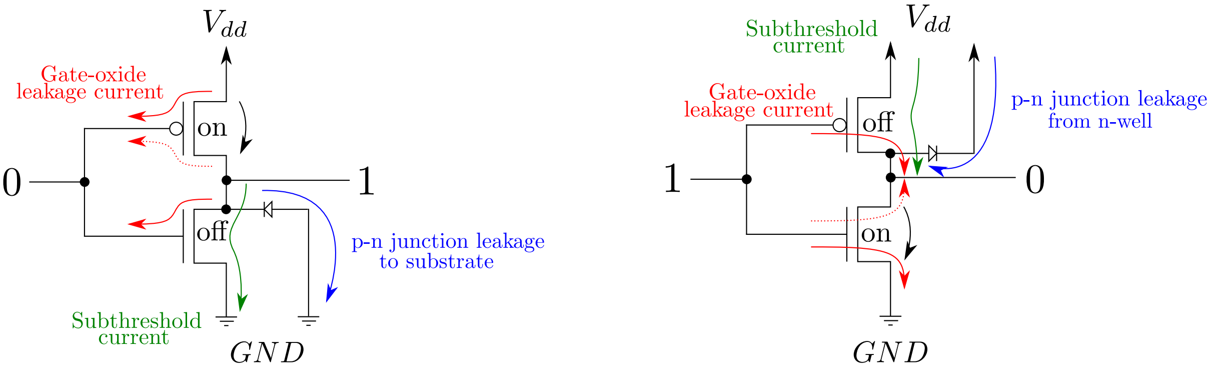


Figure 2.4: This figure illustrates the short-circuit current in a CMOS inverter gate.

Traditionally, these dynamic currents are assumed to be the predominant cause for both, energy dissipation and information leakage of hardware circuits. However, over the years, physical characteristics and electrical specifications of transistors have changed significantly. To comply with Moore's famous law [Moo65], the dimensions of Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs) have faced an aggressive scaling process in order to achieve the desired and predicted exponential increase over time in the number of transistors that can be fabricated on a single integrated circuit of a given size. In the attempt to uphold this scaling factor, valuable properties of the technology were sacrificed, as for example the negligible current consumption in idle states.

Initially, CMOS logic has been constructed in such a way that, given the idealized model of a transistor holds, no current should be consumed in any stable state. In particular, the individual logic gates are composed of a pull-up network, which establishes a conductive path between the gate output and VDD when activated, and a matching pull-down network, which is able to create a conductive path between the output and VSS (GND) respectively. For any combination of stable input signals, only one of the two networks is allowed to be active (i.e., switched on), while



V_{dd}
Gate-oxide leakage current
on
0
1
off
p-n junction leakage to substrate
Subthreshold current
GND
Subthreshold current
V_{dd}
Gate-oxide leakage current
off
p-n junction leakage from n-well
1
0
on
GND

(a) Input 0

(b) Input 1

Figure 2.5: This figure illustrates the leakage currents occurring in a CMOS inverter gate for static input signals.

the other one, and therefore at least one transistor in any path between VDD and VSS, should be switched off. Conceptually, this allows for a negligible power consumption in stable states, since for no static input combination a conductive path is formed across the power supply. Yet, by down-scaling the physical feature size, transistors progressively deviate from the idealized model. To be more precise, a nanoscale MOSFET does not resemble an ideal switch anymore but tolerates a significant off-current to flow between its terminals, even in a supposedly high resistance state. A depiction of the different types of leakage currents occurring in a CMOS inverter gate is shown in Figure 2.5. This behavior is a serious concern for hardware designers, as these so-called leakage currents consume a steadily increasing part of the power budget of modern ICs. It also leads to the situation that the global power consumption of circuits cannot be reduced to the amount of active computation anymore, measured by the number of gate toggles for example. Instead, even without any active computation (i.e., in an idle state) a significant amount of energy, proportional to the number of powered logic cells in the circuit, is consumed, independent of whether those cells are actively fed with input data or not.

Due to the structure of digital CMOS standard cells it can be observed that their individual cumulative off-current is highly determined by the composition and type of active and inactive transistors across the power supply path, which in turn directly depends on the applied input signals to the cell. In more detail, the magnitude of the leakage current exhibited by a CMOS logic cell depends on the type and formation of switched-off MOSFETs in the path between VDD and GND and the different electric potentials across them. For example, consider the simple CMOS two-input NAND gate in Figure 2.6. Here, when replacing all active (i.e., conducting) transistors with ideal wires, a number of different formations of inactive transistors can be observed, which depend on the input. Clearly, these input combinations will lead to different leakage currents exhibited by the NAND gate. Comparing the two cases (A=0, B=0) and (A=0, B=1), for example, it is obvious that the former, where two switched-off NMOS transistors are connected in series, has a significantly smaller leakage current than the latter. Connecting inactive transistors in series causes a so-called stacking effect. This effect reduces the current flowing through a stack of two inactive transistors by one order of magnitude compared to

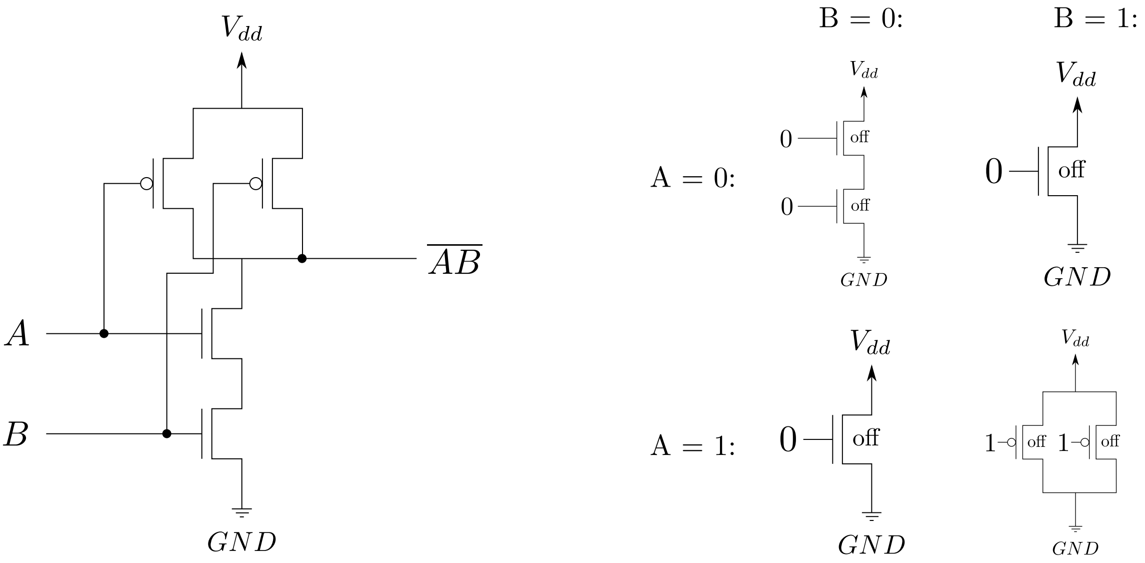


Figure 2.6: Two-input CMOS NAND gate (left) and formation of inactive transistors across the power supply path for different inputs (right), when replacing conducting transistors with ideal wires.

a single inactive one [RMM03]. For this reason, transistor stacking is also used as a leakage current mitigation technique. The largest current is leaked by a CMOS NAND gate when the input combination (A=1, B=1) is applied. In that case two inactive PMOS transistors, connected in parallel, are present between VDD and GND, whose individual leakage currents accumulate. In Figure 2.6 active transistors are replaced by ideal wires as a simplification. For that reason the two cases (A=0, B=1) and (A=1, B=0) look identical in this example. In reality, these cases would show significantly different leakage currents, due to the different electric potentials across them. In this example, considering the case (A=1, B=0) the drain of the switched-off NMOS is pulled up by a switched-on NMOS transistor (instead of a PMOS) and therefore the voltage at the drain never reaches VDD since NMOS transistors cannot produce strong ones [RCN04]. Thus, in reality the data dependency is even stronger than in the simplified scenario. In summary, the static power consumption of CMOS logic is substantially data dependent. One common leakage reduction technique is therefore to assign primarily those input signal combinations to the individual logic cells when the device is in idle which cause the least amount of leakage current.

2.3.4 Power Analysis Attacks

Power analysis is arguably the most popular form of side-channel analysis and relies on exploiting the data and operation dependencies in the power consumption of computing devices. This type of attack has been first reported in public literature in 1999 [KJJ99]. Among all possible side channels, the power consumption of cryptographic devices is usually the most simple to access and one of the most informative to obtain. Any software or hardware implementation executed on an integrated circuit will be vulnerable to power analysis attacks, unless equipped with dedicated countermeasures against this type of analysis, as the power consumption of CMOS hardware appears to be data- and operation-dependent.

Dynamic Power Side-Channel Analysis. In dynamic power side-channel analysis, the instantaneous power consumption of computing devices is correlated to internally executed operations or processed data. The most basic form of dynamic power analysis is known as Simple Power Analysis (SPA) [KJJ99]. SPA requires the direct interpretation of power consumption measurements collected during cryptographic operations to learn information about secret internals. If for example the order or timing of operations with a distinguishable power consumption footprint depends conditionally on sensitive values, the adversary can analyze the sequence of operations in a recorded trace in order to gather information about such secrets. A common example of a potentially vulnerable target is the straightforward implementation of the square-and-multiply algorithm for modular exponentiation [KJJ99, MOP07]. When this algorithm is implemented without protection against SPA, adversaries may be able to extract all bits of the secret exponent just from a single measurement by distinguishing the power consumption caused by the squaring from the multiplication. Preventing SPA is similar to preventing timing side-channel attacks. In particular, any conditional execution of instructions based on sensitive information should be avoided at all costs. In contrast to SPA, so-called Differential Power Analysis (DPA) [KJJ99] is able to exploit even tiny differences in the power consumption based on a statistical analysis. This process includes guessing a part of the secret key to compute an intermediate value that depends (non-linearly at best) on this key part and a known or chosen part of the input. Depending on a chosen bit of that intermediate value the measured side-channel traces are categorized into two groups. If the difference between the means of the two groups is significantly larger for one key candidate than for all others at one or multiple points in the trace, there is a high chance that the correct key candidate was identified. In order to perform a similar analysis on the whole intermediate value at once, instead of a single chosen bit, Correlation Power Analysis (CPA) [BCO04] has been introduced. Here, a hypothetical model has to be assumed in order to estimate the leakage. Common hypothetical leakage models for power analysis attacks are the Hamming weight of a value (mostly for software implementations) and the Hamming distance between two consecutively processed values (mostly for hardware implementations). Since the selection of a sound model is not always trivial, collision-based SCA attacks have been proposed to remove this requirement [MME10, MS16].

Static Power Side-Channel Analysis. For Static Power Side-Channel Analysis (SPSCA)[3], many of the same concepts and methods as for dynamic power attacks can be applied, including DPA, CPA and collision attacks. However, the physical nature of static power SCA is very different from dynamic power analysis, as it does not exploit a momentary transitional effect that can be observed for a finite period of time only. Instead, it is based on observing a static phenomenon that can be quantified for as long as no transition occurs in the targeted circuit

---

[3] Various different notations have been introduced for *static power side-channel analysis* in the literature, e.g. *static power analysis* [XH17] and *leakage power analysis* [AGST09]. However, since the term *static power analysis* is already an established and unrelated expression in the EDA community and since *leakage* is a frequently used term with a mostly unrelated meaning in the side-channel literature, we suggest the (admittedly quite lengthy) notation of *static power side-channel analysis* in this work and use *static power SCA* and *SPSCA* as its abbreviations.

part. The direct relation between the static power consumption of a CMOS standard cell and its inputs leads to the inconvenient and, from a side-channel perspective, highly alarming situation that on advanced CMOS hardware it is neither possible to actively process data (dynamically), nor to passively hold data (statically) in a circuit (e.g., in a flip-flop between consecutive clock cycles), without leaking information about those values via physical side-channels. While the inability to compute without dissipating information-bearing energy amounts is a direct implication of the laws of thermodynamics (at least when considering standard logic gates due to the associated irreversible state transitions), leaking information in stable states (i.e., without any transition) is not necessary from a physical viewpoint and purely caused by technology-specific defaults which are further amplified through scaling effects.

Following a number of simulation-based investigations, the first experimental attempt to quantify the impact of this security threat based on real-world power measurements was published at CHES 2014 [Mor14]. This work also provides a first basic technology comparison, as the examined FPGA families were manufactured in three different process technologies. The first successful static power attacks on an ASIC are presented in [PSKM15]. Before the start of this thesis, it had also been demonstrated, both in simulations and in practice, that various established countermeasures against dynamic power side-channel analysis can be significantly less effective against the exploitation of the static currents [LB08, ABD$^+$14, ABST14, IM14, Mor14, BST16, BBM$^+$16, MMR17]. The first successful (higher-order) static power side-channel attack on a masked implementation has been performed in [Mor14]. It was also suggested that masking schemes with a sequential manipulation of the shares (typical in software) might be in danger when an exploitation of the leakage currents is possible, since the shares may be leaked in a univariate fashion through the static power, making multivariate attacks unnecessary and potentially reducing the effective noise level. Further, the authors of [PSKM15] suggest that in case of an adversary obtaining full control over the clock signal (which is also assumed in [Mor14] and previous works) it is possible to average the static power consumption over an arbitrary time period, which allows to eliminate several sources of noise entirely. It was experimentally verified in [MMR17] that this averaging technique in static power SCA attacks (with obtained clock control) indeed allows to reduce the noise level significantly. Furthermore, in [MMR17] a successful higher-order static power attack is performed which requires fewer traces to be successful than a corresponding dynamic power analysis attack on the same target. In fact, this result shows that without dedicated countermeasures, it is harder to assure a sufficient noise level against adversaries that measure static currents than against those who rely on measuring the dynamic switching activity of a chip. Such an observation goes hand in hand with the intuition that any *static* physical effect should, by definition, be easier to quantify with a high precision (i.e., low noise) than a corresponding transitional one, simply because static phenomena are persistent and not limited to a finite period of time.

2.3.5 Countermeasures

Developing effective countermeasures against side-channel analysis attacks is not trivial and requires expertise in a number of different sub-disciplines such as cryptography, microelectronics, statistics and measurement technology [MOP07]. For more than twenty years now the cryptographic research community is investigating how to perform cryptographic operations securely in the presence of physical adversaries while staying within reasonable efficiency bounds. Up to the present day, multiple hundreds of works have been published in this field of research, proposing solutions ranging from algorithmic to transistor-level approaches, and there is no end in sight. Existing countermeasures still show room for improvement, either in terms of the security level provided (both formally and practically), or in terms of their implementation efficiency, while new techniques are explored and evaluated in a continuous manner. It is clearly beyond the scope of this chapter to introduce this subject in sufficient depth to give a full overview of the current state of the art. Thus, we will limit ourselves to a short look at the taxonomy of side-channel countermeasures and to the introduction of the general concepts from a high-level perspective. While it can be hard to classify SCA countermeasures precisely into clearly disjoint groups, there are two broad categories which are universally recognized, namely hiding and masking [MOP07].

Hiding. Hiding schemes are based on the idea of decreasing the signal-to-noise ratio (SNR) in the measurements that adversaries are able to obtain. In fact it is attempted to break the link between the available side-channel information and the processed operations and data. However, internally the identical operations are executed (sometimes shifted in time domain) and the same data is processed. Hiding schemes can be divided into two subgroups, namely *equalization* and *randomization*. *Equalization* techniques try to reduce the SNR by decreasing the exploitable signal. This goal is typically addressed by attempting to perform operations in such a way that their physical side-channel leakage, like their power consumption or electromagnetic radiation, is as constant as possible, independent of the processed data. In order to approach this independence, the power consumption needs to be filtered or balanced. These methods affect the amplitude dimension and are able to (partially) remove the exploitable part of the power consumption. Popular examples for this type of hiding countermeasure are Dual-Rail Logic styles [MOP07] such as Wave Dynamic Differential Logic (WDDL) [TV04]. *Randomization* techniques try to reduce the SNR by increasing the noise level. This can be achieved in amplitude dimension by simply employing noise engines which generate additional noise on demand in parallel to the cryptographic operation that should be protected. Or, it can be achieved in time dimension by randomly adding dummy cycles, shuffling the order of operations or randomly changing the clock frequency for example [MOP07]. Randomization approaches operating in time dimension are the most common techniques in this field. A popular example for this type of hiding countermeasure is Random Start Index Shuffling (RSIS) [VMKS12].

Masking. Masking is undoubtedly the most popular defense mechanism against side-channel analysis attacks. In contrast to hiding countermeasures, it is possible to prove the security guarantees obtained by masking schemes in theoretical leakage security models. This class of countermeasures, also known as secret sharing, relies on splitting each sensitive variable of an algorithm into a discrete number of shares in such a way that only the combination of all of the shares contains information about the sensitive values [CJRR99, PR13]. In this way, a security level in terms of the required number of leakage traces for a successful attack can be achieved which grows exponentially in the protection order (often closely related to the number of shares) while spending approximately a quadratic amount of resources [JS17, FGP$^+$18]. Yet, such a relation can only be established when the leakage of the individual shares is sufficiently independent and the measurements that an adversary can acquire are

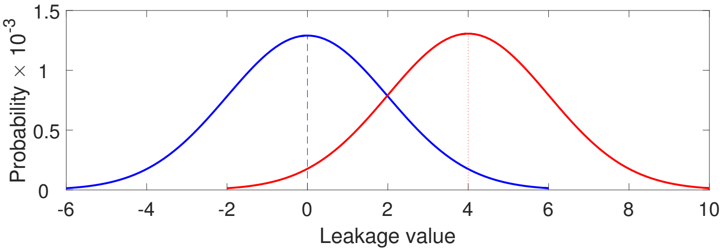


Figure 2.7: This figure shows two hypothetical leakage distributions for different values of an intermediate result in an unmasked implementation.

sufficiently noisy [SVO+10, PR13, FGP+18]. Without a sufficient amount of noise, masked implementations are not expected to provide a security level that increases exponentially in the protection order [CJRR99, SVO+10, PR13, Sta18, Moo19, BS21], making the trade-off between spent resources and obtained security guarantee ineffective. Thus, it is often necessary to combine masking and hiding schemes to achieve high levels of protection. To illustrate the effect of masking or secret sharing on the observable leakage through side channels we consider the following simple example. Let us assume that a 4-bit sensitive intermediate value gets saved into a register inside of a cryptographic device. The power consumption of the device leaks information about this operation according to the following leakage function, with $x \in \{0,1\}^4$, $\mu = 0$, $\delta = 2$, $HW(\cdot)$ being the Hamming weight function and $\mathcal{N}(\mu, \delta^2)$ being a normal distribution with mean μ and variance δ^2.

$$l(x) = HW(x) + \mathcal{N}(\mu, \delta^2)$$

In case of $x = 0000_2$, the equations on the left (blue) hold, with $\mathbf{E}(\cdot)$ returning the expected value. In case of $x = 1111_2$, the equations on the right (red) are correct.

$$\begin{aligned} x &= 0000_2 & x &= 1111_2 \\ l(x) &= HW(0000_2) + \mathcal{N}(0, 2^2) & l(x) &= HW(1111_2) + \mathcal{N}(0, 2^2) \\ l(x) &= 0 + \mathcal{N}(0, 2^2) & l(x) &= 4 + \mathcal{N}(0, 2^2) \\ \mathbf{E}(l(x)) &= 0 & \mathbf{E}(l(x)) &= 4 \end{aligned}$$

It can be observed that the expected value of the leakage function is different for the two cases. The corresponding leakage distributions and their means are depicted in Figure 2.7. Clearly, the two distributions can be distinguished by their means. Hence, the power consumption of the hypothetical register leaks about value x in the first statistical order and an adversary can determine with a high success rate for each measurement conducted which of the two values was processed internally.

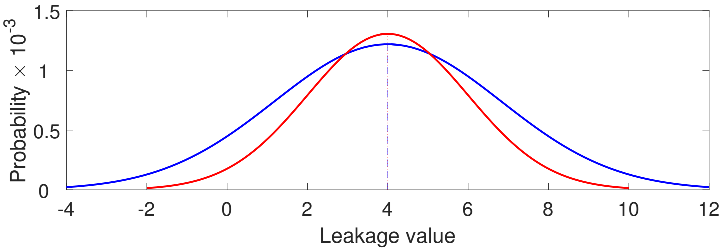


Figure 2.8: This figure shows two hypothetical leakage distributions for different values of a shared intermediate result in a first-order masked implementation.

When splitting the sensitive intermediate value x into two shares and saving the two 4-bit values into separate registers, the combined leakage function can be expressed as follows, with $m \leftarrow \{0,1\}^4$ and $x_m = x \oplus m$.

$$l(x_m) + l(m) = HW(x_m) + HW(m) + \mathcal{N}(\mu, \delta^2)$$

In case of $x = 0000_2$, the equations on the left (blue) hold, while in case of $x = 1111_2$, the equations on the right (red) are correct.

$$\begin{aligned} x &= 0000_2 \\ l(x_m) + l(m) &= HW(0000_2 \oplus m) \\ &\quad + HW(m) + \mathcal{N}(0, 2^2) \\ l(x_m) + l(m) &= 2 \cdot HW(m) + \mathcal{N}(0, 2^2) \\ \mathbf{E}(l(x_m) + l(m)) &= 4 \end{aligned}$$

$$\begin{aligned} x &= 1111_2 \\ l(x_m) + l(m) &= HW(1111_2 \oplus m) \\ &\quad + HW(m) + \mathcal{N}(0, 2^2) \\ l(x_m) + l(m) &= 4 + \mathcal{N}(0, 2^2) \\ \mathbf{E}(l(x_m) + l(m)) &= 4 \end{aligned}$$

Here, the expected value of the leakage function is identical for the two different values for x. The two corresponding leakage distributions and their means are depicted in Figure 2.8. Obviously, the distributions share the same mean value and therefore cannot be distinguished by their means. In this scenario an adversary has to acquire a significantly larger number of measurements in order to sample the leakage distributions in sufficient quality to estimate their variances and successfully distinguish them. When splitting the sensitive intermediate x into even more shares and store them in separate, independently leaking, registers, the distinction of the leakage distributions for different values of x becomes exponentially more difficult. This is the very basic concept of masking as a countermeasure against side-channel attacks.

Glitch-Resistant Masking. With respect to the protection of hardware implementations against passive and non-invasive physical attacks, glitch-resistant masking (also known as hardware-based masking or hardware masking) has become one of the most promising research directions. When implementing the masking countermeasure in hardware, physical defaults such as glitches, naturally occurring in combinatorial circuits, can easily contradict the independence assumption required for secure masking. Thus, resistance to glitches and other timing dependencies potentially leading to a recombination of masked intermediates need to be considered at the design level. This particular field was sparked by the introduction of threshold implementations (TIs) in 2006 [NRR06] and since then has been complemented by a number of further schemes, including [RBN$^+$15, CRB$^+$16, GMK16, GMK17, GM17, GM18, GIB18, FGP$^+$18]. However, in contrast to the situation in software-based masking, the majority of these schemes comes without a proof in a formal security model that attests their probing security and composability in the presence of glitches for arbitrary security orders.

(Robust) Probing Security and Composability Definitions. At CRYPTO 2003, Ishai, Sahai and Wagner introduced the t-probing model as the first theoretical abstraction to prove the security of masked implementations [ISW03]. The model considers adversaries that are able to probe up to t wires inside the targeted cryptographic implementation. Security against such adversaries requires that any combination of t probed wires is insufficient to learn information about any sensitive intermediates of the cryptographic algorithm. Here, we use the following definition.

Definition 1 (t-probing security [ISW03]) *A circuit* C *is t-probing secure if and only if every t-tuple of its intermediate variables is independent of any sensitive variable.*

One important limitation of this definition of probing security is that it does not guarantee composability. In more detail, the composition of two t-probing secure masked multiplication gadgets is not necessarily t-probing secure itself since using an output of a gadget as the input of another one can give additional information to the adversary. To deal with this insufficiency, Barthe *et al.* [BBD$^+$16] introduced the notions of t-Non-Interference and t-Strong-Non-Interference.

Definition 2 (t-Non-Interference [BBD$^+$16]) *A circuit gadget* G *is* t-Non-Interfering *(t-NI) if and only if for any set of t_1 probes on its intermediate values and every set of t_2 probes on its output shares with $t_1 + t_2 \leq t$, the totality of the probes can be simulated with only $t_1 + t_2$ shares of each input.*

Definition 3 (t-Strong-Non-Interference [BBD$^+$16]) *A circuit gadget* G *is* t-Strong-Non-Interfering *(t-SNI) if and only if for any set of t_1 probes on its intermediate values and every set of t_2 probes on its output shares with $t_1 + t_2 \leq t$, the totality of the probes can be simulated with t_1 shares of each input.*

Non-Interference (NI) and Strong Non-Interference (SNI) are security notions which support compositional reasoning for gadgets. In fact, SNI is a stronger notion than NI, which is itself a stronger notion than probing security. We say that a gadget is composable if it is SNI. In [FGP$^+$18] the above mentioned security notions are extended by modeling the adversarial probes in such a way that physical defaults such as transitions, glitches and couplings are captured. The result is called the *robust probing model* and can be used to reason about the probing security and composability of masked gadgets realized as hardware circuits.

2.3.6 Leakage Assessment

Ever since the introduction of side-channel attacks in 1999 [KJJ99] the standard approach for assessing the physical vulnerability of a device has been a more or less exhaustive verification of its resistance against known attacks while attempting to cover a broad range of intermediate values and hypothetical leakage models. This approach, however, became less feasible over the years due to the increasing amount of new attack methods and the higher complexity of potential leakage models due to the introduction of countermeasures against physical attacks. Another concern regarding this procedure is that it entails a significant risk of reporting physical security in favor of the device under test (DUT) while in reality merely a certain attack vector was missed in the process (by mistake or because it was unknown at time of evaluation) that could indeed enable key recovery [SM15]. Hence, the need for a robust and reliable standard leakage assessment method independent of concrete attack scenarios, targeted intermediates and hypothetical leakage models grew consistently over the years. In an attempt to gather and evaluate promising candidates, the National Institute of Standards and Technology (NIST) hosted a "Non-Invasive Attack Testing Workshop" in 2011. One of the most intriguing proposals at the workshop was the use of the non-specific Welch's t-test [GJJR11] for leakage detection. Leakage detection avoids any dependency on the choice of intermediates and leakage models by focusing on the detection of leakage only, without paying any attention to the possibility to exploit said leakage for key recovery. Simply put, the concept is based on supplying the device under test with different inputs, recording its leakage behavior and evaluating whether a difference can be observed. Thus, such a method is suitable for black box scenarios and allows certification of a device's physical security by third party evaluation labs without the need to test a multitude of different methods and parameter combinations. Seven years later, after some shortcomings of the moment-based nature of the t-test had been identified [Sta18], another popular statistical hypothesis test was proposed for leakage detection purposes, namely the Pearson's χ^2-test [MRSS18]. Both hypothesis tests, the t-test and the χ^2-test, are applied in the field of statistics in order to answer the question whether two sets of data are significantly different from each other. To be more precise, the evaluation of the tests examines the validity of the null hypothesis, which constitutes that both sets of data were drawn from the same population (i.e., they are indistinguishable) [SM15]. In side-channel analysis contexts, it is usually evaluated whether two groups of measurements can be distinguished with confidence. Traditionally, those two groups are acquired by supplying the DUT either with random (group Q_0) or a fixed input (group Q_1), selected by coin toss. Later, it has been demonstrated that the careful choice of two distinct fixed inputs (instead of maintaining one group for random inputs) usually leads to a lower data complexity for the distinction [DS16]. We provide the details on how to conduct the Welch's t-test and Pearson's χ^2-test below.

Welch's t-test. We denote two sets of data by Q_0 and Q_1, their cardinality by n_0 and n_1, their respective means by μ_0 and μ_1 and their standard deviations by s_0 and s_1. The t-statistics and the degrees of freedom v can then be computed using the following formulas.

$$t = \frac{\mu_0 - \mu_1}{\sqrt{\frac{s_0^2}{n_0} + \frac{s_1^2}{n_1}}} \qquad v = \frac{\left(\frac{s_0^2}{n_0} + \frac{s_1^2}{n_1}\right)^2}{\frac{\left(\frac{s_0^2}{n_0}\right)^2}{n_0 - 1} + \frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1 - 1}}$$

Afterwards, the confidence p to accept the null hypothesis can be estimated via the Student's t probability density function, where $\Gamma(\cdot)$ denotes the gamma function [SM15, MRSS18].

$$p = 2\int_{|t|}^{\infty} f(t,v)dt \qquad\qquad f(t,v) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{\sqrt{\pi v}\,\Gamma\left(\frac{v}{2}\right)}\left(1+\frac{t^2}{v}\right)^{-\frac{v+1}{2}}$$

In practice, for the sake of simplicity, it is common to only evaluate the t-statistics and to set the confidence threshold for distinguishability to $|t| > 4.5$. The statistical background of this threshold is that for $|t| > 4.5$ and $v > 1000$ the confidence p to accept the null hypothesis is smaller than 0.00001 which is equivalent to a 99.999 % confidence that the two sets were *not* drawn from the same population. Of course, when the degrees of freedom v are not explicitly evaluated, it can occur that the assumption $v > 1000$ does not hold. However, practice has shown that this particular simplification rarely produces false positive results in side-channel analysis contexts. Yet, calculating the actual confidence p is certainly preferable, scientifically correct and can still be efficiently implemented [MRSS18]. Since the Welch's t-test is designed to distinguish the means of two distributions, it can only be applied to first-order univariate analyses in its simplest form. Schneider *et al.* [SM15, SM16] extended the methodology to arbitrary orders and variates and provide the required formulas for incremental one-pass computation of all moments.

Pearson's χ^2-test. In order to mitigate some of the limitations and shortcomings of the moment-based nature of the Welch's t-test, in particular for higher-order analyses of masked implementations, Moradi *et al.* [MRSS18] suggested the Pearson's χ^2-test. In contrast to the t-test this hypothesis test analyzes the full distributions and can capture information that lies in multiple statistical moments. Thus, it prevents false negatives when moment-based analyses become suboptimal [MRSS18].

In a first step a contingency table F has to be constructed from the two sets Q_0 and Q_1 (basically two histograms). We denote the number of rows by r (= 2, when two sets are compared) and the number of columns by c (number of bins of the histograms). The χ^2-statistics x and the degrees of freedom v can then be computed using the following formulas.

$$\mathrm{x} = \sum_{i=0}^{r-1}\sum_{j=0}^{c-1}\frac{(F_{i,j}-E_{i,j})^2}{E_{i,j}} \qquad\qquad \mathrm{v} = (r-1)\cdot(c-1)$$

$E_{i,j}$ denotes the expected frequency for a given cell.

$$E_{i,j} = \frac{\left(\sum_{k=0}^{c-1} F_{i,k}\right)\cdot\left(\sum_{k=0}^{r-1} F_{k,j}\right)}{N}$$

Finally, the confidence p to accept the null hypothesis is estimated via the χ^2 probability density function, where $\Gamma(\cdot)$ denotes the gamma function [MRSS18].

$$p = \int_{\mathrm{x}}^{\infty} f(x,\mathrm{v})dx \qquad\qquad f(x,\mathrm{v}) = \begin{cases} \frac{x^{\frac{\mathrm{v}}{2}-1}e^{-\frac{x}{2}}}{2^{\frac{\mathrm{v}}{2}}\Gamma\left(\frac{\mathrm{v}}{2}\right)} & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

In contrast to the *t*-test this procedure can easily be extended to more than two sets of data ($r > 2$), which can be a valuable feature when used as a distinguisher for key recovery attacks. Generally, it can be said that in cases where the χ^2-test provides a higher confidence to reject the null hypothesis than the *t*-test (on the same side-channel data), the analysis of the leakages requires some special attention. This is usually the case when masked implementations with low noise levels are analyzed [Sta18, Moo19] or when hardware-masking schemes like threshold implementations cause leakages in multiple moments due to physical defaults such as glitches [Sta18, MRSS18].

False Positives. False positives commonly appear as a problem in classical leakage assessment. We say that a Welch's *t*-test or a Pearson's χ^2-test result is falsely positive in a leakage detection scenario if the confidence threshold is exceeded for at least one sample point, despite the absence of leakage. In other words, a false positive occurs when the test decides to reject the null hypothesis for at least one sample point where it is in fact true [WO19]. This phenomenon is caused by the point-wise independent nature of classical detection methods. A threshold of $p_{\text{th}} = 10^{-5}$ set for each individual point will lead to an aggregation of the error probability over the length of the entire trace. More formally, the likelihood that a false positive occurs at least once in a trace of length K can be described as (assuming independence between the tests [WO19]):

$$P(\text{false positive}) = 1 - (1 - p_{\text{th}})^K$$

For an exemplary value of $K = 5\,000$ and the common threshold of $p_{\text{th}} = 10^{-5}$ this formula equates to 0.0488. Thus, the probability that the detection threshold is falsely exceeded for at least one sample point is roughly 5% (when using the common methodology for *t*- and χ^2-test in leakage assessment). While the evaluator may have desired a confidence of $1 - 10^{-5} = 99.999\%$ in the reported leakage by setting $p_{\text{th}} = 10^{-5}$, the actual result can provide a confidence of only $1 - 0.0488 = 95.12\%$ considering the full trace length. For longer traces the situation is even worse. Hence, a manual investigation of the individual leakage points is sometimes necessary when performing classical leakage detection to exclude false positives. Whitnall and Oswald suggested multiple different solutions to this fundamental problem in their work [WO19], including the Bonferroni correction [Dun61], the Ŝidák correction [Ši67] and the Holm procedure [Hol79]. They also conclude that these correction techniques inevitably increase the risk of false negatives, which is undesirable from an evaluators point of view.

2.3.7 Deep Learning in Side-Channel Analysis

Historically, the field of machine learning dealt with extracting meaningful information from data by applying relatively simple mathematical models, e.g., Bayes Classifiers, Support Vector Machines or Decision Trees to a sanitized version of the input data. This required manual and time-consuming *feature engineering* to predetermine which elements might be useful in a given set of raw data and how to best represent them, e.g., Canny edge detection as a first hard-coded step for image classification. In contrast, deep learning methods are generally capable of learning from raw input data, thereby making the elaborate modeling process unnecessary. Since the breakthrough improvement of classification accuracy on the ImageNet data set in 2012 [KSH12], deep learning has been successfully applied to many diverse tasks such as speech recognition, drug discovery, natural language processing, visual art style transfer, image classification, autonomous driving and strategy games. More recently, the side channel community discovered deep learning as a tool to perform profiled attacks [HGM$^+$11, MPP16, CDP17, MDP20] with competitive results compared to classical modeling techniques, e.g., based on a multivariate normal distribution. On the other hand, the run-time effectiveness of DL-based approaches over classical machine learning is sometimes questioned [PSK$^+$18]. However, only few publications have investigated the use of deep learning for the non-profiled case. Such works include [Tim19] and [PCBP21]. In the former article a method is introduced that exploits the correlation between a correct key guess and a steep learning rate to enable key recovery. Unfortunately, the method is computationally intense as a separate model has to be trained for every key guess and its success is highly dependent on the correct labeling of the data which implies a suitable choice of leakage model and targeted intermediate. The latter article introduces a novel framework based on unsupervised learning to improve horizontal attacks on (protected) implementations of public-key cryptosystems.

2.4 Measurement Techniques and Equipment[4]

In this section we discuss common methods and tools for acquiring side-channel measurements from cryptographic devices in order to perform power analysis attacks. In the following we differentiate between approaches for dynamic and approaches for static power side-channel analysis, as the applied measurement techniques differ significantly from each other. In the case of static power measurements we also distinguish between setups using an oscilloscope and those using a Source Measure Unit (SMU). Finally, we list details about the specific setups used to obtain the practical results presented in this thesis.

2.4.1 Dynamic Power Measurements

To perform a dynamic power analysis attack it is necessary to record the instantaneous power consumption of a cryptographic device over a certain time period that includes the execution of a cryptographic algorithm (at least partially) or another targeted operation. In order to obtain a quantitative value proportional to the dynamic power consumption, usually either the voltage drop over a shunt resistor placed in the VDD or GND path of the device is monitored, or the electromagnetic emanation is measured with a near-field probe placed in close proximity to the cryptographic implementation on the device. In both cases, the resulting signal is sampled by a digital sampling oscilloscope. Optionally, the measured signal can be amplified through an AC-coupled amplifier before being recorded by the oscilloscope. Dedicated measurement boards for dynamic power analysis investigations, such as the SAKURA-G [Sak] for example, often come equipped with built-in AC amplifiers. As a result of the measurement procedure, each recorded trace consists of a number of discrete sample points. Typically, it is possible to identify the clock cycles of the computation when plotting such a side-channel trace. In some cases even details about the structure of the executed algorithm, as for example the different rounds in a block cipher implementation, become visible [MOP07]. An example, borrowed from [MWM21], is depicted in Figure 2.9. Clearly, the power peaks associated with each clock cycle can be identified. Additionally, the end of the first round can be determined between time samples

---

[4] This section contains excerpts of our publication [MMR20].

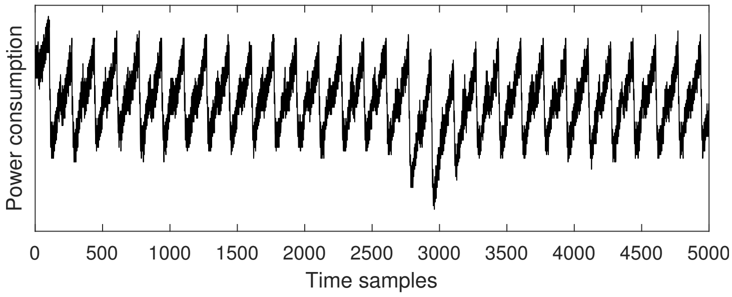


Figure 2.9: This figure shows a sample trace of a serialized PRESENT hardware implementation measured on a SAKURA-G board.

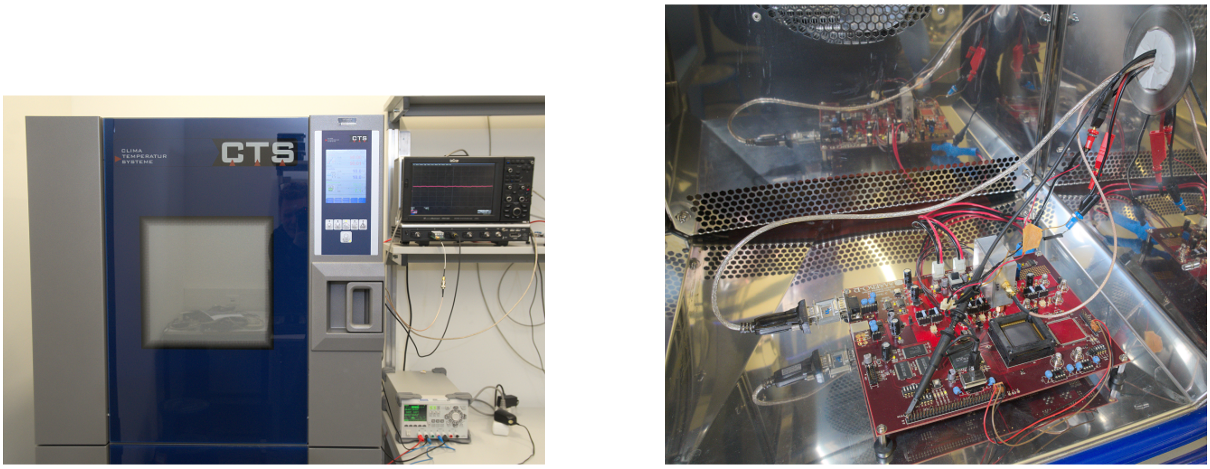
2 800 and 3 000. The acquisition of traces is then repeated as often as required to perform a successful attack.

2.4.2 Static Power Measurements

In order to measure the static power consumption and perform successful static power side-channel analysis attacks, a few essential problems need to be overcome by the measurement setup. The first one is that the differences in the current drawn by a device caused by the presence of different data values in the circuit is typically rather small, often in the range of nano to micro amperes. The second problem is the susceptibility to temperature variations. The leakage currents of CMOS devices are highly temperature dependent which results in huge shifts of the measured signal even for comparably small temperature variations, as for example when the measurement room is entered by a person. Thus, in order to obtain accurate measurements, a high precision is required and the environmental temperature should be as constant as possible.

Climate Chamber. In order to keep the temperature constant, it is advisable to perform the static leakage measurements inside a climate test chamber. In this thesis we have employed a CTS climate test chamber of series C-40/100 with 100 liters test space capacity [Cli]. The chamber achieves temperatures between $-40\,°\mathrm{C}$ and $+180\,°\mathrm{C}$ as well as a temperature change rate of $5\,\mathrm{K\,min^{-1}}$ for cooling and $3\,\mathrm{K\,min^{-1}}$ for heating. It can hold the temperature with a variation of 0.3 °C at a maximum thermal load of 1200 W at 20 °C. This should highly suffice for SCA purposes as the cryptographic devices typically analyzed are not expected to radiate a considerable amount of heat (resulting in even smaller temperature variations). The device under test should be placed inside the chamber whereas the measurement instrument for data acquisition and the power supply should be placed outside of the chamber. The cables connecting the devices can be passed through a vent in the chamber which is then carefully sealed with silicone foam. The resulting setup can look like Figure 2.10 for example.

Setup based on an Oscilloscope. When attempting to measure the static current drawn by a device under test with a common SCA-setup based on an oscilloscope as the central measurement instrument, a high DC amplification is needed as the differences in the voltage drop over a shunt resistor to be measured are extremely small. Also, many probes and amplifiers suffer from a DC



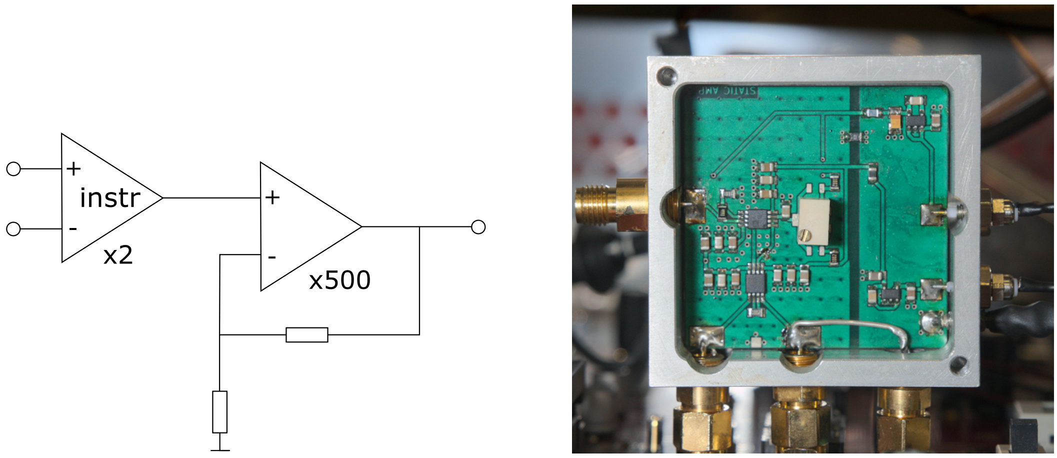
(a) Outside

(b) Inside

Figure 2.10: These photographs show a setup to perform static power measurements inside a climate chamber.

shift when they heat up during use. In order to overcome these drawbacks, we have developed a sophisticated amplifier to measure the static leakage in [MMR20]. A schematic of the amplifier and a photo of the employed board in its aluminum case can be seen in Figure 2.11. The first stage of the amplifier consists of an Analog Devices AD8421 instrumentation amplifier [Ins], which provides a very low temperature dependency with $0.2\,\mu\text{V}\,^{\circ}\text{C}^{-1}$ maximum offset voltage drift and 1 ppm/°C gain drift. This stage removes the common voltage between its two inputs which are connected to the two terminals of the shunt resistor and applies an amplification with a gain of 2. A second stage consisting of an Analog Devices AD8676 operational amplifier (op-amp) [Opa] applies a ×500 amplification to the resulting signal (i.e. the DC amplifier achieves a total gain of ×1000). This op-amp also has a low temperature dependency of $0.6\,\mu\text{V}\,^{\circ}\text{C}^{-1}$ input offset drift. The PCB of the amplifier is housed in a custom aluminum case which provides SMA connectors. Due to the high gain, the bandwidth of the amplifier is below 20 kHz which does not pose a problem since it is used to measure static signals.

During the measurement procedure with this amplifier and an oscilloscope some high-frequency (thermal) noise in the measurements might be observed which increases for higher temperatures. Hence we built a custom low-pass filter to remove these portions of the signal and to connect it between the output of the DC amplifier and the input of the oscilloscope. The filter, which is shown in Figure 2.12, is built as a passive third-order Butterworth Pi LC construction to provide a cutoff-frequency (−3 dB) of approximately 100 Hz for a $50\,\Omega$ input impedance of the oscilloscope. This modification of the setup reduced the measurement interval, i.e., the time period in which all sample points are averaged, to reach a certain signal-to-noise ratio by a factor of about 5 when operated at a temperature of 90 °C. We have used a Teledyne LeCroy HRO 66zi oscilloscope [HRO] for the static power measurements presented in this thesis. This scope provides a true 12-bit ADC, a maximum sampling rate of 2 GS/s, and a maximum bandwidth of 600 MHz. A schematic of the resulting complete static power measurement setup based on an oscilloscope is shown in Figure 2.13.



(a) Schematic

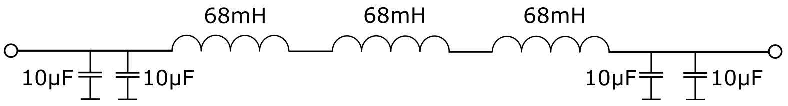
(b) Photo

Figure 2.11: This figure shows a custom low-noise DC amplifier for static power measurements.

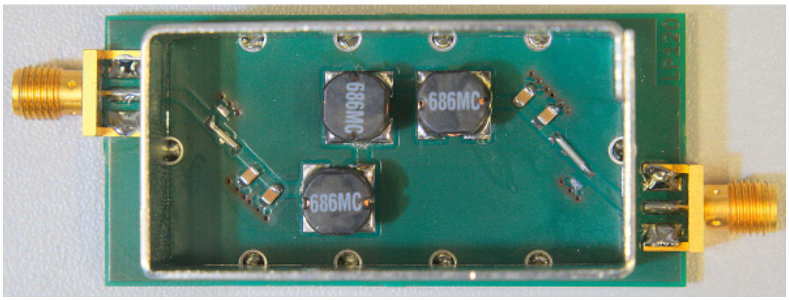
Setup based on an SMU. Instead of using an oscilloscope, it is also possible, and in most cases even preferable, to use a Source Measure Unit (SMU) like the Keithley 2450 Sourcemeter [Kei] shown in Figure 2.14. This instrument has been specifically designed for characterizing nanoscale semiconductors and other small-geometry and low-power devices. The SMU is then used to supply the voltage to the device under test and simultaneously measure the leakage currents through the device. In this case, neither a DC amplification, nor a low-pass filter is needed. The resulting setup is depicted in Figure 2.15.

Measurement Procedure. The measurement procedure used in static power measurements also differs from typical trace acquisition for dynamic power SCA. Normally, at the specific clock cycle where the targeted intermediate value is processed by the cryptographic device, the global clock signal is paused and all signals to the device are kept at a constant value. This idle state of the target is held for an arbitrarily long time interval during which the static power consumption of the device can be measured before the clock signal is continued. The measured signal usually requires a certain settling time after the clock is stopped due to the abrupt change in the power consumption. Independent of the usage of oscilloscope or SMU, the measured signal during this settling time is not informative and needs to be ignored. Typical values for the length of this period are 10-20 ms. After the end of this period it is possible to capture as many measurements as desired during the idle state and average them to a singular value. Since the leakage currents are not supposed to change during that period, all occurring variations are noise and can be averaged out. This technique is called intra-trace averaging and constitutes one major advantage of static power analysis in comparison to classical attacks. Yet, typically, recording static leakage traces requires a stronger attacker model than it would be required for a classical power analysis, as control over the clock signal is necessary.

It is often advisable to perform a simple post-processing to filter out the long-term temperature-induced variations of the static power consumption over time, i.e., over a set of measurements



(a) Schematic



(b) Photo

Figure 2.12: This figure shows a third-order (Butterworth Pi) LC low pass filter with cutoff-frequency of approximately 100 Hz.

as opposed to noise that is included in single measurements. This is especially relevant at higher temperatures. Quite obviously the climate chamber requires a lot more activity of its regulation units to maintain a constant temperature when it is set to a value far above or below the temperature of the room it is located in. These activities can be observed as low frequency noise along the whole set of measurements. The suggested post-processing step is therefore to apply a simple moving-average filter on the measurement set, for example by using the Matlab function `filter()`[5]. The effect of such filtering is depicted in Figure 2.16. The blue plot in Figure 2.16(a) corresponds to a set of 100 unaltered measurements as they were recorded by the setup. The red curve corresponds to the moving-average that is generated by the Matlab

---

[5] We also tested other filters in Matlab, for example a butterworth high-pass filter, but achieved inferior results.

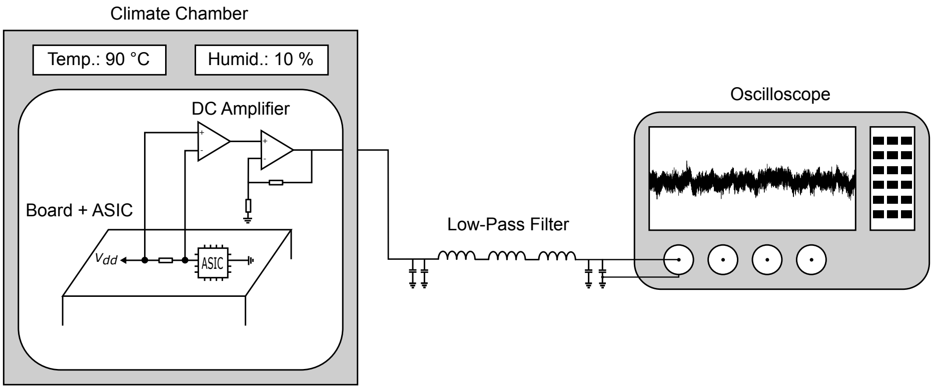


Figure 2.13: This figure illustrates the complete static power measurement setup based on an oscilloscope.

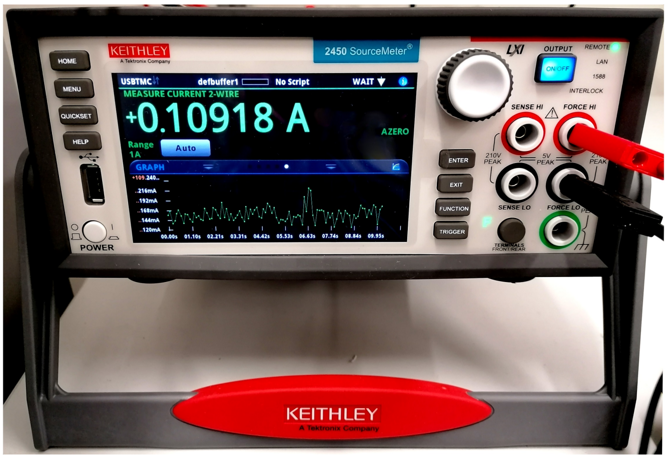


Figure 2.14: This figure shows a Keithley 2450 Source Measure Unit (SMU).

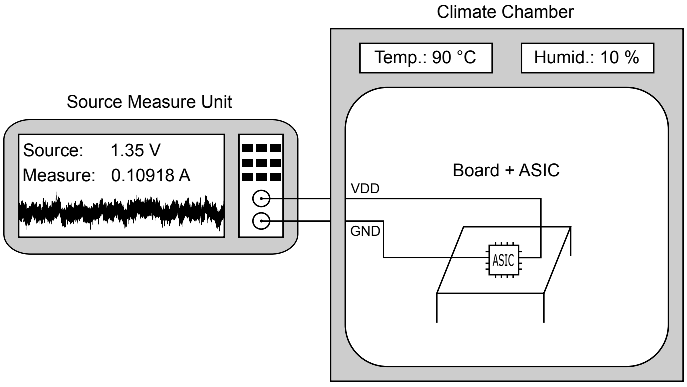


Figure 2.15: This figure illustrates the complete static power measurement setup based on a Source Measure Unit (SMU).

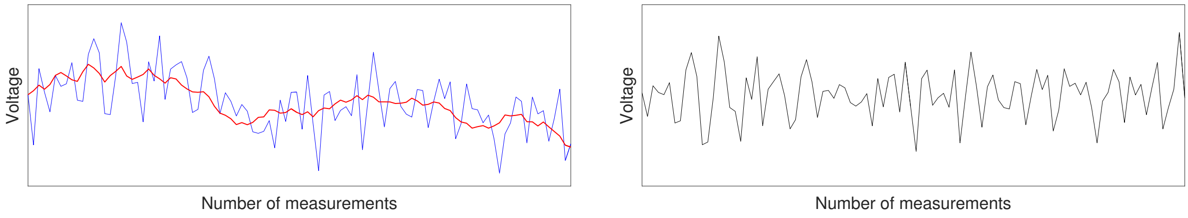
`filter()` function. The subtraction of the moving-average from the original measurements yields the black graph in Figure 2.16(b) and constitutes the resulting measurement set after the post-processing. Although the initial purpose of this technique was to improve the measurement quality at extreme temperatures (like −20 °C or 90 °C) we observed that it has a positive influence on the measurements in all cases, even at room temperature.

2.4.3 List of Equipment used in this Thesis

In the following we list the commercial and custom equipment used to acquire the experimental results presented in this thesis. For each device we reference the corresponding publications.

Electrical Measurement Instruments.

- **Teledyne LeCroy HRO 66zi Oscilloscope** [HRO], used in [MM17, KMM19, MMSS19, Moo19, MMR20, Moo20, MWM21].



(a) original measurements and moving-average curve

(b) altered measurements

Figure 2.16: This figure illustrates the post-processing technique for 100 exemplary measurements.

- **Teledyne LeCroy WaveRunner 8254M Oscilloscope** [Wav],
  used in [MMSS19, Moo20].
- **Keithley 2450 SourceMeter** [Kei],
  used in [MM21].

Commercial and Custom Measurement Boards.

- **SAKURA-G FPGA board** [Sak],
  used in [MMSS19, MWM21]
- **SASEBO-R ASIC board** [Sas],
  used in [MM17, MMR20]
- **Custom measurement board for 90 nm and 65 nm ASICs**, Figure 2.2(a),
  used in [KMM19, Moo19]
- **Custom measurement board for 40 nm ASIC**, Figure 2.2(b),
  used in [Moo20, MWM21]
- **Custom measurement board for 28 nm ASIC**, Figure 2.2(c),
  used in [MM21]

Climate Chamber.

- **CTS C-40/100 Climatic Test Chamber** [Cli],
  used in [KMM19, Moo19, MMR20, Moo20, MM21]

Amplifiers and Filters.

- **Custom DC Amplifier**, Figure 2.11,
  used in [KMM19, Moo19, MMR20, Moo20]
- **Custom Low-Pass Filter**, Figure 2.12,
  used in [KMM19, Moo19, MMR20, Moo20]

Part II

Publications

Chapter 3

Static Power Side-Channel Analysis

In this chapter we present the peer-reviewed publications accumulated in this thesis with relation to the topic of static power side-channel analysis. In total, this chapter covers one paper published in the IEEE Transactions on Very Large Scale Integration Systems (TVLSI) and four papers published in the IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES).

Contents of this Chapter

3.1 Static Power Side-Channel Analysis - An Investigation of Measurement Factors . . . 51
3.2 Static Power SCA of Sub-100 nm CMOS ASICs . . . 67
3.3 Exploring the Effect of Device Aging on Static Power Analysis Attacks . 101
3.4 Unrolled Cryptography on Silicon . . . 127
3.5 Countermeasures against Static Power Attacks . . . 157

3.1 Static Power Side-Channel Analysis - An Investigation of Measurement Factors

Publication Data

Thorben Moos, Amir Moradi, and Bastian Richter. Static power side-channel analysis - an investigation of measurement factors. *IEEE Trans. Very Large Scale Integr. Syst.*, 28(2):376–389, 2020

Content This work investigates the impact of chosen measurement factors like the supply voltage, the temperature and the measurement interval on the success of static power side-channel attacks targeting a 150 nm CMOS ASIC. For the first time in literature it is demonstrated in practice that the operating environment can be manipulated by physical adversaries to obtain measurements with an improved signal-to-noise ratio. Also, it is shown that the length of the measurement interval can be chosen in order to reduce the noise in the measurements almost arbitrarily. Finally, a setup to obtain high-quality side-channel measurements even at higher operating temperatures is proposed.

Contribution The author of this thesis is the principal author of this publication. In particular, all experiments have been conducted and evaluated by the author of this thesis and the presentation of the results is also primarily his contributions. The author would like to thank both co-authors for their significant contributions to the construction and evaluation of the measurement setup.

Static Power Side-Channel Analysis – An Investigation of Measurement Factors

Thorben Moos, Amir Moradi, and Bastian Richter

***Abstract*—The static power consumption of modern CMOS devices has become a substantial concern in the context of the side-channel security of cryptographic hardware. Its continuous growth in nanometer-scaled technologies is not only inconvenient for effective low power designs, but does also create a new target for power analysis adversaries. Additionally, it has to be noted that several of the numerous sources of static power dissipation in CMOS circuits exhibit an exponential dependency on environmental factors which a classical power analysis adversary is in control of. These factors include the operating conditions temperature and supply voltage. Furthermore, in case of clock control, the measurement interval can be adjusted arbitrarily. Our experiments on a 150nm CMOS ASIC reveal that with respect to the signal-to-noise ratio in static power side-channel analyses, stretching the measurement interval decreases the noise exponentially and even more importantly that raising the working temperature increases the signal exponentially. Control over the supply voltage has a far smaller, but still noticeable, positive impact as well. In summary, a static power analysis adversary can physically force a device to leak more information by controlling its operating environment and furthermore measure these leakages with arbitrary precision by modifying the interval length.**

***Index Terms*—static power, leakage, side-channel analysis, side-channel attacks, ASIC, operating conditions, temperature, supply voltage, measurement interval**

I. INTRODUCTION

Established cryptographic primitives usually come along with a set of mathematical security guarantees to inspire confidence in their resistance against state-of-the-art (cryptanalytic) attacks. These guarantees, however, are only valid against computationally bounded black-box adversaries. For instance, it is common to prove that no polynomial-time adversary stands a better-than-negligible chance to compromise the security of a said primitive[1] when being restricted to the observation of its inputs and outputs exclusively. Yet, in embedded contexts such a limitation is not respected, due to the constant physical exposure of the hardware to potential adversaries. Any adversary, who is capable of measuring the physical emissions of a cryptographic device during the execution of a primitive (in sufficiently high quality) has access to substantially more information than just the inputs and outputs. In particular, these observations can directly be correlated to intermediate values of the underlying cryptographic algorithms. This specific kind of adversary model invalidates the security claims of virtually all (raw/unprotected) cryptographic primitives, since they were simply not developed to withstand attacks which make use of their intermediate results. Hence, in many cases it is possible to break the security of physical instances of mathematically secure primitives (e.g., recovering the fixed key of an AES implementation) by carefully observing the emissions of the executing device. To mitigate these threats, dedicated countermeasures which minimize the leakage of information through physical side-channels need to be applied when implementing cryptography in real-world devices.

The static power consumption of CMOS hardware (a.k.a. leakage power) is one type of observable physical characteristic that can be exploited as a side-channel. Due to its, historically speaking, smaller contribution to the overall power consumption when compared to the dynamic currents, it has rarely been considered in traditional side-channel analyses. In view of its exponential growth, however, which is directly linked to the down-scaling of the technology, it has attracted more and more attention over the last decade.

A. History of Static Power Analysis

Ever since the introduction of power analysis attacks in 1999 [23] researchers have concentrated almost exclusively on the exploitation of the operation- and data-dependency that can be observed in the dynamic power consumption of cryptographic hardware. However, in the year 2007 the authors of [16] provided the first concrete evidence for the fact that the leakage currents in modern CMOS gates exhibit a strong data-dependency as well. Additionally, they pointed out that the static power consumption had already reached a considerable dimension for sub-micron CMOS technologies by then. These discoveries consequently led to the first attempts to exploit the emerging new side channel. In [24] a DPA-based attack on (simulated) static power measurements using a single-bit power model is proposed. The works presented in [7] and [8] verify the soundness of the Hamming weight model in the static power domain and conduct a successful CPA attack. Further investigations revealed extensively that multiple DPA-resistant logic styles are rather ineffective against static power analysis [24], [5], [6], [22]. The results of [12] and [9] do even suggest that an unprotected CMOS implementation of the block cipher PRESENT-80 is less vulnerable to such attacks than the same cipher implemented in the DPA-resistant logic style WDDL. To cope with the issue of a possible exploitation of the static currents Zhu et al. proposed first countermeasures in 2013 [46] and 2014 [20]. Further ones have been suggested in the following years [19], [33], [9], [45], [44]. Even extensions to template [42], [11] and multivariate attacks [11], [15] exist in the literature and one first approach

---

T. Moos, A. Moradi and B. Richter are with the Ruhr-Universität Bochum, Horst Görtz Institute for IT-Security, Germany (e-mail: {firstname.lastname}@rub.de.)

[1] by known methods or under a certain assumption

to combine the information leaked through the static and the dynamic power side-channel has been published [43].

However, apart from one small experiment in proof-of-concept manner which has been performed on an 8-bit register [16], [7], all evaluations, all countermeasures and especially all attacks in the previously mentioned articles are exclusively based on simulation results. The first contribution to this field where an analysis has been performed on actual leakage measurements, taken from a physical device, was published in 2014 [28]. Here, detailed information about the leakage currents of different FPGA elements in various process technologies is presented. Additionally, a successful key recovery on a masked and shuffled AES-128 implementation is performed by utilizing the higher-order moments of the static power consumption. The second work in this area with an experimental focus suggests that the ability to control the clock enables adversaries to arbitrarily reduce the noise in their measurements [35]. It was recently confirmed by practical experiments on a threshold implementation prototype chip that this possibility indeed poses a serious threat to algorithmic DPA countermeasures that require high noise levels, such as masking [27]. Additionally, a sophisticated measurement setup is introduced in [27] consisting mainly of a low-noise DC amplifier and a powerful climate chamber. The authors of [10] recently proposed another (distinct) measurement setup dedicated to static power analysis with the objective of being low-cost and demonstrated its suitability by an analysis of a crypto core on a 45nm Xilinx Spartan-6 FPGA. This setup is based on a DC pico-ammeter for trace acquisition and a commercial Peltier cell to control the temperature of the device under test.

B. Role of Operating Conditions

Regarding the role of operating conditions in attack scenarios it is usually (and consistently among all publications) stated that the temperature has to be kept constant during the analysis of the target. The reason for this is the exponential dependency of the static currents on thermal influences. Apart from this constraint the device under test is usually investigated under realistic conditions like room temperature and specified supply voltage. Some of the previously listed works include figures for the data dependency of static currents in CMOS logic gates under different working temperatures [16], [24], [8], [5], [33]. However, those numbers are based on library information of single standard cells and only presented to confirm the suitability of the Hamming weight model regardless of the applied temperature (as long as it is kept constant during the whole acquisition of a set of traces). Up to this point it was paid little attention to the fact that an increase of the temperature also increases the absolute difference between the leakage currents for the different possible input vectors to digital standard cells. Since the ratio between the currents for any two input vectors is roughly maintained and the absolute currents are exponentially increasing when more thermal energy is applied, the signal in a side-channel attack can – in theory – artificially be amplified in an exponential manner by raising the temperature. We evaluate the soundness of this hypothesis based on an experimental analysis in Section VI.

Another crucial parameter for the correct operation of integrated circuits is the supply voltage. To the best of our knowledge no work has investigated the influence of changes in the supply voltage on the exploitability of the static currents so far, neither in simulations nor in practice, even though strong dependencies of the static dissipation on potential-differences in CMOS transistors are known to exist. We discuss this in more detail in Section VI.

C. Related Work

The authors of [12] seem to exploit the impact of increased temperatures on the success of attacks for the first time. In [12] an implementation of the PRESENT-80 block cipher [13] is simulated in 40 nm technology and its susceptibility to static power side-channel attacks is analyzed. The simulations are performed while the working temperature has been set to 100 °C and it is referred to this operating condition as the "worst-case scenario for the designer". However, a comparison to other temperatures is not presented.

The authors of [10] propose a low cost measurement setup based on a DC pico-ammeter and verify its suitability by performing a static power side-channel analysis of a PRESENT-80 implementation on a Spartan-6 FPGA. In their practical experiments the device under test (DUT) was heated up by a conventional Peltier cell to 65 °C. A comparison to other temperatures is not presented in this work either.

Subsequently, in 2017 two works have been published that utilize the temperature as a replacement for the missing time-dimension in static power analysis to perform multivariate attacks [11], [15]. While one of these works focuses on conducting template attacks which exploit the static power consumption [11], the other comes to the conclusion that increasing the working temperature progressively eases a static power side-channel analysis in terms of the required number of measurements for a successful key recovery [15]. The authors of [15] compare different temperatures between 0 and 100 °C. However, due to the simulation-based nature of those investigations the authors are forced to make assumptions about the noise and its very own dependency on the working conditions. From our point of view a natural assumption would be that the static currents of all non-targeted parts of the circuit, i.e., the algorithmic noise (see e.g. [40], [18] for descriptions of algorithmic noise), is affected in the same way as the targeted parts and that other noise sources, e.g., the electronic noise, the measurement noise or the quantization error, are less affected. We are not entirely sure which assumptions the authors in [11] and [15] make. In both works it is claimed that the signal-to-noise ratio (SNR) is fixed to a value of $-60\,\mathrm{dB}$ regardless of the temperature. If this would be true (and the SNR is meant to be what is frequently applied as metric in the side-channel literature, c.f., Section V) the number of measurements to disclosure (MTD) should not significantly vary between the sets of measurements for different temperatures, since there exists a known anti-proportional relationship between the SNR and the MTD. In particular it was demonstrated in several articles, e.g., [18], that from the SNR alone the MTD value can be predicted. As a consequence one would expect a more or less constant number of measurements required to recover the key for a fixed SNR. This is contradictory to the results presented in [15]. We believe what the authors actually did is fixing the amount of additive white noise over the simulations at different temperatures, i.e., fixing the mean and the standard deviation of the Gaussian distribution that is added onto the noise-free simulated power traces. This would also match the fact that in [15] it is claimed to be possible to extract the same amount of information "even in the presence of lower SNR", while the presented data is actually showing that it is possible to extract the same amount of information *in the presence of more noise* (exactly because the variance of the signal is amplified by the increased temperature). Regardless of the apparent misconception of the authors (or our inability to follow their interpretation) neither the claimed assumption, i.e., fixed SNR over all experiments, nor the assumption that is suggested by the data, i.e., fixed noise standard deviation and mean over all experiments, seems to be a good capture of the reality. In any case it has to be evaluated by practical measurements whether higher temperatures lead to a larger signal-to-noise ratio and therefore to a smaller number of measurements that are required to break an implementation.

D. Our Contribution

In this paper we try to close the gap between theoretical considerations regarding the influence of measurement factors on the feasibility of static power analysis attacks and their practical verification on actual hardware. We answer the question whether an adversary can physically force a device to leak more information by controlling specific operating parameters and provide informative numbers in this regard based on more than two months of non-stop measurements. In particular we have acquired 19 distinct sets with a cardinality of at least 5 million measurements per set in a controlled environment, each for a different temperature-voltage-combination (-20 to 90 °C, 1.62 to 1.98 V), which took roughly 2.7 days for each set. Afterwards, for the most effective temperature-voltage-combination (90 °C and 1.98 V), we recorded another 8 sets of traces for different lengths of the measurement interval. Our results show very clearly that, in this case study, increasing the temperature exponentially increases the signal, that increasing the supply voltage only marginally increases the signal and finally that increasing the measurement interval exponentially decreases the noise. Additionally, it becomes obvious that all three measurement factors can effectively be combined to lower the number of measurements that are required for a successful key recovery to a minimum. Control over these parameters – in theory – allows to eliminate any source of noise except for the algorithmic noise, which highly depends on the particular implementation as well as the concrete attack scenario and will always be present in power measurements [40]. Setup-wise we have built upon [27], but (1) improved the construction of the DC amplifier to obtain stable results at extreme temperatures, (2) built a custom low-pass filter, and (3) employed a simple post-processing technique. All these modifications have been verified to be useful in diminishing the noise and improving the signal.

In short, our contribution can be summarized as follows:

- We propose a measurement setup dedicated to static power side-channel analysis which is applicable over a large range of operating conditions and detail how to avoid potential pitfalls.
- We show that by using such a setup static power side-channel adversaries can exponentially reduce the data complexity of attacks by controlling either the operating temperature or the measurement interval.
- We present results indicating that control over the supply voltage of the device under test marginally reduces the complexity as well.
- We conclude that combining all three measurement factors allows to obtain side-channel data that is free of any noise influence except for the algorithmic noise, which raises a warning flag for countermeasures that require a certain noise level to be effective.

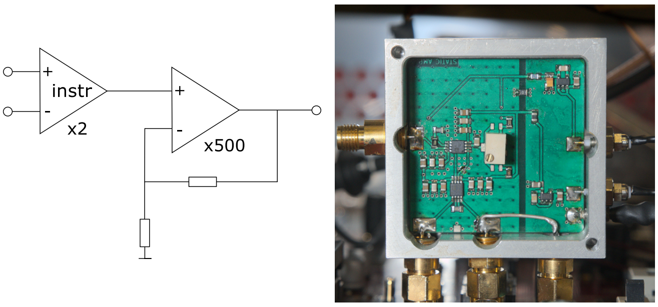
This paper is organized as follows: In Section II we describe the measurement setup for the static power side-channel measurements. Section III introduces the targeted ASIC and the PRESENT block cipher implementation that is investigated. The measurement procedure for acquiring the traces (including the post-processing) is detailed in Section IV, while the necessary evaluation tools and metrics are introduced in Section V. Our main results are then presented in Section VI. Here we investigate the influence of each of the three measurement factors: temperature, supply voltage and measurement interval, on the expolitability of the data-dependent static currents. Finally we conclude our work in Section VII.

II. MEASUREMENT SETUP

In order to measure the static power consumption of our target ASIC, we inserted a precision $1\,\Omega$ resistor with low temperature coefficient into the Vdd path. In contrast to dynamic power measurements the amplifier cannot be AC coupled since AC coupling works as a kind of high-pass filter and would eliminate our static target signal (DC offset). Thus, common AC-coupled amplifiers like the `ZFL-1000NL+` from Mini-Circuits cannot be used in this setup[2]. Instead, the voltage drop over the resistor needs to be measured differentially and with a DC-coupled amplifier. There are two main problems when measuring the static leakage. At first, the voltage difference we would like to measure is very small, typically in the range of a few micro volts. To get an accurate measurement, a high DC amplification is needed. The second problem is the susceptibility to temperature variations. The static leakage itself is highly temperature dependent which results in huge shifts of the measured signal e.g., when the measurement room is accessed. Also, many amplifiers and differential probes suffer from a DC shift when they heat up during use. In [28] a LeCroy AP 033 differential probe which features a

---

[2] Such an AC amplifier has been used in several dynamic power measurement setups, e.g., [17], [29], [30], [31], [38].



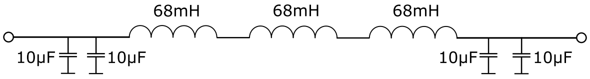
(a) schematic (b) photo

Fig. 1: Low-noise DC amplifier for static power measurements.

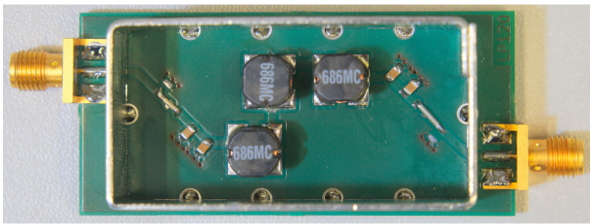
×10 amplification was used. While this probe is capable of measuring the signal with its high common DC offset, it only features a low amplification and is susceptible to thermal shifts in the measurements when the probe heats up during the long measurement procedure.

A. Low-Noise DC Amplifier

In order to overcome these drawbacks, we developed a sophisticated amplifier to measure the static leakage. A schematic of the amplifier and a photo of the employed board in its aluminum case can be seen in Figure 1. The first stage of the amplifier consists of an Analog Devices AD8421 instrumentation amplifier [1], which provides a very low temperature dependency with $0.2\,\mu\mathrm{V}/^{\circ}\mathrm{C}$ maximum offset voltage drift and $1\,\mathrm{ppm}/^{\circ}\mathrm{C}$ gain drift. The amplifier is designed for measuring low level signals as it features an extremely low noise. In particular, it delivers $3\,\mathrm{nV}/\sqrt{\mathrm{Hz}}$ input voltage noise and $200\,\mathrm{fA}/\sqrt{\mathrm{Hz}}$ current noise with only $2\,\mathrm{mA}$ quiescent current [1]. This stage removes the common voltage between its two inputs which are connected to the two terminals of the shunt resistor and applies an amplification with a gain of 2. In contrast to [27] we did not make use of the adjustable offset of the instrumentation amplifier, but rather fixed it to a specific value, since the formerly employed potentiometer increased the noise in our measurements at higher temperatures. A second stage consisting of an Analog Devices AD8676 operational amplifier (op-amp) [2] applies a $\times 500$ amplification to the resulting signal (i.e. the DC amplifier achieves a total gain of $\times 1000$). This op-amp also has a low temperature dependency of $0.6\,\mu\mathrm{V}/^{\circ}\mathrm{C}$ input offset drift and a noise of only $0.10\,\mu\mathrm{V}$ p-p ($0.1\,\mathrm{Hz}$ to $10\,\mathrm{Hz}$) [2]. The PCB of the amplifier is housed in a custom aluminum case which provides SMA connectors. Due to the high gain, the bandwidth of the amplifier is below $20\,\mathrm{kHz}$ which does not pose a problem since we are working with static signals[3].

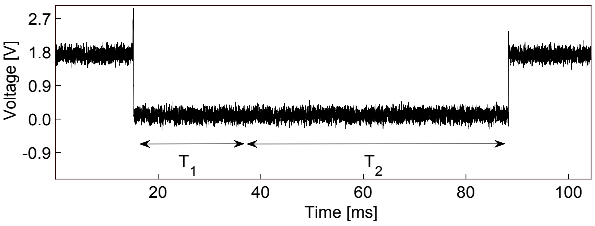


(a) schematic

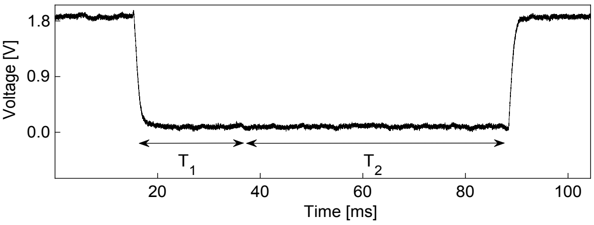


(b) photo

Fig. 2: Third-order (Butterworth Pi) LC low pass filter with cutoff-frequency of ~100 Hz.



(a) without low pass filter



(b) with low pass filter

Fig. 3: Exemplary depiction of two measurements, one with and one without low-pass filter, for a measurement interval of $50\,\mathrm{ms}$. Time period T_1 (here $20\,\mathrm{ms}$) corresponds to the interval that is ignored due to the memory effect. Time period T_2 (here $50\,\mathrm{ms}$) corresponds to the measurement interval. All values that are measured in T_2 are averaged to a singular static power value.

B. Low-Pass Filter

During the measurement procedure we observed some high-frequency noise in the measurements which increased for higher temperatures. Hence we built a custom low-pass filter to remove these portions of the signal and connected it between the output of the DC amplifier and the input of the oscilloscope. The filter, which is shown in Figure 2, is built as a passive third-order Butterworth Pi LC construction to provide a cutoff-frequency ($-3\,\mathrm{dB}$) of approximately 100 Hz for a $50\,\Omega$ input impedance of the oscilloscope. A visual impression of its effect is given in Figure 3 by means of one sample measurement with and one without the low-pass filter applied. The time periods denoted as T_1 and T_2 are introduced in Section IV. As one can see in Figure 3 the amplitude of the oscillating static signal is far smaller when the low-pass filter is applied, especially in relation to the voltage difference between the regions before and after the clock is stopped. This modification of the setup alone already reduced the measurement interval to reach a certain signal-to-noise ratio by a factor of about 5 when operated at a temperature of $90\,^{\circ}\mathrm{C}$.

---

[3] Details of the developed amplifier (schematic, PCB layout) are accessible through the authors' webpage.

C. Evaluation Board

The Side-channel Attack Standard Evaluation Board (SASEBO-R) [3] that we used for our experiments was specifically designed to evaluate the security of cryptographic hardware implementations against side-channel attacks. The board provides a socket for an ASIC prototype that is connected by a 16-bit bidirectional data bus as well as a 16-bit address signal for control and communication purposes. We are able to control the mounted ASIC by a Xilinx Virtex-II Pro FPGA, which itself is connected to a 24-MHz oscillator. Since measuring small signals over long wires can induce measurement errors, we kept the distance between the shunt resistor and the amplifier short by designing the housing of our developed amplifier in such a way that it can be plugged directly on top of the SASEBO-R board by the SMA connectors.

D. Oscilloscope

We used a Teledyne LeCroy HRO 66zi oscilloscope for the measurements. This scope provides a true 12-bit ADC, a maximum sampling rate of 2 GS/s, and a maximum bandwidth of 600 MHz.

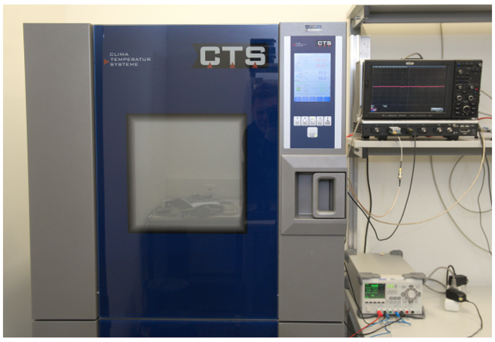
E. Climate Chamber

To quantify the influence of the temperature on the quality of our side-channel acquisitions we performed the static leakage measurements inside a CTS climate test chamber of series C-40/100 with 100 litres test space capacity. The chamber achieves temperatures between $-40\,°\mathrm{C}$ and $+180\,°\mathrm{C}$ as well as a temperature change rate of $5\,\mathrm{K/min}$ for cooling and $3\,\mathrm{K/min}$ for heating. It can hold the temperature with a variation of $0.3\,°\mathrm{C}$ at a maximum thermal load of $1200\,\mathrm{W}$ at $20\,°\mathrm{C}$. This should highly suffice for our purposes as the target is not expected to radiate a considerable amount of heat (resulting in even smaller temperature variations). We placed the SASEBO-R board together with the mounted ASIC prototype and the DC amplifier inside the chamber, whereas the low-pass filter, the oscilloscope and the power supply units for the board and the amplifier have been placed outside of the chamber. In this regard we had to put two power supply cables for the amplifier and one for the board, as well as one SMA coaxial cable for the amplified static power signals, an RS-232 cable for the communications and a trigger probe cable through a vent in the chamber that was carefully sealed with silicone foam.

The full setup including all main components is depicted in Figure 4 as a photography and in Figure 5 as a schematic.

III. TARGET

The target for our experiments is a 150nm CMOS ASIC prototype chip with a nominal supply voltage of $1.8\,\mathrm{V}$. A photo of the prototyped chip is shown by Figure 6. Among 5 other cores the chip features the PRESENT-80 block cipher realized as a 3-share threshold implementation [34]. Although this work is not focusing on masked implementations or higher-order attacks, we chose this specific core for our investigations to make the results comparable to what has been reported in [27]. In this regard we treat the core as a regular unprotected PRESENT-80 implementation by setting all masks to zero, which corresponds to the `PRNG OFF` mode of operation described in [27]. By doing this we make sure that the core operates deterministically, i.e., for identical plaintexts all intermediate values and the shared output are identical as well. This is explained in more detail in Section VI.


(a) outside

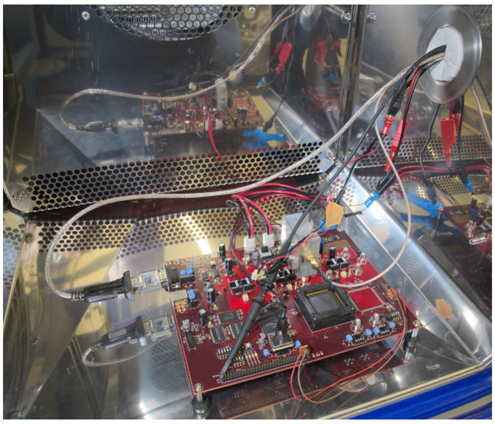

(b) inside

Fig. 4: Photographs of the complete setup including the DC amplifier, the low-pass filter, the board, the oscilloscope, the climate chamber and some power supply units.

PRESENT-80 is an ultra-lightweight block cipher (ISO/IEC 29192-2:2012 standard) that operates on a block size of 64 bits and a key length of 80 bits and consists of 31 computation rounds [13]. The term threshold implementation refers to a masking scheme based on Boolean secret sharing and multi party computation that implements non-linear functions of symmetric block ciphers efficiently in such a way that provable security against first-order power analysis attacks can be guaranteed, even in the presence of glitches [32]. The specific application of this scheme to the PRESENT-80 block cipher is introduced in [34]. Our investigated ASIC core implements the profile 2 of [34]. This profile refers to a serial implementation of PRESENT-80 with a shared data path (with 3 shares) but an unshared key schedule. A schematic of the nibble-serial architecture can be seen in Figure 7.

All intermediate values and data buses are 4-bit wide. As the graphics show, the S-box – which has an algebraic degree

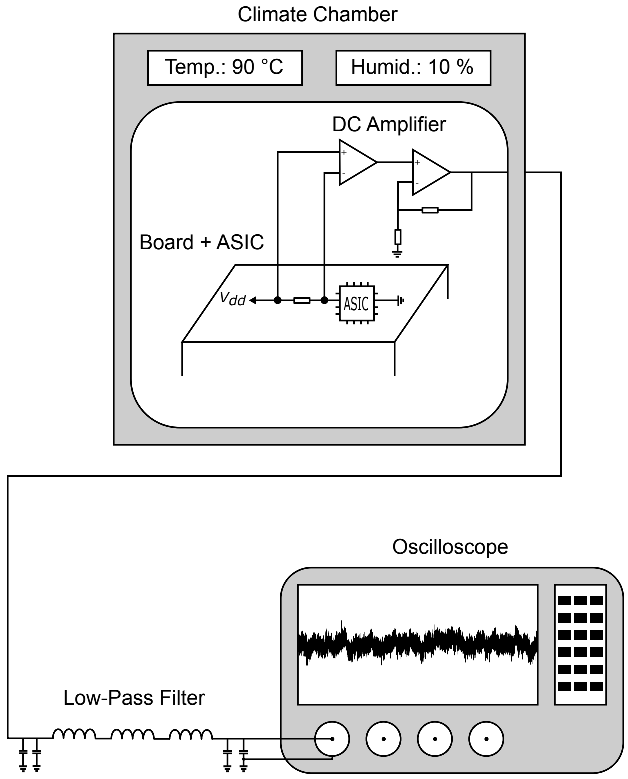
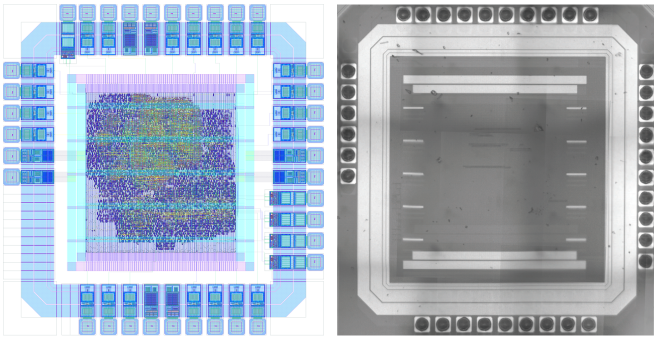


Fig. 5: Schematic of the overall experimental setup including the DC amplifier, the low-pass filter, the oscilloscope and the climate chamber.



(a) layout (b) photo

Fig. 6: ASIC prototype with 6 cores in 150 nm CMOS [27].

of 3 – is decomposed into two non-linear quadratic functions F and G. Those 4-bit boxes are then split into 3 shares each. The three G-boxes are processed at the same time in the ASIC and each of them receives 2 inputs out of the 3 data shares. The corresponding outputs are stored into registers. Afterwards, the three F-boxes are evaluated in parallel. The 4-bit words of the round state are processed in a pipelined manner by one instance of the shared S-box. Thus, (due to the register between the F and G functions) 17 clock cycles are required to evaluate the complete substitution layer of the cipher for one round. After the last nibble of the shares has been processed, the outputs are routed according to the linear layer (PLayer) of the cipher and saved into the register again. Therefore, each full computation round of the PRESENT-80 cipher takes 18 clock cycles on the investigated ASIC core.

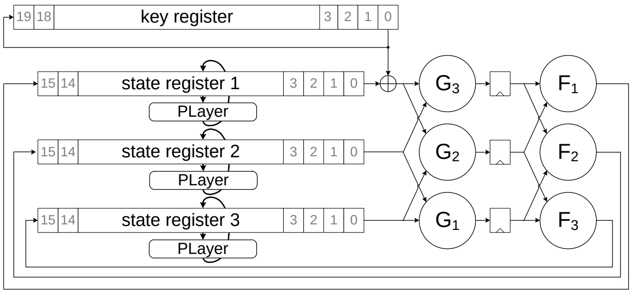


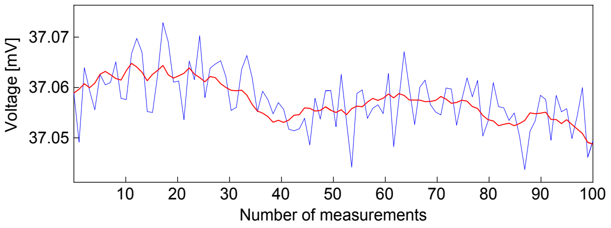
Fig. 7: Nibble-serial architecture of the PRESENT-80 threshold implementation core.

The initial masking of the input (with all zeros in our case) as well as the unmasking of the output are performed on the chip itself. Hence the communication with the ASIC is performed in an unshared, conventional manner.

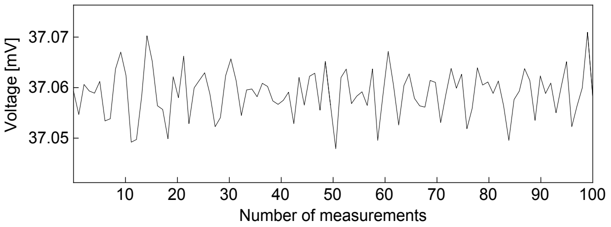
IV. MEASUREMENT PROCEDURE

In order to measure the static currents, we executed the following procedure. At the specific clock cycle, where the targeted intermediate value is processed, the clock signal of the PRESENT core is stopped and all other input signals to the ASIC are kept constant at a deterministic value. This idle state of the target is held for an arbitrarily long time interval during which the static power consumption of the device can be measured before the clock signal is switched back on. Thus, in our experiments recording the static leakage traces requires a stronger attacker model than it would be required for a classical power analysis, as full control over the clock signal is necessary. The power consumption values that are obtained in the mentioned time interval are then averaged to a singular value. Since the leakage currents are not supposed to change during that period, all occurring variations are noise and can be averaged out. This technique is called intra-trace averaging and constitutes one major advantage of static power analysis in comparison to classical attacks when control over the clock signal is obtained (see [35]). Due to the very high gain of our developed DC amplifier (×1000) and the very low cutoff-frequency of our low pass filter (100 Hz), a significant impact of the memory effect (described in [30]) on the measurement quality can be observed. The sudden drop of the power consumption when the clock signal is stopped influences the measured static power values for up to the next 20 ms. Hence the first 20 ms of the idle state are discarded and not included in our measurements. After that period, the actual measurement interval starts. This procedure is illustrated in Figure 3 exemplarily for a measurement with and a measurement without low-pass filter applied. The time period which is denoted by T_1 corresponds to the first 20 ms of the idle state that are discarded due to the influence of the memory effect. The second time period T_2 indicates the measurement interval.

In contrast to [27] we also employed a simple post-processing technique. The idea is to filter out the long-term temperature-induced variations of the static power



(a) original measurements and moving-average curve



(b) altered measurements

Fig. 8: Illustration of the post-processing technique for 100 exemplary measurements at $20\,^{\circ}\mathrm{C}$, $1.8\,\mathrm{V}$, measurement interval of $10\,\mathrm{ms}$ and window-size of 8.

consumption over time, i.e., over a set of measurements (as opposed to noise that is included in single measurements). Quite obviously the climate chamber requires a lot more activity of its regulation units to maintain a constant temperature when it is set to a value far above or below the temperature of the room it is located in. These activities can be observed as low frequency noise along the whole set of measurements. Our post-processing step is therefore to apply a simple moving-average filter onto the measurement set by using the Matlab function `filter()`[4]. The effect of such filtering is depicted in Figure 8. The blue plot in Figure 8a corresponds to a set of 100 unaltered measurements as they were recorded from the oscilloscope. The red curve corresponds to the moving-average that is generated by the Matlab `filter()` function. The subtraction of the moving-average from the original measurements results in the black graph in Figure 8b and constitutes the resulting measurement set after the post-processing. We tested several window-sizes as parameter for the `filter()` function and revealed that a window-size of 8 leads on average to the best results on our measurements. In general we observed that for measurements with a long measurement interval, i.e., the more time-demanding ones, a smaller window-size led to optimal results, while for the measurements with a short interval larger windows were more successful. However, to keep all results comparable we have always used a window-size of 8 for the experiments that are presented in Section VI. Although the initial purpose of the post-processing was to improve the measurement quality at extreme temperatures (like $-20\,^{\circ}\mathrm{C}$ or $90\,^{\circ}\mathrm{C}$) we observed that it has a positive influence on the measurements in all cases, even at room temperature.

Figure 9 depicts the whole measurement procedure as a simple flowchart.

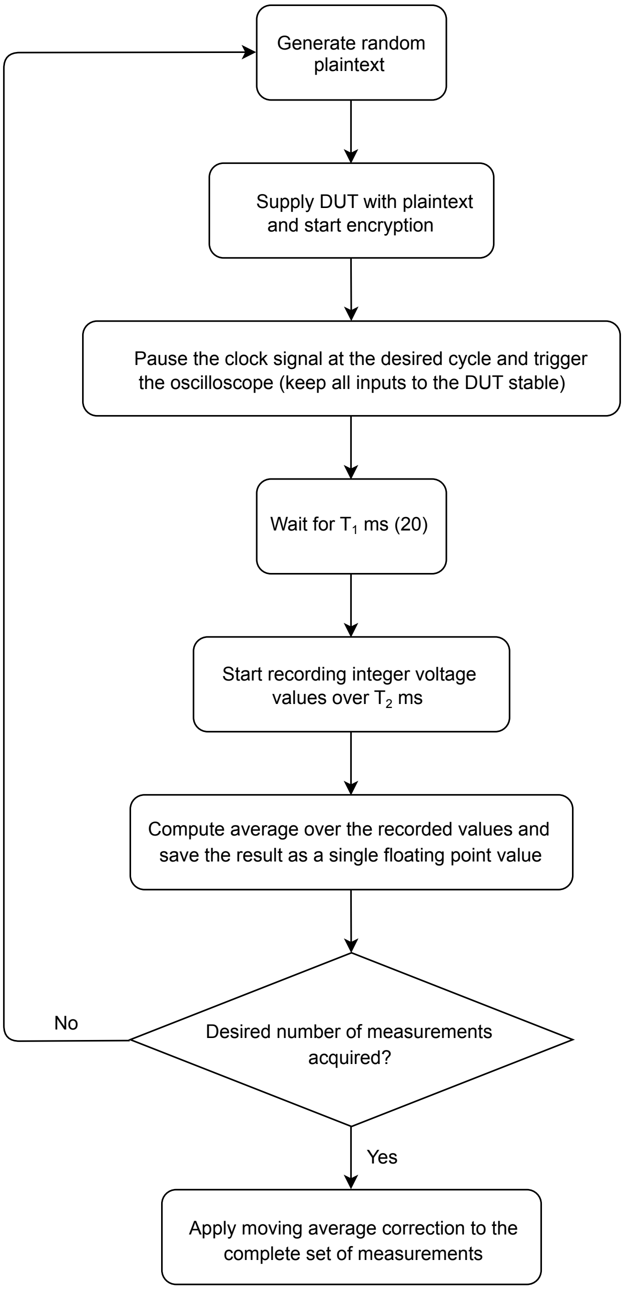

Generate random plaintext
Supply DUT with plaintext and start encryption
Pause the clock signal at the desired cycle and trigger the oscilloscope (keep all inputs to the DUT stable)
Wait for T1 ms (20)
Start recording integer voltage values over T2 ms
Compute average over the recorded values and save the result as a single floating point value
Desired number of measurements acquired?
No
Yes
Apply moving average correction to the complete set of measurements

Fig. 9: Flowchart describing the measurement procedure.

V. EVALUATION TOOLS AND METRICS

Section VI analyzes in detail how the investigated measurement factors influence the amount of information that is included in, or can be extracted from, the corresponding side-channel measurements. In this regard several evaluation tools and metrics which are common in the side-channel analysis literature are used. We introduce these tools and metrics shortly in this section.

A. *Signal-to-Noise Ratio*

The signal-to-noise ratio (SNR), introduced by Mangard in 2004 [25], is one of the most common metrics to quantify

---

[4] We also tested other filters in Matlab, for example a butterworth high-pass filter, but achieved inferior results.

the quality of side-channel measurements and to determine the points of interest in a dynamic power trace. The corresponding formula is given in Equation 1.

$$SNR = \frac{Var(Signal)}{Var(Noise)} \tag{1}$$

The variance of the signal is defined as any variation in the measurements (e.g., power consumption or electromagnetic radiation) that is caused by the targeted intermediate value, while the variance of the noise describes all further variations in the traces that are not caused by this value. To assess those parameters for a specific sample point in a set of traces (acquired for random inputs), one has to sort the traces into a number of groups corresponding to the specific value the targeted intermediate result attains (e.g., 16 distinct groups for a 4-bit intermediate value). In the case of (SPN-based) block ciphers, for example, which make use of a bijective non-linear mapping and key addition, one can directly calculate the SNR for the intermediate values after the first round (resp. before the last round) from the input (resp. output) of the cipher. The variance of the signal is then calculated as the variance of the means of the individual groups, while the variance of the noise can be calculated as the overall mean over the variances of the individual groups.

B. *Correlation Power Analysis*

Correlation power analysis (CPA) was introduced by Brier et al. at CHES 2004 [14] to overcome some drawbacks of classical DPA. The main advantage is that a power model can be used to create a hypothesis for the leakage of a full intermediate value instead of targeting only a single bit at a time. This hypothetical power consumption is then compared to the actual power consumption by means of a (Pearson) correlation coefficient, which measures the linear dependency. The corresponding formula for two discrete vectors X, Y is given in Equation 2. The mean of the two vectors is denoted by $\overline{X}, \overline{Y}$.

$$\rho = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}} \tag{2}$$

In a successful attack the highest correlation coefficient directly translates to a correctly guessed part of the key. Common models, in addition to the identity model, include for example the Hamming weight of an intermediate result or the Hamming distance between two processed values.

C. *Measurements to Disclosure*

Historically, the number of traces that are required to perform a successful attack on an implementation (DPA, CPA, ...) has been the most common metric to assess the resistance of a device against such attacks. Nowadays there exist tools, like for example the non-specific Welch's t-test (see [37] for a detailed methodology), that are able to evaluate the leakage of a device without performing any specific attack and without being dependent on a correct choice of a leakage model. However, those metrics fail to provide information about the hardness of an actual key recovery, which makes them unsuitable for a variety of purposes. In our case, for example, a suitable model and a successful attack are already known and the goal is to evaluate how large the impact of changes in a specific operating parameter on the exploitability of the implementation is. In this case the number of required measurements to disclose the correct key is still the most preferable metric. To the best of our knowledge the term ”measurements to disclosure” together with its abbreviation ”MTD” has been first used and defined as a metric by Kiri et al. at CHES 2005 [41]. The authors describe it as the cross-over point between the correlation coefficient for the correct key and the maximum correlation coefficient among all wrong key guesses when plotting the coefficients for all key guesses over the number of samples considered. For a relation between the SNR and the MTD see for example [18].

D. *Success Rate*

The success rate of a power analysis attack is the probability that the attack succeeds in recovering the correct key candidate by isolating it from a restricted set of key guesses [39], [36]. The most straightforward option to evaluate the success rate of an attack is to simply perform the attack multiple times. Many efforts have been devoted to the exploration of more efficient ways to estimate the success rate rather than this empirical one (the interested reader is referred to [39], [36]). However, in this work we do indeed perform the attack multiple times on disjoint subsets of a larger measurement set whenever success rates are reported

VI. MEASUREMENT FACTORS

In this section we present measurement results that have been acquired over a time period of roughly four months and represent the equivalent of more than two months of non-stop data acquisition[5]. We build upon the results that are reported in [27] and try to improve the attacks on unprotected implementations in terms of the required number of measurements by controlling the operating parameters. While the influence of the measurement interval on the noise has been mentioned in [35] and [27], although not as detailed as in this work, the influence of the operating conditions temperature and supply voltage has not yet been reported based on practical side-channel measurements. In order to keep all results comparable we target the same threshold implementation prototype chip as [27] and operate the same PRESENT core.

All reported values for the measurements to disclosure (MTD) metric refer to a standard Correlation Power Analysis (CPA) attack [14] on the combined Hamming weight of the outputs of the three F-boxes that can be seen in Figure 7 (4 bit × 3 boxes = 12 bit), effectively targeting one key nibble (4 bit) at a time. The Hamming weight model has been proposed and verified as a suitable power model for static power side-channel attacks in [7] for the first time. Other popular models from the dynamic power domain that rely

---

[5] Between the acquisition of the different sets the ASIC and parts of the setup have to rest at target climate to adopt the temperature accordingly before the next set can be recorded.

on transitional effects, such as the Hamming distance model for example, are not applicable since static power attacks capture a stable state and not a transition between states. Let x_1, x_2, x_3 be the outputs of state register 1, 2 and 3 in Figure 7 respectively. Let k be the targeted nibble of the round key. The power model is then computed as shown in Equation 3

$$\begin{aligned} &\mathrm{HW}(F_1(G_3(x_1 \oplus k, x_2), G_2(x_1 \oplus k, x_3)))+ \\ &\mathrm{HW}(F_2(G_3(x_1 \oplus k, x_2), G_1(x_2, x_3)))+ \\ &\mathrm{HW}(F_3(G_2(x_1 \oplus k, x_3), G_1(x_2, x_3))) \end{aligned} \tag{3}$$

The leakage currents of all parts of the circuit that are not directly targeted, such as the G-boxes and the state registers contribute to the algorithmic noise. To perform this kind of attack on the targeted threshold implementation core it is required to have knowledge of the masks that are involved in the computation, which a regular power analysis adversary against a securely implemented masking scheme would (ideally) not have. But, as mentioned before, we operate the core by setting all masks to zero, which allows us to predict the exact intermediate values, under the correct key hypothesis, that are actually processed by the circuit. This corresponds to the usual adversarial situation when targeting an unprotected PRESENT implementation, with the only difference that each S-box output corresponds to a 12 bit value instead of a 4 bit one (which certainly eases the attack). We would like to stress here that the whole purpose of our practical evaluation in this section is to investigate the influence of the measurement factors on the success of power analysis attacks. We do in no way claim that the presented attacks on the targeted implementation with fixed masks are a realistic scenario for any adversary against a real-world device. We just aim for conformity with the previous work described in [27] and restrict all our claims to unprotected implementations.

For all three measurement factors that are investigated in this section we provide estimations of the noise to determine whether it is influenced by the altered parameters. Additionally we report the number of measurements that are required for a successful recovery of one key-nibble by means of a CPA attack. From both of those values the influence of the operating parameters on the signal becomes obvious. For instance, due to the known anti-proportional relationship between the signal-to-noise ratio (SNR) and the measurements to disclosure (MTD) [18], a constant noise level and a lowered MTD indicate an increased signal. Similarly a decreased noise level and a constant MTD indicate a decreased signal. Estimated signal values would therefore be redundant. Furthermore the signal-to-noise ratio itself is mostly used to determine the points of interest in a dynamic power trace, which is not required in static power analysis attacks like the ones reported in this work, since each trace corresponds to a single measured value anyway. The only reasonable use case of the SNR in this context would be to identify which intermediate value is possibly leaked in a particular clock cycle, in order to perform the attack on this value or find the exact clock cycle that needs to be targeted. However, this is not a necessity in our experiments as we have detailed information about the implementation and know exactly which intermediate value is processed within which clock cycle.

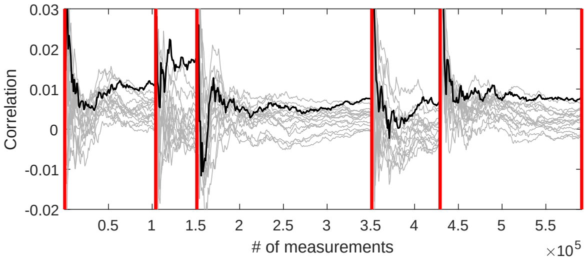


Fig. 10: Visual depiction of our technique to use disjoint subsets of one large set of traces to disclose the key multiple times. The shown MTD values (achieved for 90 °C, 1.98 V, 10 ms) from left to right are 85 000, 28 000, 181 000, 59 000, 143 000.

At all temperature-voltage combinations we have collected at least 5 million traces with a constant measurement interval of 10 ms. In all cases we have tried to disclose the correct key nibble multiple times using this set. For example if a CPA succeeds by indicating the highest (absolute) correlation value for the correct key candidate after roughly 150 000 traces and this correlation value remains to be the largest one among all candidates for at least further 20 000 traces (to avoid false positives) we state that the first MTD value for the attack on this set is 150 000. Then we ignore the first 170 000 traces and repeat the process from trace 170 001. In other words we use disjoint subsets of the whole measurement set to disclose the key multiple times in order to obtain multiple MTD values. This enables us to report average MTD values and success rates for each set (only if the average MTD lies below 2.5 million, otherwise only one disclosure is possible). In case of the most effective temperature voltage combination and a measurement interval of 10 ms for example we were able to disclose the correct key 48 times using 5 million traces. We have illustrated this procedure for 5 consecutive MTD values in Figure 10. The average number of measurements to disclosure (MTD) for this snippet of traces would be 95 200 and the number of measurements to reach a success rate larger than 50 % would be 85 000. Admittedly, the single MTD values may not be fully independent, although being computed on disjoint subsets of the whole set, because strictly speaking they do not originate from statistically independent experiments. However, we have observed that assuming the independence of the particular subsets leads to sound and reproducible results. Furthermore, as apparent from Figure 10, the MTD values can vary quite significantly. Thus, we believe that averaging several of those values and reporting success rates leads to more meaningful results and is therefore superior to only reporting single MTD values (like for example in [27]).

The 8 additional sets of traces that we have recorded at the most effective temperature voltage combinations, but for different measurement intervals (1 ms to 200 ms), have a smaller cardinality (because of the significantly longer run time for larger intervals). Hence, unfortunately, we are only able to present a single MTD value per set here.

A. Factor: Temperature

According to [21] all leakage effects that are based on solid-state physics, such as subthreshold leakage and diode currents, show extreme thermal dependencies. The subthreshold current for example, which is the dominating source of static power consumption in our technology, depends exponentially on the temperature [21]. From simulated measurements it is known that the factor between the leakage currents for any two input vectors to digital standard cells is roughly maintained [16], [24], [8], [5], [33]. Hence the difference between the classes in a power analysis attack should increase when raising the temperature, which corresponds to an increase of the signal in a side-channel attack. For most adversaries against embedded systems it should be feasible to influence the temperature of the environment, since physical access is part of the adversary model. In our experiments we put the target device into a climate chamber, like explained in Section II, and fixed the temperature and the relative humidity to a determined value. The supply voltage was kept at 1.8 V, which is the nominal supply voltage of the chip. In the range of −20 °C to 0 °C we raised the temperature in steps of 5 °C between different sets and were not able to control the humidity. In the range of 0 °C to 80 °C we raised the temperature in steps of 10 °C and kept a constant relative humidity of 20 %. Finally we measured two sets for 85 °C and 90 °C at a constant relative humidity of 10 %. The humidity was always set in order to have a small impact on the experiments. In general we chose a rather dry climate in order to not face any problems with condensation of water vapor at the electronic components. Furthermore, we observed that the climate chamber is able to keep the temperature more stable when the relative humidity is controlled as well. Since the absolute humidity increases when the relative humidity is kept at a fixed value and the temperature is increased we needed to reduce the relative humidity from 20 % to 10% for the temperatures above 80 °C.

As explained above, we expect the signal to increase for higher temperatures. However, this only eases an attack if the noise is not equally (or greater) affected. Hence as a first step we estimated the variance of the noise for all of the 19 temperature sets, according to the description in Section V over the first 50 000 measured values. The results can be seen in Figure 11. Apparently the noise increases approximately in a linear fashion with the temperature. To emphasize this a first degree (linear) polynomial curve was fitted to the data. At a first glance, this behavior appears to be a negative result when trying to increase the exploitable information in the measurements by raising the temperature. On the other hand, this result was expected, since not only the static currents associated with the targeted intermediate value are increased, but the algorithmic noise is supposed to grow in a similar fashion. Additionally, it can be expected that the measurements include more thermal noise when raising the temperature. The important question is here whether the increase of the signal is large enough to overcome this linear increase of the noise. To answer this question we performed CPA attacks using the Hamming weight of the 12-bit output of the F-boxes. As explained before, each set has a cardinality of (at least) 5 million measurements. Whenever possible we tried to disclose the correct key candidate multiple times by using disjoint subsets of the whole measurement set. As a result we are able to report how many traces are required to reach a certain success rate, as shown by Figure 12. Please note that no temperatures below 10 °C are plotted here. The reason for this is that for these temperatures the correct key candidate could not be disclosed even a single time with 5 million measurements. For the higher temperatures it becomes obvious that the number of measurements that are required to extract the same amount of information is reduced in an exponential manner. In order to emphasize this we have added an exponentially fitted curve. The same kind of exponential decrease can be observed in Figure 13 for the average MTD values. Despite of the linear increase of the noise for increasing temperatures the MTD values are exponentially decreasing. This confirms that the signal is exponentially increased by raising the temperature and more generally that the signal-to-noise ratio is exponentially increased. Since the signal is

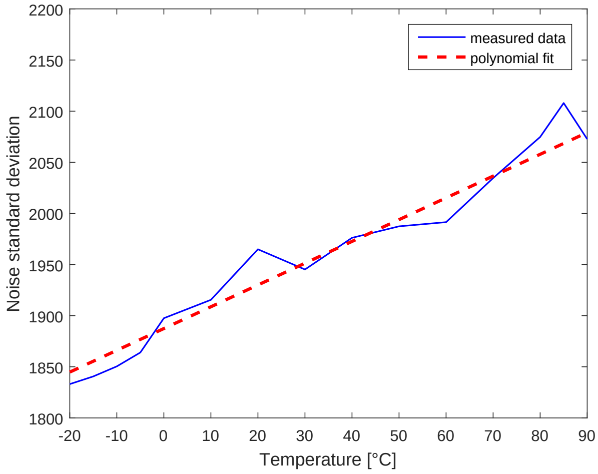


Fig. 11: Estimated noise standard deviation for temperatures between −20 °C and 90 °C.

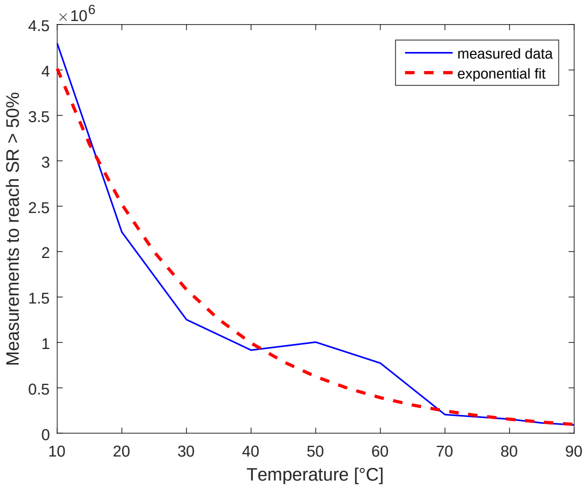


Fig. 12: Number of measurements required to overcome a success rate of 50 % for temperatures between 10 °C and 90 °C.

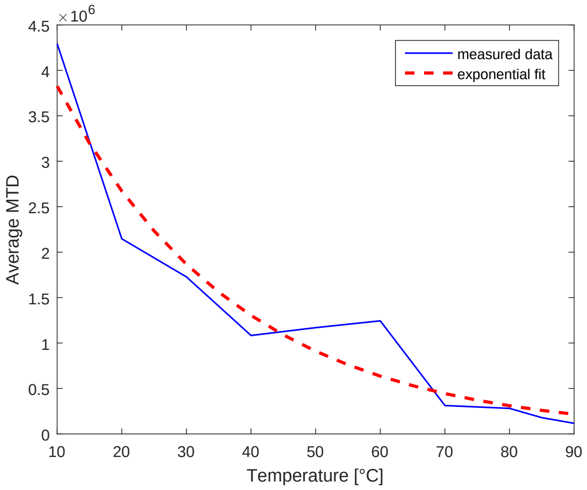


Fig. 13: Average number of measurements required to disclose the correct key candidate for temperatures between 10 °C and 90 °C.

growing much stronger than the noise, it is – in theory – possible to raise the temperature up to a point where each noise source apart from the algorithmic noise becomes negligible.

B. Factor: Supply Voltage

Apart from the obvious linear dependency of the static power consumption on the supply voltage ($P_{leak} = I_{leak} \cdot V_{DD}$) there are several other dependencies listed in [21]. The gate leakage for example is doubled when the supply voltage is increased by 100 mV. The drain induced barrier lowering (DIBL) effect leads to a reduction of the effective threshold voltage when the supply voltage is increased [21]. This on the other hand increases the subthreshold conduction exponentially. Finally, the gate induced drain leakage (GIDL) effect increases the junction current exponentially in a specific region of the supply voltage. However, we do not expect a very large dependency of the static currents of our 150 nm ASIC on the supply voltage for mainly two reasons. First of all the gate leakage and the junction current are (more or less) negligible sources of static dissipation in technologies larger than 100 nm. Secondly the DIBL effect is only relevant for short channel length as well and even for a 65 nm technology the supply voltage needs to be increased by 192 mV to raise the overall leakage by 10% [21]. In total we expect at least a linear decrease in the number of required measurements to disclosure when increasing the supply voltage. Concerning the feasibility of controlling the supply voltage of the device under test it should be kept in mind that a regular power analysis adversary is supposed to place (or find) a resistor in the GND or V_{DD} path of the device under test and to measure the voltage drop across this component by means of a digital sampling oscilloscope. Thus, it should be easily possible to influence the supply voltage with this kind of capabilities. In our experiments we adjusted the supply voltage by a potentiometer in the feedback path of the linear voltage regulator on the measurement board and verified the correct setting while the setup was present in the climate chamber at target temperature.

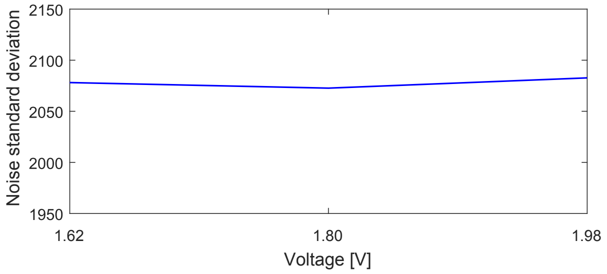


Fig. 14: Estimated noise standard deviation for supply voltages between 1.62 V and 1.98 V and temperature of 90 °C.

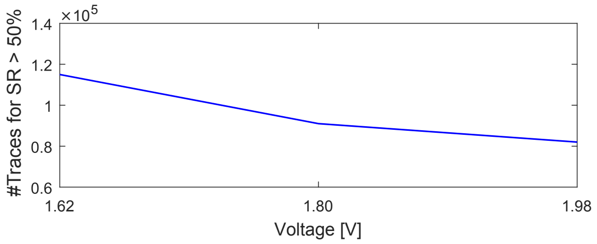


Fig. 15: Number of measurements required to overcome a success rate of 50 % for supply voltages between 1.62 V and 1.98 V and temperature of 90 °C.

Since it seemed infeasible to determine the influences of the supply voltage for all temperatures, we chose the most successful temperature in terms of success rate, i.e., 90 °C, and changed the supply voltage by 10% in comparison to the nominal supply voltage in both, positive and negative direction. We did not evaluate more extreme changes in order to not damage the chip. The results for the noise estimation at 90 °C can be seen in Figure 14. Almost no voltage-induced change in the noise level can be observed. If at all, the noise is slightly increased. When taking a look at the MTD values in Figure 15 and Figure 16 it can be seen that the attacks become slightly more successful by raising the voltage, but the effect is much less drastic than what could be observed for the temperature. Obviously, when taking only three data points into account, it is difficult to make a statement about the type of dependency that could be observed, which is why we leave this open to interpretation. Finally, please note that, in contrast to the temperature, the supply voltage also has a direct (quadratic) influence on the dynamic power consumption.

C. Factor: Measurement Interval

In [27] a clear trade-off between intra-trace averaging and inter-trace averaging is observed. In particular, by stretching the measurement interval the noise can be reduced and the attacks succeed with fewer traces, but the time to acquire a specific number of traces is also increased. During the period when the clock signal is stopped, none of the intermediate values of the computation that are currently processed by the circuit are supposed to change. Thus, the same transistors in the respective CMOS gates are active or inactive and the leakage currents of the whole chip should stay the same. All variations that can be observed in the measured signal during

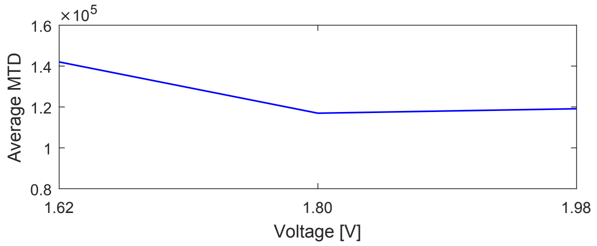


Fig. 16: Average number of measurements required to disclose the correct key candidate for supply voltages between 1.62 V and 1.98 V and temperature of 90 °C.

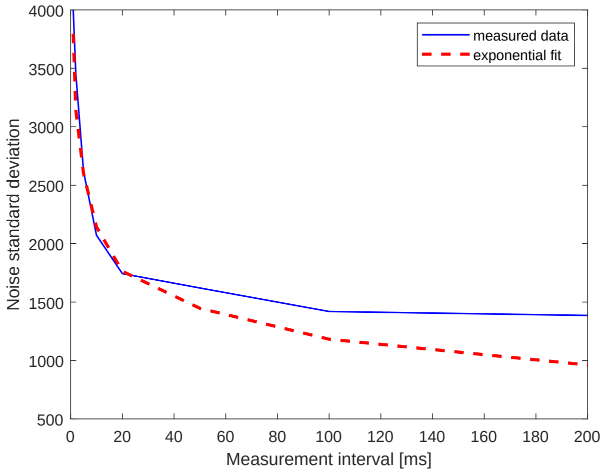


Fig. 17: Estimated noise standard deviation for measurement intervals between 1 ms and 200 ms at a supply voltage of 1.98 V and a temperature of 90 °C.

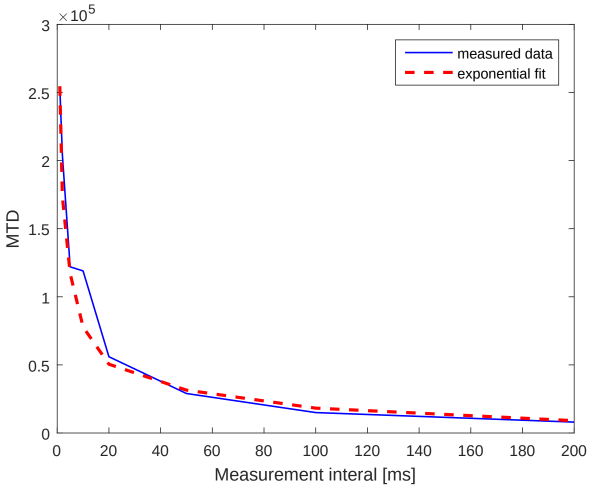


Fig. 18: Number of measurements required to disclose the correct key candidate for measurement intervals between 1 ms and 200 ms at a supply voltage of 1.98 V and a temperature of 90 °C.

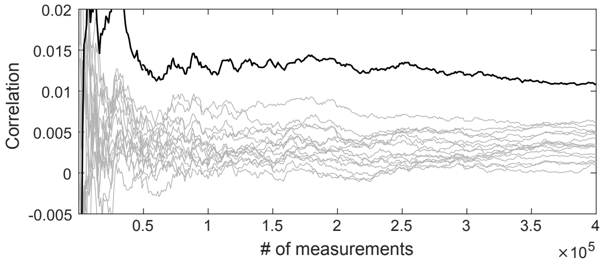


Fig. 19: CPA on Hamming weight of 12-bit F-box output for a measurement interval of 200 ms at a supply voltage of 1.98 V and a temperature of 90 °C.

that period are electronic noise (e.g., from the power supply or conducted and radiated emissions [26]) and by extending the time interval that is averaged into a single value the influence of the noise can be minimized. Note that this can not be achieved solely by averaging over more sample points, for example when increasing the sampling frequency while the interval stays the same. Using our setup the static power signal is usually sufficiently well sampled at moderate sampling frequencies (e.g., 1 MS/s) due to the low bandwidth of the DC amplifier and the low-pass filter. However, some of the noise sources have a very low frequency and therefore require to be sampled over a time period of a certain length in order to be eliminated.

Independent of the time it takes to acquire a set of traces we want to investigate whether we can apply the noise reduction techniques through averaging onto the measurements at the most informative temperature-voltage combination. In particular, we want to take the measurement setting that requires the least amount of traces for a successful key recovery due to an already increased signal (90 °C and 1.98 V) and try to additionally decrease the noise by stretching the interval as much as possible. Therefore we acquired trace sets for the following measurement intervals: 1 ms, 2 ms, 5 ms, 10 ms, 20 ms, 50 ms, 100 ms, 200 ms. The results of the noise estimation can be seen in Figure 17. The noise level decreases exponentially when linearly increasing the length of the measurement interval until roughly 20 ms. Afterwards the development seems to stagnate. This can be observed when comparing it to the exponentially fitted curve over the data points. However, in Figure 18 it can be seen that the number of required measurements for a key recovery decreases even beyond the 20 ms in an exponential fashion. In any way the decrease of the noise seems to be lower bounded by the amount of algorithmic noise in the measurements, since this part cannot be averaged out by intra-trace averaging (see [27]). For the sake of completeness we should mention that we also performed the experiments with even longer measurement intervals. However, neither the noise nor the MTD could be reduced any further (due to the lower bound). In fact, they even started to slowly increase again. Our assumption is that for very long measurement intervals (>200 ms) the temperature-induced variations coming from the active regulation of the climate chamber start to affect single measurements, instead of being present between traces in a set. Hence, we achieved the overall best results at a temperature of 90 °C, a supply voltage of 1.98 V and a measurement interval of 200 ms. The result of the CPA attack under this setting can be seen in Figure 19. It is shown that 8 000 traces are required to identify the correct key candidate. This in fact corresponds to the number of traces that

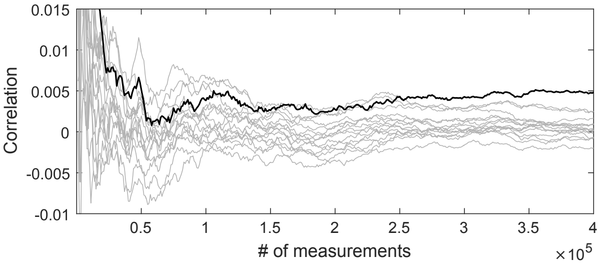


Fig. 20: CPA on Hamming weight of 12-bit F-box output for a measurement interval of $1\,\mathrm{ms}$ at a supply voltage of $1.98\,\mathrm{V}$ and a temperature of $90\,^{\circ}\mathrm{C}$.

the corresponding CPA on the dynamic power measurements (on the same ASIC chip) required in [27]. For comparison purposes, the result of the same attack for a measurement interval of $1\,\mathrm{ms}$ is given in Figure 20 (multiple examples for a measurement interval of $10\,\mathrm{ms}$ are given in Figure 10).

VII. CONCLUSIONS

In this work we have presented an extensive case study on the effects of the three measurement factors temperature, supply voltage and measurement interval on the amount of information that can be extracted from static power measurements. We are able to show that by controlling either the temperature or the measurement interval (in case of clock control) the number of traces that are required for a successful key recovery can exponentially be reduced. Additionally we observed that modifying the supply voltage at least marginally eases such attacks as well. In particular by adjusting all three parameters an adversary can theoretically end up with a set of traces that only contains algorithmic noise. We conclude that the existence of the investigated measurement factors and their, in some cases, exponential impact on the success of attacks further strengthen the position of the static power side channel as a realistic target for adversaries against cryptographic hardware. In this regard we would like to encourage research and industry to incorporate static power attacks into their security evaluation and certification processes.

Considering that our target ASIC was manufactured in a rather old $150\,\mathrm{nm}$ technology, the results are even more astonishing. The static power dissipation in this technology is still several times smaller than the dynamic power consumption. But still, only by controlling some measurement factors, a successful static power analysis attack on an unprotected implementation could be performed with as many traces as a corresponding dynamic power analysis on the same target. According to [4] the data dependency of the static current in digital standard cells stays the same for smaller technology sizes while especially the subthreshold leakage increases more than linearly. Additionally, the exponential dependencies of some static power sources on the supply voltage become only relevant for more advanced technologies, which again favors the adversaries. Hence, we suspect that the results presented in this work are even more drastic for smaller feature sizes.

As a suggestion for future work on this topic, quite obviously test chips in more advanced technology generations need to be investigated. Additionally, it has to be verified whether control over these measurement factors enables a similar improvement of higher-order attacks against securely masked implementations. Finally, effective countermeasures need to be constructed to counteract the exploitation of this emerging side channel.

ACKNOWLEDGMENT

The authors would like to acknowledge Axel Poschmann for the hardware designs and Stefan Heyse for his help on taping out the prototype chip. The work described in this paper has been supported in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and through the project 271752544 "NaSCA: Nano-Scale Side-Channel Analysis".

REFERENCES

[1] Analog Devices AD8421 Data Sheet Rev. 0. http://www.analog.com/media/en/technical-documentation/data-sheets/AD8421.pdf.

[2] Analog Devices AD8676 Data Sheet Rev. C. http://www.analog.com/media/en/technical-documentation/data-sheets/AD8676.pdf.

[3] Side-channel Attack Standard Evaluation Board SASEBO-R Specification – Version 1.0. http://www.risec.aist.go.jp/project/sasebo/download/SASEBO-R_Spec_Ver1.0_English.pdf. Research Center for Information Security, National Institute of Advanced Industrial Science and Technology, Japan.

[4] Z. Abbas and M. Olivieri. Impact of technology scaling on leakage power in nano-scale bulk CMOS digital standard cells. *Microelectronics Journal*, 45(2):179–195, February 2014.

[5] M. Alioto, S. Bongiovanni, M. Djukanovic, G. Scotti, and A. Trifiletti. Effectiveness of Leakage Power Analysis Attacks on DPA-Resistant Logic Styles Under Process Variations. *Transactions on Circuits and Systems I: Regular Papers*, 61(2):429–442, February 2014.

[6] M. Alioto, S. Bongiovanni, G. Scotti, and A. Trifiletti. Leakage Power Analysis Attacks Against a Bit Slice Implementation of the Serpent Block Cipher. In *MIXDES 2014*, pages 241–246. IEEE, June 2014.

[7] M. Alioto, L. Giancane, G. Scotti, and A. Trifiletti. Leakage Power Analysis Attacks: Well-Defined Procedure and First Experimental Results. In *ICM 2009*, pages 46–49. IEEE, December 2009.

[8] M. Alioto, L. Giancane, G. Scotti, and A. Trifiletti. Leakage Power Analysis Attacks: A Novel Class of Attacks to Nanometer Cryptographic Circuits. *Transactions on Circuits and Systems I: Regular Papers*, 57(2):355–367, February 2010.

[9] D. Bellizia, S. Bongiovanni, P. Monsurrò, G. Scotti, and A. Trifiletti. Univariate Power Analysis Attacks Exploiting Static Dissipation of Nanometer CMOS VLSI Circuits for Cryptographic Applications. *Transactions on Emerging Topics in Computing*, 5(3):329–339, May 2016.

[10] D. Bellizia, D. Cellucci, V. D. Stefano, G. Scotti, and A. Trifiletti. Novel Measurements Setup for Attacks Exploiting Static Power using DC Picoammeter. In *ECCTD 2017*, pages 1–4. IEEE, September 2017.

[11] D. Bellizia, M. Djukanovic, G. Scotti, and A. Trifiletti. Template attacks exploiting static power and application to CMOS lightweight cryptohardware. *International Journal of Circuit Theory and Applications*, 45(2):229–241, August 2017.

[12] D. Bellizia, G. Scotti, and A. Trifiletti. Implementation of the PRESENT-80 Block Cipher and Analysis of its Vulnerability to Side Channel Attacks Exploiting Static Power. In *MIXDES 2016*, pages 211–216. IEEE, June 2016.

[13] A. Bogdanov, L. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *CHES 2007*, volume 4727 of *LNCS*, pages 450–466. Springer, September 2007.

[14] E. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with a Leakage Model. In *CHES 2004*, volume 3156 of *LNCS*, pages 16–29. Springer, August 2004.

[15] M. Djukanovic, D. Bellizia, G. Scotti, and A. Trifiletti. Multivariate Analysis Exploiting Static Power on Nanoscale CMOS Circuits for Cryptographic Applications. In *AFRICACRYPT 2017*, volume 10239 of *LNCS*, pages 79–94. Springer, May 2017.

[16] J. Giorgetti, G. Scotti, A. Simonetti, and A. Trifiletti. Analysis of Data Dependence of Leakage Current in CMOS Cryptographic Hardware. In *GLSVLSI 2007*, pages 78–83. ACM, March 2007.

[17] A. Gornik, A. Moradi, J. Oehm, and C. Paar. A Hardware-Based Countermeasure to Reduce Side-Channel Leakage: Design, Implementation, and Evaluation. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 34(8):1308–1319, 2015.

[18] S. Guilley, H. Maghrebi, Y. Souissi, L. Sauvage, and J.-L. Danger. Quantifying the Quality of Side-Channel Acquisitions. *COSADE 2011*, pages 16–28, February 2011.

[19] B. Halak, J. Murphy, and A. Yakovlev. Power Balanced Circuits for Leakage-Power-Attacks Resilient Design. In *SAI 2015*, pages 1178–1183. IEEE, July 2015.

[20] N. hao Zhu, Y. Zhou, and H. Liu. A Standard Cell-Based Leakage Power Analysis Attack Countermeasure Using Symmetric Dual-Rail Logic. *Journal of Shanghai Jiaotong University*, 19(2):169–172, April 2014.

[21] D. Helms. *Leakage Models for High Level Power Estimation*. PhD thesis, Carl von Ossietzky Universität Oldenburg, 2009.

[22] S. S. Immaculate and K. Manoharan. Analysis of Leakage Power Attacks on DPA Resistant Logic Styles: A Survey. *International Journal of Computer Science Trends and Technology*, 2(5):136–141, September 2014.

[23] P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *CRYPTO 1999*, LNCS, pages 388–397. Springer, December 1999.

[24] L. Lin and W. Burleson. Leakage-Based Differential Power Analysis (LDPA) on Sub-90nm CMOS Cryptosystems. In *ISCAS 2008*, pages 252–255. IEEE, May 2008.

[25] S. Mangard. Hardware Countermeasures against DPA - A Statistical Analysis of Their Effectiveness. In *CT-RSA 2004*, volume 2964 of *LNCS*, pages 222–235. Springer, February 2004.

[26] S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, 2007.

[27] T. Moos, A. Moradi, and B. Richter. Static Power Side-Channel Analysis of Threshold Implementation Prototype Chip. In *DATE 2017*, pages 1324 – 1329. IEEE, March 2017.

[28] A. Moradi. Side-Channel Leakage through Static Power — Should We Care about in Practice? In *CHES 2014*, volume 8731 of *LNCS*, pages 562–579. Springer, September 2014.

[29] A. Moradi and G. Hinterwälder. Side-Channel Security Analysis of Ultra-Low-Power FRAM-Based MCUs. In *COSADE 2016*, volume 9689 of *LNCS*, pages 239–254. Springer, 2016.

[30] A. Moradi and O. Mischke. On the Simplicity of Converting Leakages from Multivariate to Univariate – Case Study of a Glitch-Resistant Masking Scheme. In *CHES 2013*, volume 8086 of *LNCS*, pages 1–20. Springer, August 2013.

[31] A. Moradi and T. Schneider. Improved Side-Channel Analysis Attacks on Xilinx Bitstream Encryption of 5, 6, and 7 Series. In *COSADE 2016*, volume 9689 of *LNCS*, pages 71–87. Springer, 2016.

[32] S. Nikova, V. Rijmen, and M. Schläffer. Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. *Journal of Cryptology*, 24:292–321, April 2011.

[33] C. Padmini and J. Ravindra. CALPAN: Countermeasure against Leakage Power Analysis Attack by Normalized DDPL. In *ICCPCT 2016*, pages 1–7. IEEE, March 2016.

[34] A. Poschmann, A. Moradi, K. Khoo, C.-W. Lim, H. Wang, and S. Ling. Side-Channel Resistant Crypto for less than 2300 GE. *Journal of Cryptology*, 24:322–345, April 2011.

[35] S. M. D. Pozo, F.-X. Standaert, D. Kamel, and A. Moradi. Side-Channel Attacks from Static Power: When Should we Care? In *DATE 2015*, pages 145–150. IEEE, March 2015.

[36] M. Rivain. On the Exact Success Rate of Side Channel Analysis in the Gaussian Model. In *SAC 2008*, volume 5381 of *LNCS*, pages 165–183. Springer, 2008.

[37] T. Schneider and A. Moradi. Leakage assessment methodology - A clear roadmap for side-channel evaluations. In *CHES 2015*, volume 9293 of *LNCS*, pages 495–513. Springer, 2015.

[38] T. Schneider, A. Moradi, and T. Güneysu. Arithmetic Addition over Boolean Masking - Towards First- and Second-Order Resistance in Hardware. In *ACNS 2015*, volume 9092 of *LNCS*, pages 559–578. Springer, 2015.

[39] F.-X. Standaert, T. G. Malkin, and M. Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 443–461. Springer, 2009.

[40] F.-X. Standaert, E. Peeters, C. Archambeau, and J.-J. Quisquater. Towards Security Limits in Side-Channel Attacks. In *CHES 2006*, volume 4249 of *LNCS*, pages 30–45. Springer, October 2006.

[41] K. Tiri, D. Hwang, A. Hodjat, B.-C. Lai, S. Yang, P. Schaumont, and I. Verbauwhede. Prototype IC with WDDL and Differential Routing – DPA Resistance Assessment. In *CHES 2005*, volume 3659 of *LNCS*, pages 354–365. Springer, August 2005.

[42] J. Xu and H. M. Heys. Template Attacks Based on Static Power Analysis of Block Ciphers in 45-nm CMOS Environment. In *MWSCAS 2017*, pages 1256–1259. IEEE, August 2017.

[43] W. Yu and S. Köse. Security implications of simultaneous dynamic and leakage power analysis attacks on nanoscale cryptographic circuits. *Electronics Letters*, 52(6):466–468, March 2016.

[44] W. Yu and S. Köse. False Key-Controlled Aggressive Voltage Scaling: A Countermeasure Against LPA Attacks. *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(12):2149–2153, March 2017.

[45] W. Yu and S. Köse. Security-Adaptive Voltage Conversion as a Lightweight Countermeasure Against LPA Attacks. *Transactions on VLSI Systems*, 25(7):2183–2187, March 2017.

[46] N. Zhu, Y. Zhou, and H. Liu. Counteracting Leakage Power Analysis Attack Using Random Ring Oscillators. In *SNS & PCS 2013*, pages 74–77. IEEE, May 2013.

3.2 Static Power SCA of Sub-100 nm CMOS ASICs

Publication Data

> Thorben Moos. Static power SCA of sub-100 nm CMOS asics and the insecurity of masking schemes in low-noise environments. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):202–232, 2019

The acceptance rate for Volume 2019 of the IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES) was **19.6%** [Acca].

Content This work analyzes the effectiveness of static power analysis attacks to extract cryptographic keys from two different ASIC prototypes manufactured in 90 nm and 65 nm CMOS technology. It is shown that the 65 nm ASIC is substantially more susceptible to attacks, despite the fact that both chips have been developed and designed from the same code using the identical procedure. Furthermore, it is confirmed that operating conditions like the temperature and supply voltage can be abused by adversaries to increase the amplitude of the leaked data dependencies. With respect to the analysis of protected cryptographic implementations, it is revealed that the noise reduction inherent to static power attacks may have severe consequences for the security of masked implementations. When measurements with an extremely low noise influence can be recorded, masking schemes are not expected to provide the desired security level. Additionally, it becomes clear that commonly applied moment-based analysis techniques may be insufficient when evaluating the static power side-channel leakage of masked implementations.

Contribution The author of this thesis is the sole author of this publication.

Static Power SCA of Sub-100 nm CMOS ASICs

and the Insecurity of Masking Schemes in Low-Noise Environments

Thorben Moos

Ruhr University Bochum, Horst Görtz Institute for IT Security, Germany
thorben.moos@rub.de

Abstract. Semiconductor technology scaling faced tough engineering challenges while moving towards and beyond the deep sub-micron range. One of the most demanding issues, limiting the shrinkage process until the present day, is the difficulty to control the leakage currents in nanometer-scaled field-effect transistors. Previous articles have shown that this source of energy dissipation, at least in case of digital CMOS logic, can successfully be exploited as a side-channel to recover the secrets of cryptographic implementations. In this work, we present the first fair technology comparison with respect to static power side-channel measurements on real silicon and demonstrate that the effect of down-scaling on the potency of this security threat is huge. To this end, we designed two ASICs in sub-100 nm CMOS nodes (90 nm, 65 nm) and got them fabricated by one of the leading foundries. Our experiments, which we performed at different operating conditions, show consistently that the ASIC technology with the smaller minimum feature size (65 nm) indeed exhibits substantially more informative leakages (factor of ~10) than the 90 nm one, even though all targeted instances have been derived from identical RTL code. However, the contribution of this work extends well beyond a mere technology comparison. With respect to the real-world impact of static power attacks, we present the first realistic scenarios that allow to perform a static power side-channel analysis (including noise reduction) without requiring control over the clock signal of the target. Furthermore, as a follow-up to some proof-of-concept work indicating the vulnerability of masking schemes to static power attacks, we perform a detailed study on how the reduction of the noise level in static leakage measurements affects the security provided by masked implementations. As a result of this study, we do not only find out that the threat for masking schemes is indeed real, but also that common leakage assessment techniques, such as the Welch's t-test, together with essentially any moment-based analysis of the leakage traces, is simply not sufficient in low-noise contexts. In fact, we are able to show that either a conversion (resp. compression) of the leakage order or the recently proposed χ^2 test need to be considered in assessment *and* attack to avoid false negatives.

Keywords: Static Power · Leakage Current · Side-Channel Analysis · SPSCA · Masking

1 Introduction

The fundamental physical limits of computation dictate what can and what cannot be achieved by computing machines [BL85]. It has been shown many years ago, for example, that the majority of classical logic gates, being the essential building blocks of computing technology, cannot be evaluated without a certain amount of dissipation [Lan61, BL85]. This statement holds, regardless of the underlying device technology. In particular, state transitions performed by conventional logic operations are often of an irreversible nature, which means that information is discarded since two or more distinct logical states have a single successor [Ben03][1]. Such transitions *must* be accompanied by a loss of energy to the environment. This has been manifested in Landauer's principle [Lan61] and is a direct implication of the second law of thermodynamics [Llo00]. Whether information is discarded by a logic operation (i.e., an irreversible transition takes place) or not and therefore whether it is dissipated to the environment depends on the processed data [BL85]. Hence, as a matter of fact, computation, as it is currently carried out, does not only imply energy dissipation, but also leakage of information through physical side-channels[2] – entirely independent of any technological details.

This discussion, however, focuses on transitional leakages occurring during an active computation process exclusively. From a thermodynamic standpoint this is sufficient, since there is no necessity for dissipation without a transition of states. In other words, it should be possible to pause a physical computation process and to hold a stable state, keeping sensitive intermediates enclosed in the circuit, without being doomed to an undesired disclosure of information. This is in fact exactly what is described by the famous *only computation leaks* paradigm, introduced in [MR04]. The authors formulate the assumption that *"computation, and only computation, leaks information"*, implying that *"there is no information leakage in the absence of computation"*. Yet, as previous works regarding the information leakage of CMOS devices in stable states have shown, this assumption does no longer approximate the behavior of current semiconductor technologies to a sufficient degree.

Power Dissipation of CMOS Logic

Modern circuit technologies need to achieve many different objectives in parallel, with energy efficiency being only one of them. High performance, reliability, manufacturability and cost effectiveness are fundamental concerns, besides a number of further considerations depending on the desired area of application. Thus, not all effort can be dedicated to the reduction of the energy consumption and it can be observed that technologies suitable for very-large-scale integration (VLSI) in practice usually dissipate significantly more energy than what is demanded by the fundamental physical limits. Complementary metal-oxide-semiconductor (CMOS) logic gates, for example, consume a relatively large data-dependent current during the state transition from one output value to another, due to the associated charging and discharging of output capacitances[3]. Traditionally, this current is assumed to be the predominant cause for both, energy dissipation and information leakage, in this particular technology. However, over the years, physical characteristics and electrical specifications of transistors have changed significantly. To comply with Moores law [Moo65], the dimensions of metal-oxide-semiconductor field-effect transistors (MOSFETs) have faced an aggressive scaling process in order to achieve the desired and predicted exponential increase over time in the number of transistors that can be fabricated on a single integrated circuit (IC) of a given size. In the attempt to uphold this scaling factor, valuable properties of the technology were sacrificed, as for example the negligible current consumption in idle states.

Initially, CMOS logic has been constructed in such a way that, given the idealized model of a transistor holds, no current should be consumed in any stable state. In particular, the individual logic gates are composed of a pull-up network, which establishes a conductive path between the gate output and V_{DD} when activated, and a matching pull-down network,

---

[1] In a digital two-input `AND` gate, for example, the input combinations (0,0), (0,1), (1,0) are all mapped to output (0) and thus cannot be reversed.

[2] Logical reversibility can indeed be achieved by specialized and more complex logic gates, bearing the potential to eventually evade the lower bound of Landau [BL85, Llo00], however, a suitable device technology for nearly physical reversibility needs yet to be developed. In practice, any computing device will dissipate at least some energy [Llo00].

[3] CMOS gates also consume a (less data-dependent) short-circuit current during any output transition due to the short period of time where both, the pull-up and the pull-down network are conducting.

which is able to create a conductive path between the output and V$_{SS}$ (GND) respectively. For any combination of stable input signals, only one of the two networks is allowed to be active (i.e., switched on), while the other one, and therefore at least one transistor in any path between V$_{DD}$ and V$_{SS}$, should be switched off. Conceptually, this allows for a negligible power consumption in stable states, as for no static input combination a conductive path is formed across the power supply. Yet, by down-scaling the physical feature size, transistors progressively deviate from the idealized model. To be more precise, a nanoscale MOSFET does not resemble an ideal switch anymore but tolerates a significant off-current to flow between its terminals, even in a supposedly high resistance state. This behavior is a serious concern for hardware designers, as these so-called leakage currents consume a steadily increasing part of the power budget of modern ICs. It also leads to the situation that the global power consumption of circuits cannot be reduced to the amount of active computation anymore, measured by the number of gate toggles for example. Instead, even without any active computation (i.e., in an idle state) a significant amount of energy, proportional to the number of powered logic cells in the circuit, is consumed, independent of whether those cells are actively fed with input data or not. Thus, it is no surprise that leakage current reduction techniques such as power gating (MTCMOS), dual threshold CMOS (DTCMOS) or input vector control (IVC) gained increasing popularity among the VLSI community in the last decades [RMMM03].

Due to the structure of digital CMOS standard cells it can be observed that their individual cumulative off-current is highly determined by the composition and type of active and inactive transistors across the power supply path, which in turn directly depends on the applied input signals to the cell [AO13]. In other words, the static power consumption of CMOS logic is substantially data dependent. One common leakage reduction technique is therefore to assign primarily those input signal combinations to the individual logic cells when the device is in idle which cause the least amount of leakage current. The direct relation between the static power consumption of a cell and its inputs leads to the inconvenient and, from a side-channel perspective, highly alarming situation that on advanced CMOS hardware it is neither possible to actively process data, nor to passively keep (temporary) data in a circuit (e.g., in a flip-flop between consecutive clock cycles), without leaking information about those values via physical side-channels[4]. While the inability to compute without dissipating information-bearing energy amounts is a direct implication of the laws of thermodynamics (at least when considering standard logic gates due to the associated irreversible state transitions), leaking information in stable states (i.e., without any transition) is not necessary from a physical viewpoint and purely caused by technology-specific defaults which are further amplified through scaling effects. Thus, designers of security critical integrated circuits should be aware of the inherent information leakage of CMOS logic in active as well as in inactive states and the potential vulnerability of their devices to side-channel analysis attacks.

Side-Channel Analysis (SCA)

Side-channel analysis attacks exploit the data-dependent dissipation of computing devices in order to extract secret information from circuitry that executes cryptographic primitives. In fact, this threat is not limited to cryptography but applies to any manipulation of sensitive data on physical hardware. The repetitive processing of a fixed symmetric encryption key by a block cipher implementation is just one prime example of a potentially vulnerable target. Obviously, side-channel attacks which rely on measuring the physical emissions of an implementation, in contrast to, for example, its often remotely available execution time, are primarily a concern for devices that an adversary can obtain physical access to. Those devices are typically found in embedded systems. Among the possibilities to measure

---

[4] We consider only temporary memory elements such as flip-flops and latches here, whose output line, carrying the saved information, is connected to the input of further logic or memory cells.

and quantify the instantaneous data-dependent energy dissipation of an embedded device, power analysis [KJJ99] and electromagnetic emanation (EM) analysis [GMO01] have proven to be the most promising techniques with respect to their efficiency and simplicity, as opposed to, for example, thermal [HS14], acoustic [GST14] or optical [SNK+12] analysis of a target. Accordingly, it is no surprise that the lion's share of attention from academia and industry in the area of physical security of cryptographic hardware is devoted to these two sources of information leakage and their mitigation.

Static Power Side-Channel Analysis (SPSCA)[5]

The main body of research in the field of power analysis attacks focuses on the exploitation of dynamic effects which occur during the computation process, such as the switching of a digital gate output from low to high or vice versa. However, since the dynamic energy consumption (per logic unit) is declining, while the static power dissipation grows significantly in CMOS integrated circuits manufactured in advanced technologies [EB05], researchers have started to investigate the static power consumption as well. It has been shown in previous publications that this source of information leakage can successfully be exploited. [MMR18] provides a thorough description of the history of static power side-channel analysis (SPSCA) throughout the last decade, including a more or less complete list of publications in the area. Following a number of simulation-based investigations, Moradi demonstrated the first practical attempt to quantify the impact of this security threat based on real-world measurements at CHES 2014 for field programmable gate arrays (FPGAs) [Mor14]. Additionally, a first basic technology comparison is presented in [Mor14], as the examined FPGA families were manufactured in three different process technologies. Apart from this work, notable advances in the area include demonstrating that various established countermeasures against dynamic power side-channel analysis are essentially ineffective against the exploitation of the static currents [LB08, ABD+14, ABST14, IM14, Mor14, BST16, BBM+16, MMR17] and providing experimental evidence for the fact that influencing the working conditions of an operating integrated circuit can exponentially ease its exploitation [MMR18].
Of particular interest to the SCA community is certainly the concrete impact of the presence of static power side-channel leakage on the security offered by masking schemes. Masking is undoubtedly the most popular defense mechanism against (dynamic power/EM) side-channel analysis and to the best of our knowledge the only suitable option to achieve provable security claims under reasonable leakage assumptions. The term masking, a.k.a. secret sharing, refers hereby to a class of countermeasures that rely on splitting each sensitive variable of an algorithm into a discrete number of shares in such a way that only the combination of all of the shares contains information about the sensitive values [CJRR99, PR13]. In this way, a security level in terms of required number of leakage traces can be achieved which grows exponentially in the protection order (often closely related to the number of shares) while spending approximately a quadratic amount of resources [JS17, FGP+18]. Yet, such a relation can only be established when the leakage of the individual shares is sufficiently independent and the measurements that an adversary can acquire are sufficiently noisy [SVO+10, PR13, FGP+18]. Without a sufficient amount of noise, masked implementations are not expected to provide a security level that increases significantly in the protection order [CJRR99, SVO+10, PR13, Sta19], making the trade-off

---

[5] Various different notations have been introduced for *static power side-channel analysis* in the literature, e.g. *static power analysis* [XH17] and *leakage power analysis* [AGST09]. However, since the term *static power analysis* is already an established and unrelated expression in the EDA community and since *leakage* is a frequently used term with a mostly unrelated meaning in the side-channel literature, we stick to the (admittedly quite lengthy) notation of *static power side-channel analysis* in this work and use *static power SCA* and *SPSCA* as its abbreviations.

between spent resources and obtained security guarantee ineffective[6].
The first successful (higher-order) static power side-channel attack on a masked implementation has been performed in [Mor14]. It was also suggested in [Mor14] that masking schemes with a sequential manipulation of the shares (typical in software) might be in danger when an exploitation of the leakage currents is possible, since the shares may be leaked in a univariate fashion through the static power, making multivariate attacks unnecessary and potentially reducing the effective noise level. Further, and even more important to this work, [PSKM15] suggested that in case of an adversary obtaining full control over the clock signal (which was also assumed by [Mor14] and previous works) it is possible to average the static power consumption over an arbitrary time period, which allows to eliminate several sources of noise entirely. It was experimentally verified in [MMR17], and later more empirically in [MMR18], that this averaging technique in static power SCA attacks (with obtained clock control) indeed allows to reduce the noise level significantly. Furthermore, in [MMR17] a successful higher-order static power attack is performed which requires fewer traces to be successful than a corresponding dynamic power analysis attack on the same target. Considering that the static signal on their examined 150 nm chip should be orders of magnitude smaller than the dynamic one, this result clearly indicates that the noise in the static power traces could successfully be eliminated to a large extent. In fact, this result shows that without dedicated countermeasures, it is harder to assure a sufficient noise level against adversaries that measure static currents than against those who rely on measuring the dynamic switching activity of a chip. Such an observation goes hand in hand with the intuition that any *static* physical effect should, by definition, be easier to quantify with a high precision (i.e., low noise) than a corresponding transitional one, simply because static phenomena are persistent and not limited to a finite period of time.
To summarize, only a few practical works can be found in the literature which contribute to the discussion whether this side-channel can actually be of any harm to state-of-the-art cryptographic devices. While these articles deliver very valuable results, they also suffer from a number of shortcomings, making it difficult to fully oversee the concrete potential of this security threat, yet. We give two examples of such shortcomings in the following.
First of all, the technology comparison presented in [Mor14] leaves a lot of room for interpretation. In particular, the author discovers that no clear correlation between the feature size of the underlying CMOS technology of the FPGAs and the magnitude or the exploitability of their leakage currents can be observed. This contradicts not only what is suggested by the theory, but also what can be observed in the leakage characterization sheets of corresponding standard cell libraries [AO13]. In this case it is quite clear (from our point of view) that the inaccuracy of the results comes from the fact that, instead of ASICs, FPGA implementations were targeted. In fact, the three analyzed FPGA families differ in many structural and architectural regards from each other, apart from their underlying CMOS process node. Most of these technological differences and details are kept confidential as intellectual property (IP) by its vendors. Thus, it can never truly be determined which factors contribute to the observation that certain instances have a smaller or larger data-dependent leakage current on one FPGA device than on another. Finally, to the best of our knowledge, the three different FPGA devices were not even manufactured by the same foundry. Thus, a truly fair technology comparison examining the effect of down-scaling on the potency of this side-channel needs yet to be delivered.
The second work which requires a confirmation of its results on a different platform and under different conditions is [MMR17]. This article gives a first indication of the potential inherent susceptibility of masking schemes to static power attacks (which was predicted by [PSKM15]). But, in fact, only a single attack scenario is shown, without

---

[6] This becomes obvious and when taking a look at information theoretic plots and the lower bounds for the required number of observations to distinguish leakage distributions of boolean masked information [CJRR99, SVO^{+}10, PR13].

any statistical evidence for the reproducibility of the results, and no leakage assessment has been performed on the target[7]. From our point of view, it remains unclear whether the noise reduction through averaging actually led to a signal-to-noise ratio (SNR) where masking is essentially ineffective[8] or whether the noise level was simply reduced to a point where the SNR became greater than that in the compared dynamic power attack. Further, the analysis was performed on a rather outdated technology (150 nm) and without leakage-enhancing operating conditions which have proven to boost the SNR in such experiments [MMR18]. Thus, a more detailed analysis of the topic, preferably on a more advanced device technology and under different operating conditions is required to give a definite answer to the question whether and under which conditions masking and other side-channel countermeasures which require a certain noise level to be effective are inherently susceptible to SPSCA.

Our contribution

The contribution of this work is manifold. To begin with, we have developed two digital ASIC prototypes in sub-100 nm low power CMOS technology, one 90 nm and one 65 nm chip, and got them fabricated by one of the major foundries. All instances relevant to this work have been derived from identical RTL code and were implemented using an identical design procedure. Thus, we are able to provide a fair comparison between both technologies regarding the vulnerability of architectural and cryptographic instances to static power side-channel attacks. As a result of this comparison, we conclude that the data-dependent currents increase drastically when moving towards smaller CMOS technology nodes. In our case, the leakage exhibited by the 65 nm ASIC is roughly 10× as informative as the one on the 90 nm chip. Additionally, for the first time in literature, we perform static power SCA attacks on sub-100 nm CMOS ASICs under leakage-enhancing operating conditions, which allows us to validate the considerable impact that the applied temperature and core voltage can have on the exploitability of the static currents in CMOS devices. Interestingly, we find out that especially the influence of the temperature is much stronger in the more advanced process node. By raising the temperature from 20 °C to 90 °C and the core voltage from 1.2 V to 1.6 V the difference of means between two leakage distributions can be amplified by a factor of approximately 12 on the 65 nm chip.

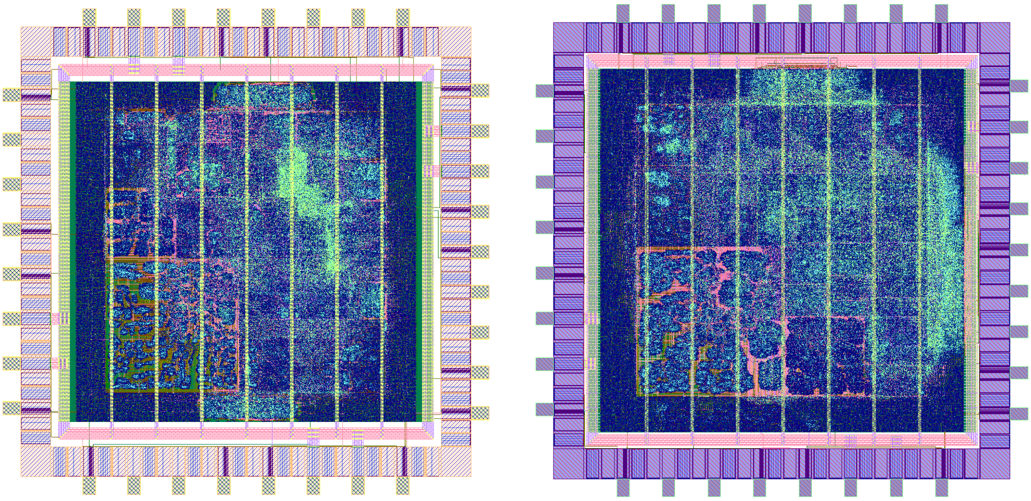
As a next step, we investigate the susceptibility of masked implementations to SPSCA and conclude that due to noise reduction techniques (i.e., averaging over time) adversaries can obtain measurements with such a low noise influence that masking is essentially ineffective. Furthermore, we argue that state-of-the-art leakage assessment techniques like the Welch's t-test are not suitable when analyzing masked implementations in very low noise environments as they cause false negatives. In fact, we come to the conclusion that moment-based analysis in general is not preferable in low-noise scenarios and that either a conversion, respectively compression, of the leakage order, or the recently presented χ^2 test need to be considered for assessment *and* attack. Finally, we show that for a variety of hardware implementations of cryptographic primitives clock control is no strict requirement to carry out a static power side-channel analysis. In particular, we demonstrate that whenever sensitive information remains in the circuit before or after a cryptographic operation is performed, it can be exploited. In this regard, we perform the first SPSCA attacks that do *not* require a stronger attacker model than conventional dynamic power analysis attacks[9]. Additionally, we show that in some cases it is even beneficial when certain

---

[7] To the best of our knowledge, none of the previously cited works has conducted a leakage evaluation by means of a statistical test, such as the Welch's t-test, either.

[8] [PSKM15] explains that noise averaging in static power SCA can be used to move from the *effective masking* zone to the *ineffective masking* zone.

[9] Although we perform these experiments at an increased temperature and supply voltage, control over these parameters is not conceptually necessary here and only used to reduce the required amount of traces.



(a) 65nm ASIC layout

(b) 90nm ASIC layout

Figure 1: Layout of the ASIC prototypes

parts of the circuit are actively computing during the measurement phase. In the end, we come to the conclusion that dedicated countermeasures against static power side-channel leakage are urgently needed and that masked implementations must be accompanied by a significant amount of algorithmic noise in order to not be susceptible.

2 Experiments

In this section, after shortly introducing the two developed ASIC prototypes and the measurement setup used for the experiments, we present a thorough vulnerability analysis of the devices under test with respect to their susceptibility to static power side-channel attacks. At first, we investigate the effects that manipulations of the operating conditions can have on their exploitability. Then, we analyze architectural and cryptographic instances on both chips to compare the magnitude of the information leakage exhibited by each of the two CMOS technologies. Finally, we use the most successful configuration, in terms of technology node and operating conditions, to carry out more sophisticated attacks.

Target

We have developed two ASIC prototypes in sub-100 nm CMOS technologies, whose layouts can be seen in Figure 1. Both chips are manufactured in low power CMOS technology, using low, high and standard threshold voltage cells. Both require a nominal core voltage of 1.2 V, an IO voltage of 2.5 V and use 9 metal layers for routing. They feature 33 IO pins in total, 17 for logic signals, 16 for power supply. Both chips have been packaged in JLCC-44 package and can be plugged on a custom measurement board which in turn is powered and controlled by a BASYS3 FPGA board. The chips contain a total of 27 different cipher cores, partially equipped with countermeasures against physical attacks, such as masking. All instances have been derived from the same RTL code in both chips and were implemented using the exact same design procedure. However, due to the different technology size some of the cores have a different utilization and a slightly different placement and routing.
Both ASICs contain 8 global 128-bit input registers, which serve the purpose of supplying the cryptographic cores with plaintext and key information, as well as 4 global 128-bit

output registers, which propagate the cores' output to the IO cells. For the cores that are protected by masking countermeasures this information can either be transmitted and received in a pre-shared form through the IO cells, or it is shared internally using fresh masks generated by a randomness source on the chip[10]. The largest block and key size among the cipher cores is 128 bit (AES-128). Accordingly, the size of the shift registers was chosen in order to be able to store a 128-bit key as well as a 128-bit plaintext and ciphertext, each split into 4 shares. In addition to the global input/output (IO) registers each core has its own local IO registers. The global registers are connected to all local registers. All of the cipher cores are clock-gated. Thus, an exemplary input procedure looks as follows. Through a 4-bit data bus a plaintext is given to the global plaintext register, which has been selected by a 4-bit address bus. The same is done for the key. Now, the clock of the targeted crypto core is activated and the plaintext and key are copied into its local registers. The global registers are cleared once the input is copied into the target core. Thus, during measurement the only difference in the state of the device lies in the targeted crypto core.

Setup & Procedure

Our measurement setup and procedure are similar to what has been proposed in [MMR18]. In particular, we use a custom DC amplifier, featuring a $\times 1,000$ amplification and a low-pass filter to get rid of the high-frequency noise in the measurements. Furthermore, we perform all experiments in a climate chamber to guarantee a constant temperature during the acquisition of the traces. The use of such a climate chamber as vital ingredient to any dedicated static power measurement setup was first proposed in [Mor14] and subsequently tested in [MMR17]. Each of our reported static power measurements is obtained by averaging 2 million time samples recorded over a period of 1 s by a LeCroy HRO 66zi sampling oscilloscope (i.e., sampling rate of 2 MS/s, measurement interval of 1 s). The chips were operated at 5 MHz whenever the clock signal was running.

Case Study 1: 1024-bit High-Fanout Register, 65 nm vs. 90 nm

As a first experiment in proof-of-concept manner we target an architectural instance which is expected to exhibit a large data-dependent leakage current, namely a high-fanout state register. In particular we chose the 8 global 128-bit input registers of the ASICs. For this initial experiment it is sufficient to view the 8 registers as one large 1024-bit register. The most important property of this instance for our upcoming analysis is that it is connected to all of the 27 cipher cores that are included in the ASICs. Thus, the output lines of the flip-flops of the 1024-bit register have a comparably large fanout, even though not all register bits are connected to all of the cores. In particular, the average fanout of these flip-flops is 11. Now, as soon as one bit of information is stored in one of them (by applying the value to its input and clocking once) it is directly propagated to the input of 11 further cells on average. An illustration of such a fanout of one single flip-flop to further memory elements can be seen in Figure 2. As detailed in [AO13], both, logic and memory cells leak information about the values that are applied to their input lines via the static power consumption. Thus, the information stored in one flip-flop is not only leaked by the cell itself (which indeed only has a relatively small contribution to the overall leakage), but also by the further 11 cells it is connected to. For this reason, we expect a clearly noticeable difference in the leakage currents when setting the whole 1024-bit state to either all 1s or all 0s.

We first verify this assumption on the 90 nm prototype by means of 5,000 static power measurements that are recorded after filling the registers' content with the randomly

---

[10] As shown in [SM15], sending only pre-shared input data to the target and receiving the output in shared form can be essential to avoid false positives in side-channel security evaluations.

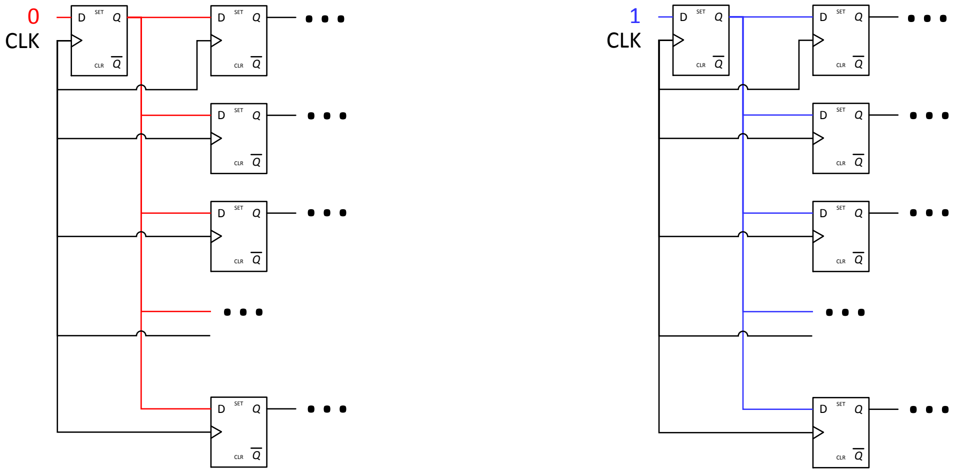
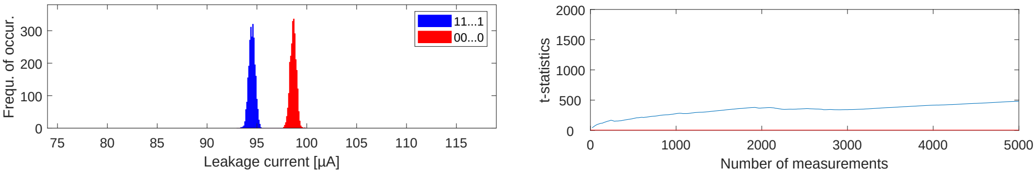


Figure 2: Exemplary depiction of a single bit saved in the high-fanout register, either set to 0 (left) or to 1 (right). The average fanout per first stage flip-flop is 11.

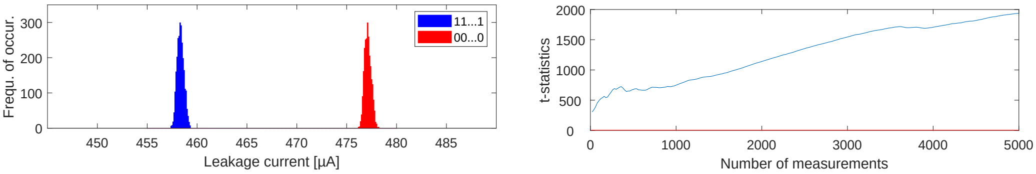
selected input (either all 1s or all 0s) and then stopping the clock signal of the chip and keeping all IO signals constant. The result of those measurements, presented as a histogram, can be seen in Figure 3(a). As expected, we obtain two non-overlapping leakage distributions which can easily be distinguished. The difference between the means of the two distributions is 4.1 µA and the average total current consumed by the ASIC in this idle state is 96.5 µA. All measurements presented in this section were performed using the previously mentioned DC amplifier, which applies a ×1,000 amplification to the static power consumption. However, please note that this amplification is already corrected (i.e., removed) in the reported values. In particular, whenever reporting a leakage current of, for example, 10 µA it means that the amplified static power signal was measured as a voltage drop of 10 mV over a 1 Ω resistor in the V_{DD} path of the ASIC. In order to quantify the distinguishability of the acquired leakage distributions we also performed a Welch's t-test, whose results can be seen in Figure 3(b). Clearly, the t-statistic does not only overcome the threshold of 4.5, which is normally set to decide whether side-channel leakage is detected or not, but it even reaches a value of about 480.

Before comparing these values to the 65 nm chip, we try to amplify the leakage by manipulating the operating conditions. [MMR18] showed that certain measurement factors are capable of boosting the signal-to-noise ratio in static power side-channel measurements significantly. Since we already applied a large measurement interval of 1 s per acquisition, we did not try to increase this parameter even further, but instead concentrated on the operating conditions temperature and supply voltage. In [MMR18] it is demonstrated that the exploitability of a 150 nm CMOS ASIC could exponentially be increased by raising the temperature. Further, increasing the voltage that is applied to the core area of the chip led to a marginal improvement of the attack success as well. The authors mention that the effect of both parameters is expected to increase significantly in more advanced CMOS technologies, such as our two sub-100 nm nodes. Thus, we repeated the initial experiment for another three times. First, we increased the supply voltage, then we raised the temperature, and finally, we manipulated both parameters. The results can be seen in Figure 3. Please note, that the scale of the x-axis in all four histograms is identical, only the range is different. It becomes apparent that both operating conditions have a significant impact on the data-dependent leakage currents. Interestingly, increasing



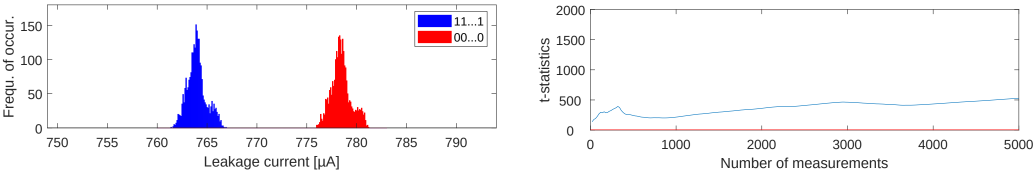
(a) histogram for 20 °C and 1.2 V

(b) *t*-test for 20 °C and 1.2 V



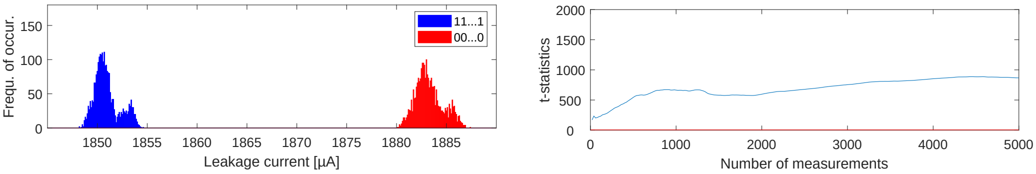
(c) histogram for 20 °C and 1.6 V

(d) *t*-test for 20 °C and 1.6 V



(e) histogram for 90 °C and 1.2 V

(f) *t*-test for 90 °C and 1.2 V



(g) histogram for 90 °C and 1.6 V

(h) *t*-test for 90 °C and 1.6 V

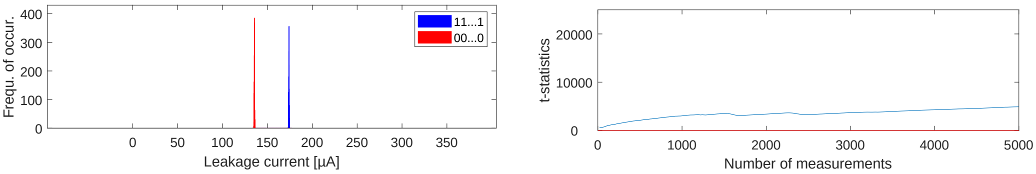
Figure 3: Histograms and *t*-test results for 5,000 static power measurements of a 1024-bit high-fanout register in 90 nm CMOS technology, filled either with only 1s or only 0s.

the supply voltage by 33.3% has a larger positive effect on the distinguishability of the distributions than raising the temperature by 70 °C (from 20 °C to 90 °C). This can be observed in both metrics, the difference of means between the distributions and the t-test results. However, the largest difference of means can be achieved by increasing both parameters. Yet, this does not directly lead to an improvement in the t-test results when compared to the scenario where only the voltage is changed. This is due to the additional noise at higher temperatures. As already mentioned in [MMR18], setting the controlled environment in the climate chamber to a temperature far above the room climate, leads to a constant activity of the regulation units, which can be observed as low frequency noise along the recorded set of traces. This type of noise causes the increased variance of the leakage distributions that can be seen in the histograms for the measurement sets that were recorded at 90 °C. However, as also explained in [MMR18], this type of noise can easily be removed by post-processing the traces using a high-pass filter. In this particular experiment we chose to not post-process the traces and rather report the raw, unaltered values as taken from the oscilloscope, in order to not distort the comparison. Yet, in all further case studies following in this section we made use of the moving average filter, as proposed in [MMR18]. Thus, in this experiment the difference of means is indeed the more important metric as it is not significantly influenced by the temperature noise. In total, by raising the temperature to 90 °C and increasing the supplied core voltage to 1.6 V, the difference between the mean values of the two distributions could be amplified by a factor of about 8 to a value of 32.3 µA.

After examining how informative the leakage currents of a 1024-bit high-fanout register in 90 nm technology are, we repeated the exact same kind of experiments on the 65 nm ASIC. The corresponding results are depicted in Figure 4. A couple of interesting differences can be noticed. First of all, while the 90 nm results showed a larger leakage current when the register is filled up with 0s, the opposite can be observed for the 65 nm technology. We refrain from speculating about potential reasons here and stress that this difference is due to internals of the particular standard cells. It is noteworthy that the exact same type of standard cells (i.e., with an identical name) were used for the whole register instance in both technologies, including all cells whose input is connected to the output lines of the register flip-flops. In other words this instance has the exact same netlist on both ASICs. Another difference between Figures 3 and 4 is clearly the magnitude of the currents. Please note that the scale on the x-axis of the histograms in Figure 4 is 10× as large as in Figure 3. This is also the reason why the distributions appear to be narrower, i.e., have a smaller variance, which is indeed not true. It's simply the distance between the distributions which is significantly larger.

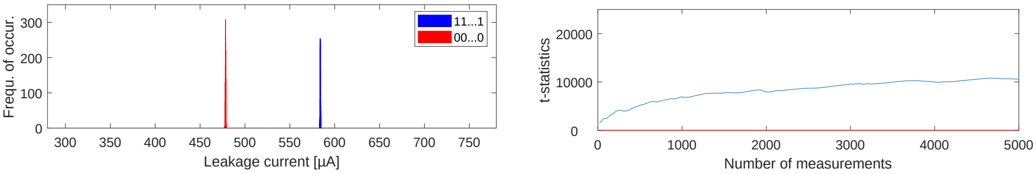
One may also notice that, in contrast to the 90 nm results, raising the temperature has a significantly larger impact on the distinguishability of the distributions than increasing the supply voltage in these experiments. This is not only reflected by the difference of means, but also in the t-test results, which is remarkable since the low frequency temperature noise is included in these measurements as well. Table 1 summarizes the data-dependency of both technologies for the different experiments to enable an easy comparison of the vulnerability of the two ASIC prototypes.

Finally, it can be said that in case an adversary is able to manipulate the operating conditions (temperature and supply voltage) of a device under test it is possible to amplify the static power side-channel leakage significantly (in our case by one order of magnitude), given that it is manufactured in an advanced CMOS process. Additionally, we have observed that the 65 nm chip exhibits substantially more informative leakages (also one order of magnitude) than the 90 nm one. Our 65 nm ASIC operating under a supply voltage that has been increased by 33.3% and in a 90 °C environment is more than 100× as susceptible to static power side-channel attacks as our 90 nm chip at nominal supply voltage and room temperature.



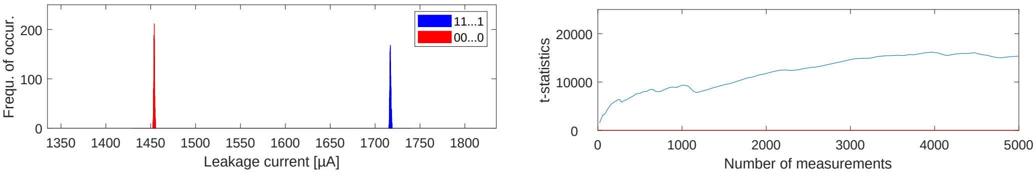
(a) histogram for 20 °C and 1.2 V

(b) *t*-test for 20 °C and 1.2 V



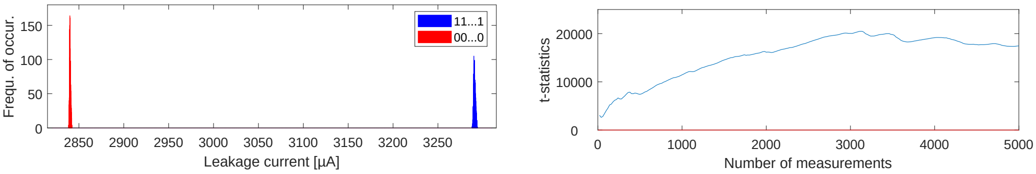
(c) histogram for 20 °C and 1.6 V

(d) *t*-test for 20 °C and 1.6 V



(e) histogram for 90 °C and 1.2 V

(f) *t*-test for 90 °C and 1.2 V



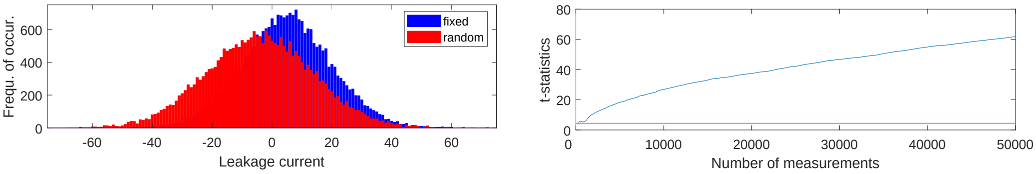
(g) histogram for 90 °C and 1.6 V

(h) *t*-test for 90 °C and 1.6 V

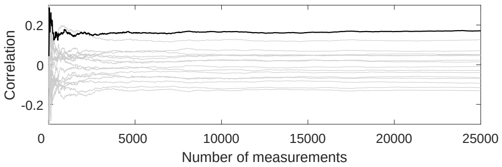
Figure 4: Histograms and *t*-test results for 5,000 static power measurements of a 1024-bit high-fanout register in 65 nm CMOS technology, filled either with only 1s or only 0s.

Table 1: Comparison of high-fanout register leakage in 65 nm vs. 90 nm technology for different operating conditions.

Techn.	Voltage	Temp.	Diff. of Means	Avg. Tot. Curr.	*t*-stat.
90 nm	1.2 V	20 °C	4.1353 µA	96.5 µA	480
90 nm	1.6 V	20 °C	18.7822 µA	467.3 µA	1,938
90 nm	1.2 V	90 °C	14.4754 µA	771.1 µA	526
90 nm	1.6 V	90 °C	32.3217 µA	1,867.3 µA	867
65 nm	1.2 V	20 °C	38.4927 µA	154.9 µA	4,890
65 nm	1.6 V	20 °C	105.5205 µA	529.9 µA	10,570
65 nm	1.2 V	90 °C	263.1579 µA	1,585.1 µA	15,360
65 nm	1.6 V	90 °C	450.6296 µA	3,067.2 µA	17,460



(a) histogram (b) first-order t-test



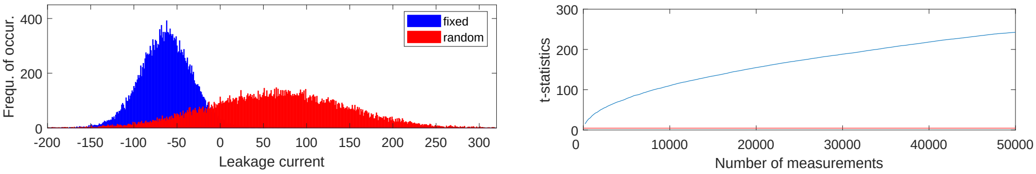
(c) CPA using the HW of one Sbox output after the first round targeting a key nibble

Figure 5: Leakage evaluation and attack using 50,000 fixed vs. random measurements of a nibble-serial implementation of the PRESENT-80 block cipher in 90 nm CMOS technology, recorded at 90 °C and 1.6 V.

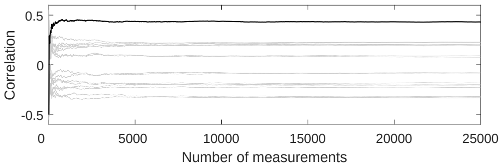
Case Study 2: Serial (Unprot.) PRESENT, 65 nm vs. 90 nm, 90 °C, 1.6 V

The second case study of our technology comparison targets an actual cryptographic primitive implemented on both ASICs. The measurements are performed at a temperature of 90 °C and with a supply voltage 1.6 V, since these operating conditions proved to enhance the information leakage through the static power consumption the most. In particular, we analyze the vulnerability of a nibble-serial implementation of the ultra-lightweight block cipher PRESENT-80 [BKL+07], without any side-channel countermeasures applied. The hardware implementation that we used is similar to the profile 1 of [PMK+11]. At first, we performed a leakage evaluation of the hardware primitive implemented in 90 nm technology using a non-specific (fixed vs. random) Welch's t-test, following the guidelines developed in [SM15]. In this regard the PRESENT core is supplied with randomly interleaved sequences of fixed and random plaintexts. Then the computation is executed until the end of the first round, where the clock signal of the ASIC is stopped and the leakage current drawn by the chip is measured. Please note that all global registers, analyzed in the previous case study, are cleared before measuring the static power in order to not obtain any false-positive t-test results, arising from the leakage of the saved plaintext. Thus, only the state which is currently present in the serialized PRESENT circuit differs between multiple measurements. The result of those acquisitions can be seen in Figure 5. As apparent from the histogram, the leakage distributions for the fixed and the random plaintext can easily be distinguished by visual inspection. Furthermore, the t-test overcomes the 4.5 threshold, indicating a detectable leakage, after less than 300 measurements. We also performed a correlation power analysis (CPA) [BCO04] on the traces that were measured for random plaintext inputs and target a key nibble of the first round key by using the Hamming weight (HW) of the Sbox output as a power model. Figure 5(c) shows that the attack succeeds in isolating the correct key candidate from the incorrect key guesses.

Afterwards we performed the same leakage evaluation and key recovery attack on the identical instance in the 65 nm technology. The corresponding results are depicted in Figure 6. Similar to the previous case study the polarity of the distributions is reversed, even though the same fixed plaintext as on the other chip was used. Additionally, it can be observed that the distributions are much easier to distinguish, not only in the difference of their means, but also in their variances. The corresponding t-test leads to a roughly 4× as large t-statistics value and the CPA succeeds with less traces and a larger absolute



(a) histogram (b) first-order t-test



(c) CPA using the HW of one Sbox output after the first round targeting a key nibble

Figure 6: Leakage evaluation and attack using 50,000 fixed vs. random measurements of a nibble-serial implementation of the PRESENT-80 block cipher in 65 nm CMOS technology, recorded at 90 °C and 1.6 V.

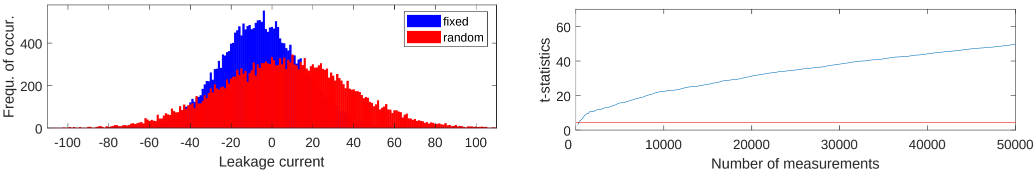
Table 2: Comparison of PRESENT block cipher implementation leakage in 65 nm vs. 90 nm technology for best case operating conditions (adversaries point of view).

Techn.	Voltage	Temp.	Diff. of Means	t-stat.	Correlation	MTD
90 nm	1.6 V	90 °C	9.15	61.96	0.17	2,180
65 nm	1.6 V	90 °C	128.46	242.5	0.43	100

correlation value as before. The concrete values are listed in Table 2 for an easy comparison. As already indicated by the previous case study, the 65 nm ASIC is significantly more vulnerable to static power side-channel attacks. The distance between the means of the fixed and the random distribution is about 14× higher and the attack requires less than $\frac{1}{20}$ of the number of traces, compared to the 90 nm chip.

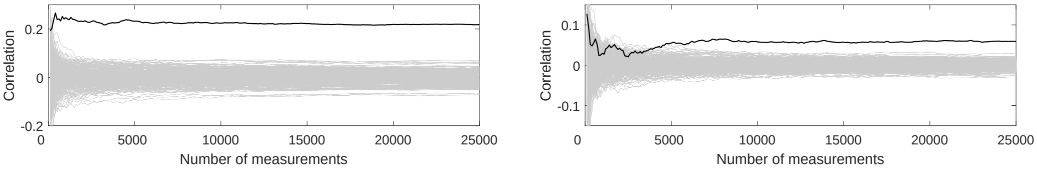
Case Study 3: Serial (Unprot.) AES, 65 nm, 90 °C, 1.6 V

As a next step we target a byte-serial implementation of AES. The examined circuit is the compact hardware implementation of AES, proposed in [MPL+11]. From this part on we concentrate on exploiting the 65 nm ASIC exclusively, since, based on the previously presented results it can be expected that it leads to more successful results due to a higher SNR. In this regard, we measured the static power consumption of the AES implementation when the encryption is paused after the end of the first round. Again, 50,000 traces for randomly interleaved fixed and random plaintexts are recorded. The corresponding results are presented in Figure 7. It can be seen that the AES hardware implementation is similarly susceptible to static power side-channel attacks as the PRESENT core. In this case we performed two CPA attacks on the traces that were recorded for random inputs. On one hand, we target the HW of the Sbox output which is currently evaluated by the Sbox module to reveal a byte of the first round key. And on the other hand, we correlate the HW of the Sbox output of a different byte, which is already saved in the state register and currently applied to the `MixColumns` operation of AES. Although both attacks do succeed with the available amount of traces, the CPA on the state byte which is currently processed by the Sbox requires much less traces and shows an overall higher correlation for the correct key candidate. This is obviously caused by the fact that this intermediate value is leaked by a larger combinatorial circuit, implementing the non-linear function. However,



(a) histogram

(b) first-order *t*-test



(c) CPA targeting an intermediate value that is applied to the combinatorial Sbox circuit

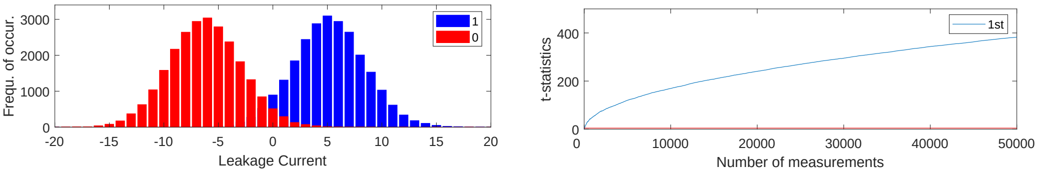
(d) CPA targeting an intermediate value that is applied to the MixColumns operation of AES

Figure 7: Leakage evaluation and attack using 50,000 fixed vs. random measurements of a byte-serial implementation of AES-128 in 65 nm CMOS technology, recorded at 90 °C and 1.6 V.

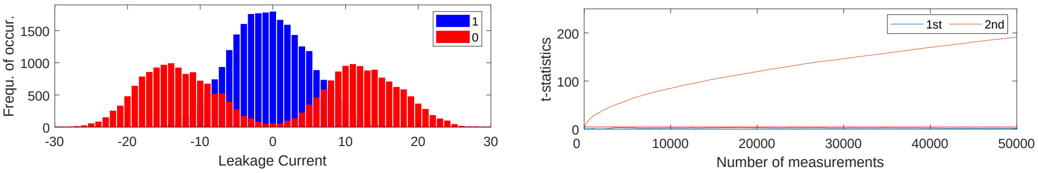
the fact that the leakage of the much smaller and linear `MixColumns` operation is sufficient to exploit it in a key recovery attack, shows that a static power analysis adversary is not forced to measure a new set of traces, stopping the clock in a different cycle, for every key byte when attacking serialized implementations. In theory, when the state register flip-flops are connected to sufficiently leaking memory or logic cells, a single set of traces is sufficient.

Case Study 4: Masked High-Fanout Register Bit, 65 nm, 90 °C, 1.6 V

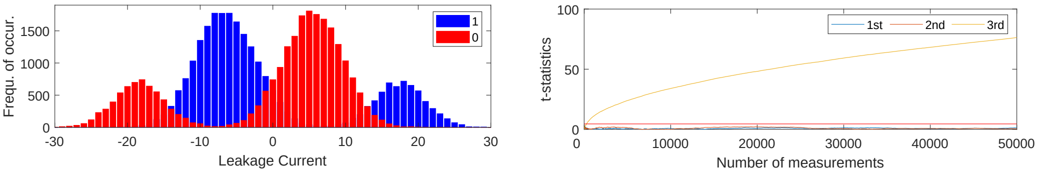
Masking, a.k.a. secret sharing, is without a doubt one of the most popular and theoretically sound countermeasures against side-channel attacks. In particular, when protecting a cryptographic implementation by means of a masking scheme, it is possible to achieve a security level, in terms of required number of side-channel observations for a successful attack, that grows exponentially in the masking order, while spending approximately a quadratic amount of resources [FGP⁺18]. However, masking can only deliver such a security guarantee in case the leakage of the individual shares is sufficiently independent and the traces that an adversary can acquire are sufficiently noisy. Due to the fact that temporary physical defaults such as transitions, glitches or couplings are not captured by the way the static power consumption is measured (and therefore cannot influence such measurements) it is comparably easy to achieve independence of the shares with respect to static power side-channel measurements. Yet, it is significantly more challenging to guarantee a sufficient noise level as most of the usual noise sources can be eliminated by averaging over time [MMR17, MMR18]. In this case study we take a look at the leakage of a single bit of information, split into multiple shares which are independently leaked by high-fanout flip-flops. Again, only the 65 nm ASIC is targeted and the operating conditions are set to 90 °C and 1.6 V, in order to obtain the best possible signal-to-noise ratio. We have measured 50,000 traces for randomly interleaved values of the secret bit and for each of the 5 different masking orders. Furthermore we performed (higher-order) *t*-test evaluations using the formulas introduced in [SM15] on the obtained leakage distributions. The corresponding results are illustrated in Figure 8. It can be seen, that, independent of the masking order, the distributions are clearly distinguishable by visual inspection. In particular, one can easily differentiate the Hamming weight classes of the shared secret. It is obvious that the SNR in these experiments is extremely high. The *t*-test results



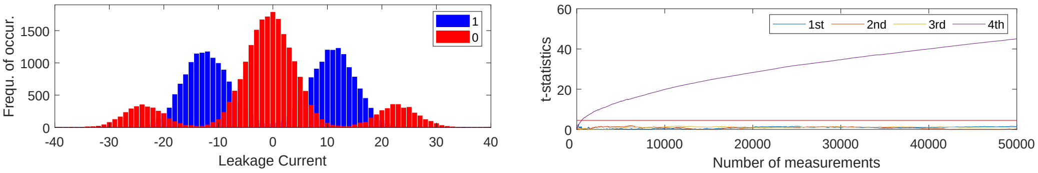
(a) histogram for 1 share (b) first-order t-test



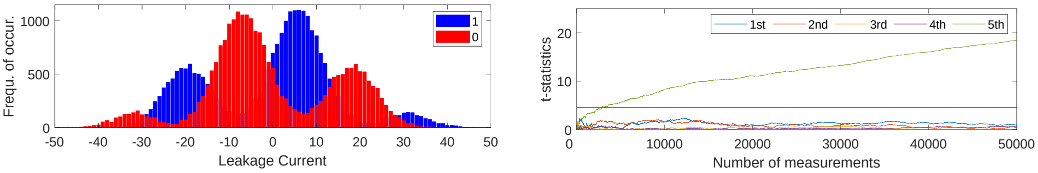
(c) histogram for 2 shares (d) first- to second-order t-test



(e) histogram for 3 shares (f) first- to third-order t-test



(g) histogram for 4 shares (h) first- to fourth-order t-test



(i) histogram for 5 shares (j) first- to fifth-order t-test

Figure 8: Histograms and (higher-order) t-test results for 50,000 static power measurements of 1-bit of information shared among 1, 2, ..., 5 (top to bottom) high-fanout register bits in 65 nm CMOS technology, recorded at 90 °C and 1.6 V.

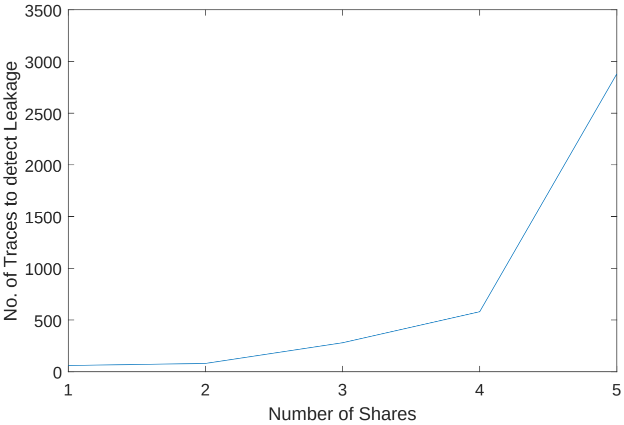


Figure 9: Number of traces to detect leakage for different masking orders.

show that leakage is only present in the expected statistical moments, corresponding to the number of shares. However, even though it does not seem to be significantly more difficult to distinguish the leakage distributions in the higher-order masked cases from their histograms, the t-test performs much worse in terms of the absolute magnitude of the t-statistics and number of required measurements to detect the leakage. This is also depicted in Figure 9. Such a result would suggest that the masking countermeasure is indeed in the *effective masking* zone [PSKM15], since the detection of the leakage becomes a lot more difficult when the masking order is increased. Yet, in the following we will detail that this is in fact a false negative result caused by the moment-based nature of the t-test analysis.

Leakage assessment approaches like the non-specific Welch's t-test have been introduced to simplify side-channel security evaluations of cryptographic implementations. Instead of the concrete exploitation of an implementation these methods are limited to the mere detection of side-channel leakage, independent of the recovery of a secret [SM15]. On the one hand this avoids the necessity to test a multitude of different attack scenarios and intermediate values to target. On the other hand such an approach naturally entails a high risk of false positives. In this context, by false positive we denote the reporting of detectable leakage which is not exploitable in an attack, e.g. leakage of the plaintext or ciphertext or some key-independent intermediates. This is inherent to leakage assessment approaches such as the non-specific Welch's t-test and constitutes the price that needs to be paid for not evaluating a multitude of attacks. However, what should at all cost be avoided are false negatives. By false negative we denote a scenario where a leakage test reports absence of detectable leakage considering a certain amount of traces (even when repeating multiple fix. vs. ran. or fix. vs. fix. tests for different fixed values), while there is indeed leakage present and exploitable with the available amount of traces. Such a scenario is the worst case for an evaluator as it undermines the whole purpose of leakage assessment tools.

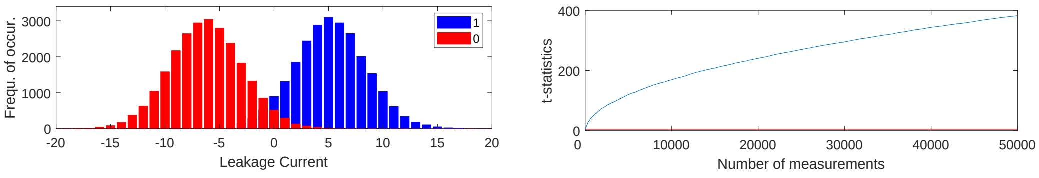
In [Sta19] Standaert describes a scenario where such a false negative result can occur in practice, namely when evaluating (higher-order) masked implementations with low noise levels. The fundamental problem of the moment-based test vector leakage assessment (TVLA) methodology in such cases is that an adversarial strategy is assumed. And estimating statistical moments is not the optimal strategy to attack masked implementations with low noise levels [Sta19]. In fact, the number of traces to detect leakage by a moment-based analysis can be much larger than the number of traces to exploit said leakage and recover a secret by a different strategy (than estimating statistical moments) [Sta19]. This is exactly what we observe in Figure 9, as the following comparison shows.

There exist (at least) two alternatives to the estimation of higher-order moments for leakage evaluation. The first one is the conversion, respectively compression, of the leakage order introduced in [MM17] and the second one is the recently proposed χ^2-test [MRSS18]. The former is based on applying a regular first-order t-test on slices of the leakage distributions and the latter compares full distributions to one another, without being limited to a single moment. We have applied both methods to our masked leakage distributions in Figures 10 and 11. One may notice that the success of both methods is much less affected by the masking order than the higher-order t-test. This becomes apparent in Figure 12 where the number of measurements required to overcome the leakage detection threshold is shown over the masking order. In this regard, we conclude that not only masking is indeed ineffective in very low noise scenarios, which can actually be achieved when performing real-world static power measurements, but also that moment-based leakage assessment techniques such as the Welch's t-test are not suitable in scenarios when the masking order is high and the noise level is low.

On the Need of Clock Control

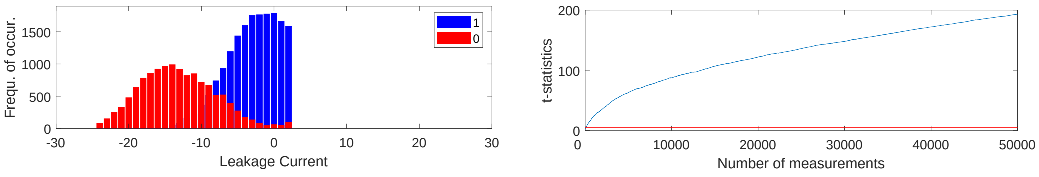
Traditionally, control over the clock signal of the device under test is an inevitable prerequisite for static power SCA attacks. Thus, performing such an analysis requires a stronger attacker model than classical power analysis adversaries do. [PSKM15] showed that without control over the clock, static power side-channel measurements are less informative than the dynamic power side-channel. Obviously, this is due to the fact that the sensitive intermediate values are present for only one or a few clock cycles in the circuit. Hence, their static power consumption cannot be measured over an extended period of time. However, the longer a certain value is present in the circuit and remains unchanged, the easier it becomes to exploit the leakage current of the respective gates carrying or receiving this information. Thus, theoretically, in case a cryptographic implementation does *not* ensure that *any* sensitive intermediate information is present for at most a few clock cycles in the circuit, this implementation can be susceptible to a static power analysis without the adversary having access to the clock signal. This assumption is explored in the following.

It is usually argued that measuring the static power consumption of, for example, a register content, even if it remains unchanged, cannot be done adequately if the device is actively performing computations somewhere else on the same chip, as the dynamic power consumption of that active computation would dominate the measured voltage drop, induce too much noise and limit the vertical resolution that can be set on the digital sampling oscilloscope. Thus, as a first step we evaluate whether the dynamic power consumption actually has a negative impact on the static power measurements. To this end we have repeated the exact same experiments from the previous case study, but instead of stopping the clock after filling the registers we disabled all the registers after filling them (using their EN pin) and enabled an LFSR-based PRNG on another part of the chip while measuring the total current drawn by the ASIC. The results of such experiments can be found in Appendix A (Figures 19, 20, 21 and 22). It turns out, that the measurements are not more, but in fact less noisy than the previous ones with the stopped clock signal. On the one hand this is due to the fact that the employed DC amplifier and low pass filter (see [MMR18]) have such a low bandwidth and cutoff frequency that no vertical amplitude caused by the dynamic power consumption can be observed. On the other hand this may be caused by the fact that the drop in the power consumption, shown in Figure 3 of [MMR18], is much smaller in this case. Accordingly, it is very well possible to measure the static currents associated with an intermediate value, even when other computations are performed at time of measurement. It is just required that the value remains long enough unchanged in order to measure it precisely. And in fact, many scenarios can be imagined where a sensitive intermediate value remains in a circuit for more than a couple of clock cycles.



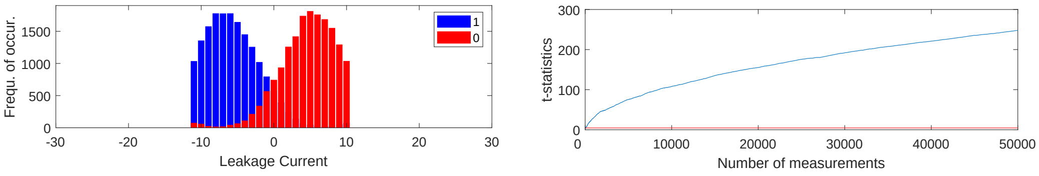
(a) selected slice of histogram for 1 share

(b) first-order *t*-test



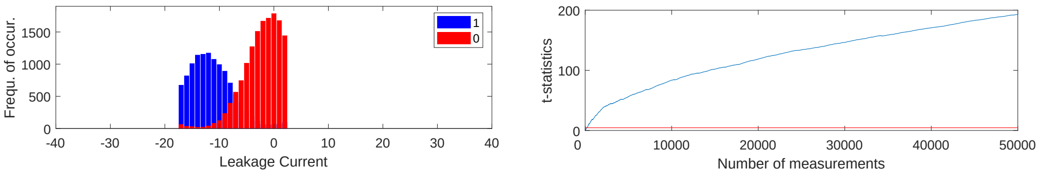
(c) selected slice of histogram for 2 shares

(d) first-order *t*-test



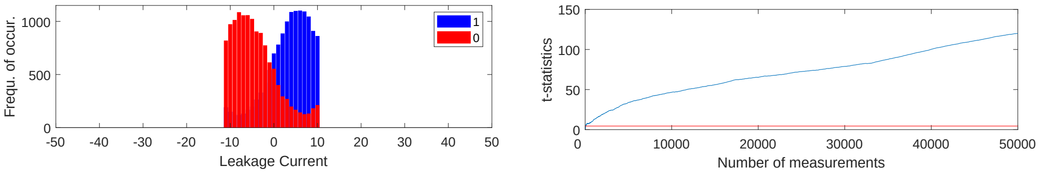
(e) selected slice of histogram for 3 shares

(f) first-order *t*-test



(g) selected slice of histogram for 4 shares

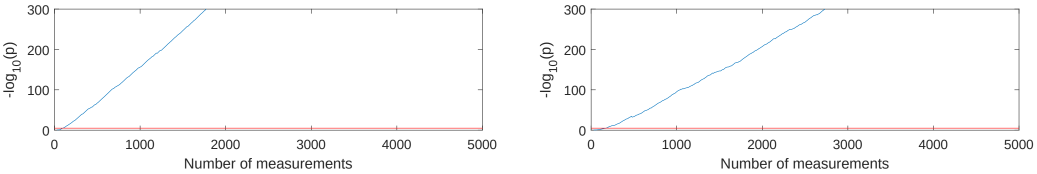
(h) first-order *t*-test



(i) selected slice of histogram for 5 shares

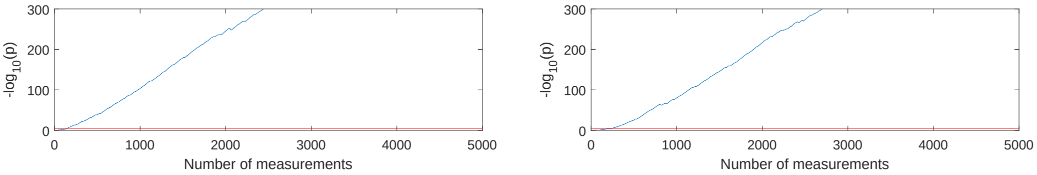
(j) first-order *t*-test

Figure 10: Histogram slices and *t*-test results for 50,000 static power measurements of 1-bit of information shared among 1, 2, ..., 5 (top to bottom) high-fanout register bits in 65 nm CMOS technology, recorded at 90 °C and 1.6 V.



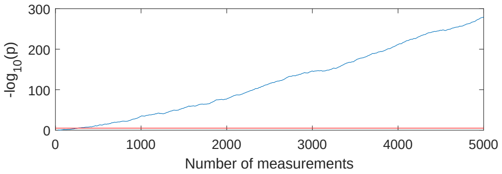
(a) χ^2-test for 1 share

(b) χ^2-test for 2 shares



(c) χ^2-test for 3 shares

(d) χ^2-test for 4 shares



(e) χ^2-test for 5 shares

Figure 11: χ^2-test results for the first 5,000 of the total 50,000 static power measurements of 1-bit of information shared among 1, 2, ..., 5 high-fanout register bits in 65 nm CMOS technology, recorded at 90 °C and 1.6 V.

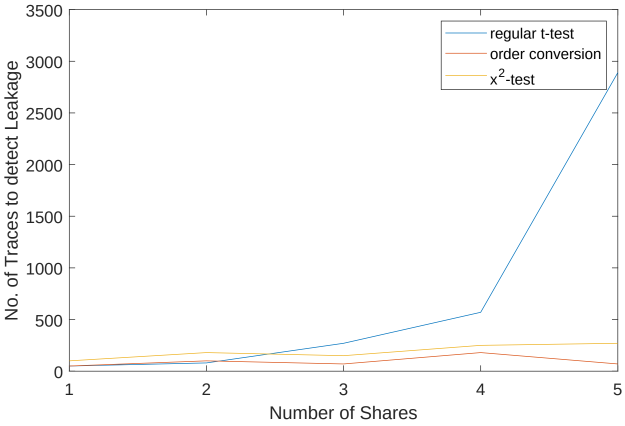


Figure 12: Number of traces to detect leakage for different number of shares using three different methods.

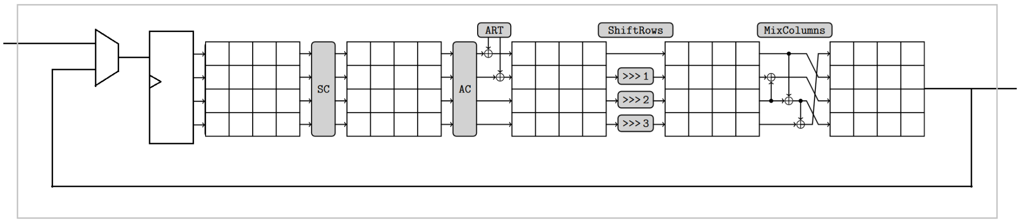


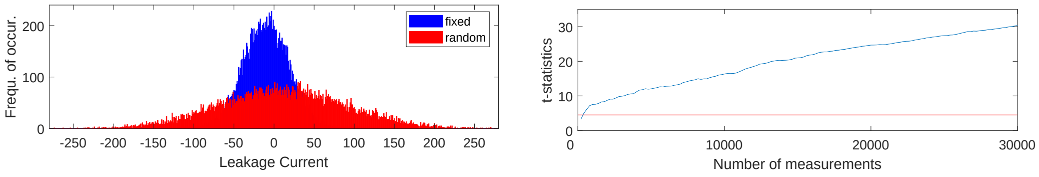
Figure 13: Exemplary depiction of a SKINNY hardware implementation (Partially copied from [BJK+16].)

Case Study 5: Round-Based (Unprot.) SKINNY, 65 nm, 90 °C, 1.6 V, PRNG running

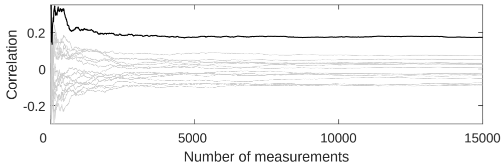
In this case study we present the first realistic scenario, where a static power SCA can be conducted without requiring control over the clock signal of the target. Following the previous discussion this is only realistic when sensitive intermediate values remain unchanged for an extended period of time in an implementation. A regular cryptographic cipher core will only be enabled and clocked when data needs to be encrypted. If not, the core will most likely be in a stable state (i.e., disabled via EN signal or clock-gated). After an encryption has been performed, either all input and intermediate registers are cleared immediately, or the current values remain in the circuit until the next encryption. Often the second option is chosen in order to save delay, power consumption and area (selecting a D-FF without RST signal). Unrolled and pipelined implementations are often not even supposed to be reset between encryptions. In other cases the cipher core is not reset immediately after each encryption, but rather right before the next plaintext needs to be processed, which also allows sensitive intermediates to remain in the circuit for an arbitrarily large period of time. Here we consider a round-based implementation of the SKINNY block cipher [BJK+16], as it is depicted in Figure 13. In particular, a multiplexer decides whether a new plaintext or a previous round output are saved into the state register. The remaining round function is supposed to be purely combinatorial. Typically, such an implementation would be clocked by a state machine until the ciphertext is stable at the output. When this is the case, it means that the second-to-last round output is present in the state register and stays applied to the combinatorial round function. As long as the state register is not immediately cleared, or a new plaintext is encrypted, we can actually exploit the values remaining in the circuit to recover the last round key by calculating back from the ciphertext all the way to the Sbox input of the last round. We have performed a leakage evaluation and the described attack on our SKINNY implementation on the 65 nm ASIC, under a temperature of 90 °C and a supply voltage of 1.6 V. The results are depicted in Figure 14. It can be seen, that the attack succeeds already after a few measurements in isolating the correct key candidate.

Case Study 6: Serial AES TI, 65 nm, 90 °C, 1.6 V

The final case study of this section targets a first-order AES threshold implementation. In this way, we aim to verify whether masked block cipher implementations are actually vulnerable with a comparably small number of measurements to static power side-channel attacks. The targeted circuit is the hardware implementation proposed in [MPL+11]. Figure 15 shows the results of a leakage assessment on this implementation by three different methods. All three techniques, namely higher-order t-test, order conversion and χ^2-test, succeed in detecting the leakage. As a next step we use the three distinguishers to perform a DPA attack on an Sbox output bit, targeting a key byte in the first round.

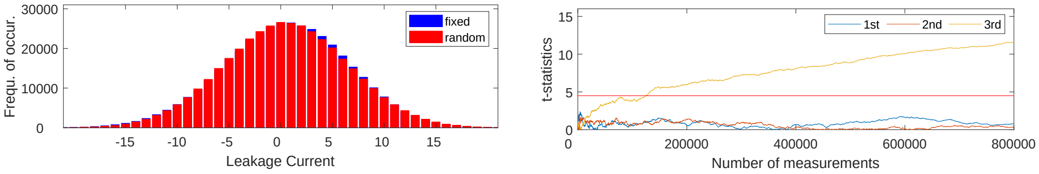


(a) histogram (b) first-order t-test

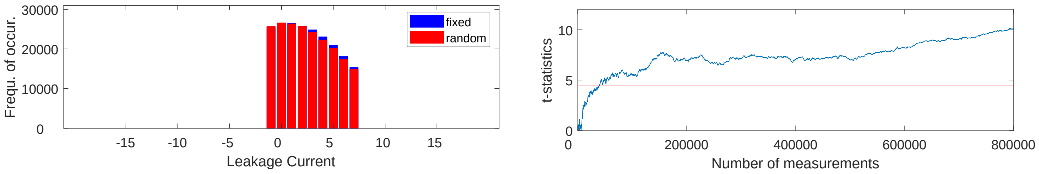


(c) CPA using the HW of one Sbox input in the last round targeting a key nibble

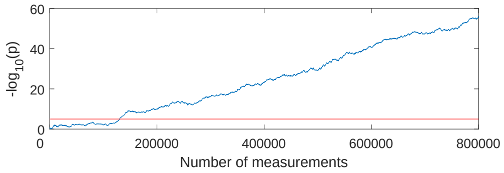
Figure 14: Leakage evaluation and attack using 30,000 fixed vs. random measurements of a nibble-serial implementation of the SKINNY block cipher in 65 nm CMOS technology, recorded at 90 °C and 1.6 V.



(a) histogram (b) first- to third-order t-test

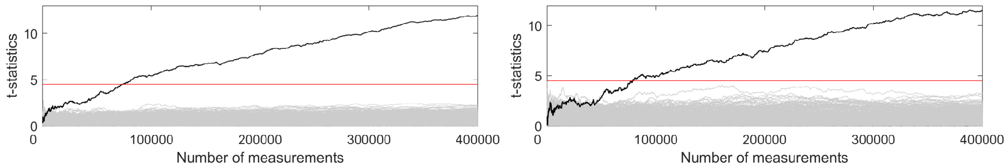


(c) selected slice of histogram (d) first-order t-test on slice

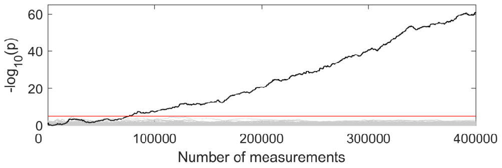


(e) χ^2-test

Figure 15: Leakage evaluation using 800,000 fixed vs. random measurements of a byte-serial AES threshold implementation in 65 nm CMOS technology, recorded at 90 °C and 1.6 V.



(a) third-order DPA using t-test (b) first-order DPA on sliced distrib. using t-test



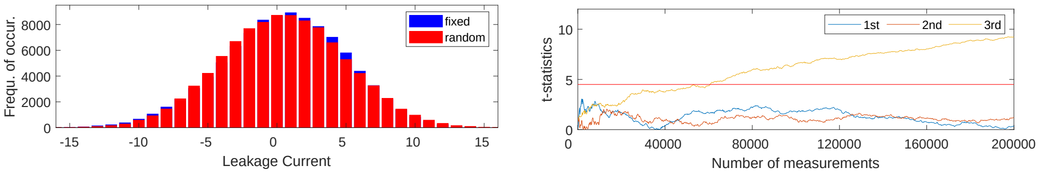
(c) DPA using χ^2-test

Figure 16: DPA attacks using different distinguishers on an Sbox output bit of an AES threshold implementation in 65 nm technology, recorded at 90 °C and 1.6 V.

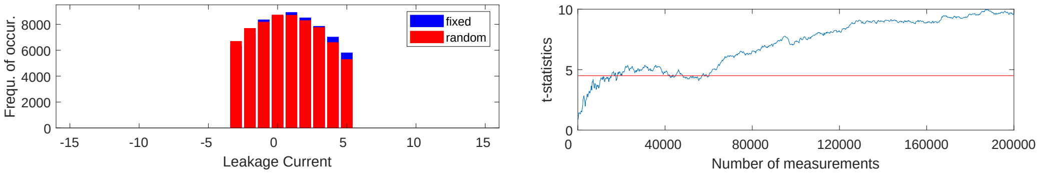
Again, all three methods succeed, as apparent in Figure 16. The required number of traces to overcome the threshold is similar among the three. However, we noticed that the LFSR-based PRNG, responsible for generating the fresh randomness which is required by the AES threshold implementation contributes significantly to the noise level, due to the fact that it holds a large state of random values during each of our measurements, which are leaked through the static power as well. Thus, in order to avoid this we decided, based on the results we achieved in the previous case studies for when the clock signal is not stopped, to keep the PRNG running during the measurements. Accordingly, its effect can be averaged out in each single measurement and it does not contribute to the algorithmic noise anymore. We repeated the previous evaluation and attacks again using this idea and achieved the results presented in Figures 17 and 18. Please note that this time we measured only 200,000 traces instead of 800,000. While both, the higher-order t-test and the order conversion require roughly 60,000 traces to detect the leakage and 20,000 to recover a key byte, the χ^2-test requires only 30,0000 for the detection and 12,000 for the recovery.

3 Conclusion

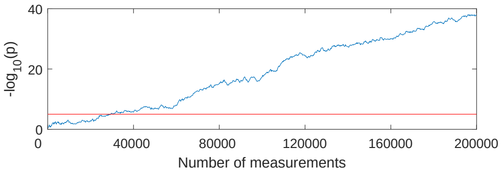
In this work, we have shown that the potency of the static power side-channel increases significantly when moving towards smaller feature sizes. Additionally, we could verify that manipulating the operating conditions of integrated circuits in advanced technologies can significantly boost the available information in corresponding static power measurements. This development, together with the possibility to reduce the effective noise level in such attacks poses a serious security risk for cryptographic hardware in advanced CMOS technologies. Countermeasures such as masking, which require a certain noise level to be effective are particularly affected by this development. Furthermore, these countermeasures cannot be properly evaluated by established evaluation methodologies, such as the moment-based TVLA methodology, since those are prone to produce false negatives in low noise environments when the masking order is high. Even devices that do not allow an adversary to obtain control over the clock signal need to pay attention whether sensitive intermediate values remain in the circuit for an extended period of time, e.g. in an idling cipher core. Finally, we conclude that dedicated countermeasures need to be developed to cope with this side-channel. To protect masking schemes from being susceptible, a suitable option is clearly the generation of additional algorithmic noise.



(a) histogram (b) first- to third-order *t*-test

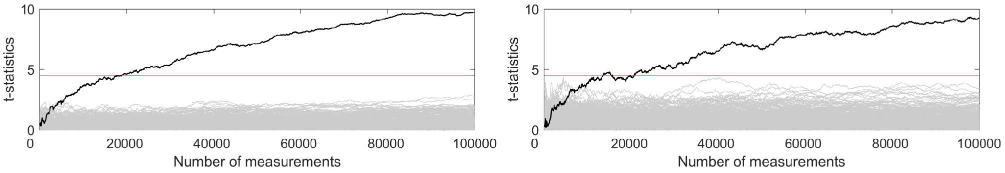


(c) selected slice of histogram (d) first-order *t*-test on slice

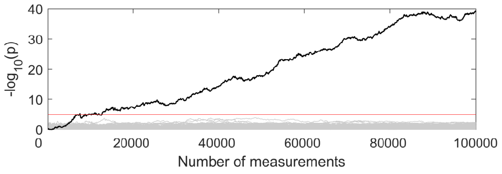


(e) χ^2-test

Figure 17: Same experiments as in Figure 15, but with only 200,000 traces and running the LFSR-based PRNG, responsible for delivering the fresh randomness, during the measurements to minimize algorithmic noise.



(a) third-order DPA using *t*-test (b) first-order DPA on sliced distrib. using *t*-test



(c) DPA using χ^2-test

Figure 18: Same attacks as in Figure 16, but with only 200,000 traces and running the LFSR-based PRNG, responsible for delivering the fresh randomness, during the measurements to minimize algorithmic noise.

Acknowledgments

The work described in this paper has been supported in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and through the project 271752544 "NaSCA: Nano-Scale Side-Channel Analysis".

References

[ABD+14] Massimo Alioto, Simone Bongiovanni, Milena Djukanovic, Giuseppe Scotti, and Alessandro Trifiletti. Effectiveness of Leakage Power Analysis Attacks on DPA-Resistant Logic Styles Under Process Variations. *Transactions on Circuits and Systems I: Regular Papers*, 61(2):429–442, February 2014.

[ABST14] Massimo Alioto, Simone Bongiovanni, Giuseppe Scotti, and Alessandro Trifiletti. Leakage Power Analysis Attacks Against a Bit Slice Implementation of the Serpent Block Cipher. In *MIXDES 2014*, pages 241–246. IEEE, June 2014.

[AGST09] M. Alioto, L. Giancane, G. Scotti, and A. Trifiletti. Leakage power analysis attacks: Well-defined procedure and first experimental results. In *2009 International Conference on Microelectronics - ICM*, pages 46–49, Dec 2009.

[AO13] Zia Abbas and Mauro Olivieri. Impact of technology scaling on leakage power in nano-scale bulk cmos digital standard cells. *Microelectronics Journal*, 45, 01 2013.

[BBM+16] Davide Bellizia, Simone Bongiovanni, Pietro Monsurrò, Giuseppe Scotti, and Alessandro Trifiletti. Univariate Power Analysis Attacks Exploiting Static Dissipation of Nanometer CMOS VLSI Circuits for Cryptographic Applications. *Transactions on Emerging Topics in Computing*, 5(3):329–339, May 2016.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

[Ben03] Charles H. Bennett. Notes on landauer's principle, reversible computation, and maxwell's demon. *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, 34(3):501–510, 2003.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In Pascal Paillier and Ingrid

Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

[BL85] Charles H. Bennett and Rolf Landauer. The fundamental physical limits of computation. 253(1):48–56, July 1985.

[BST16] Davide Bellizia, Giuseppe Scotti, and Alessandro Trifiletti. Implementation of the PRESENT-80 Block Cipher and Analysis of its Vulnerability to Side Channel Attacks Exploiting Static Power. In *MIXDES 2016*, pages 211–216. IEEE, June 2016.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Wiener [Wie99], pages 398–412.

[EB05] W. M. Elgharbawy and M. A. Bayoumi. Leakage sources and possible solutions in nanometer cmos technologies. *IEEE Circuits and Systems Magazine*, 5(4):6–17, Fourth 2005.

[FGP$^+$18] Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and François-Xavier Standaert. Composable masking schemes in the presence of physical defaults & the robust probing model. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):89–120, 2018.

[GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.

[GST14] Daniel Genkin, Adi Shamir, and Eran Tromer. RSA key extraction via low-bandwidth acoustic cryptanalysis. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 444–461. Springer, 2014.

[HS14] Michael Hutter and Jörn-Marc Schmidt. The temperature side channel and heating fault attacks. *IACR Cryptology ePrint Archive*, 2014:190, 2014.

[IM14] S. Shiney Immaculate and K. Manoharan. Analysis of Leakage Power Attacks on DPA Resistant Logic Styles: A Survey. *International Journal of Computer Science Trends and Technology*, 2(5):136–141, September 2014.

[JS17] Anthony Journault and François-Xavier Standaert. Very high order masking: Efficient implementation and security evaluation. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 623–643. Springer, 2017.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Wiener [Wie99], pages 388–397.

[Lan61] Rolf Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191, 1961.

[LB08] Lang Lin and Wayne Burleson. Leakage-Based Differential Power Analysis (LDPA) on Sub-90nm CMOS Cryptosystems. In *ISCAS 2008*, pages 252–255. IEEE, May 2008.

[Llo00] Seth Lloyd. Ultimate physical limits to computation. *Nature*, 406:1047–1054, 2000.

[MM17] Thorben Moos and Amir Moradi. On the easiness of turning higher-order leakages into first-order. In Sylvain Guilley, editor, *Constructive Side-Channel Analysis and Secure Design - 8th International Workshop, COSADE 2017, Paris, France, April 13-14, 2017, Revised Selected Papers*, volume 10348 of *Lecture Notes in Computer Science*, pages 153–170. Springer, 2017.

[MMR17] Thorben Moos, Amir Moradi, and Bastian Richter. Static power side-channel analysis of a threshold implementation prototype chip. In David Atienza and Giorgio Di Natale, editors, *Design, Automation & Test in Europe Conference & Exhibition, DATE 2017, Lausanne, Switzerland, March 27-31, 2017*, pages 1324–1329. IEEE, 2017.

[MMR18] Thorben Moos, Amir Moradi, and Bastian Richter. Static power side-channel analysis - A survey on measurement factors. *IACR Cryptology ePrint Archive*, 2018:676, 2018.

[Moo65] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), April 1965.

[Mor14] Amir Moradi. Side-channel leakage through static power - should we care about in practice? In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 562–579. Springer, 2014.

[MPL$^+$11] Amir Moradi, Axel Poschmann, San Ling, Christof Paar, and Huaxiong Wang. Pushing the limits: A very compact and a threshold implementation of AES. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 69–88. Springer, 2011.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 278–296. Springer, 2004.

[MRSS18] Amir Moradi, Bastian Richter, Tobias Schneider, and François-Xavier Standaert. Leakage detection with the x2-test. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):209–237, 2018.

[PMK$^+$11] Axel Poschmann, Amir Moradi, Khoongming Khoo, Chu-Wee Lim, Huaxiong Wang, and San Ling. Side-channel resistant crypto for less than 2, 300 GE. *J. Cryptology*, 24(2):322–345, 2011.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 142–159. Springer, 2013.

[PSKM15] Santos Merino Del Pozo, François-Xavier Standaert, Dina Kamel, and Amir Moradi. Side-channel attacks from static power: when should we care? In Wolfgang Nebel and David Atienza, editors, *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, DATE 2015, Grenoble, France, March 9-13, 2015*, pages 145–150. ACM, 2015.

[RMMM03] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer cmos circuits. *Proceedings of the IEEE*, 91(2):305–327, Feb 2003.

[SM15] Tobias Schneider and Amir Moradi. Leakage assessment methodology - A clear roadmap for side-channel evaluations. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.

[SNK$^+$12] Alexander Schlösser, Dmitry Nedospasov, Juliane Krämer, Susanna Orlic, and Jean-Pierre Seifert. Simple photonic emission analysis of AES - photonic side channel analysis for the rest of us. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*, pages 41–57. Springer, 2012.

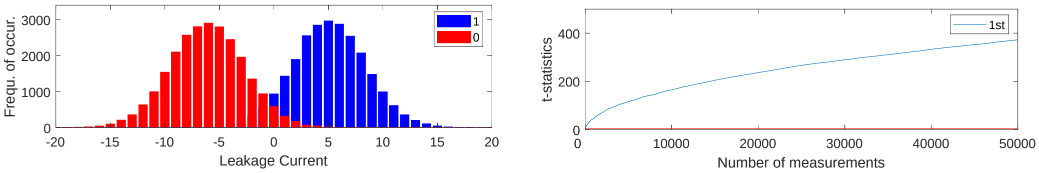
[Sta19] François-Xavier Standaert. How (not) to use welch's t-test in side-channel security evaluations. In Begül Bilgin and Jean-Bernard Fischer, editors, *Smart Card Research and Advanced Applications*, pages 65–79. Springer, 2019.

[SVO$^+$10] François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The world is not enough: Another look on second-order DPA. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.

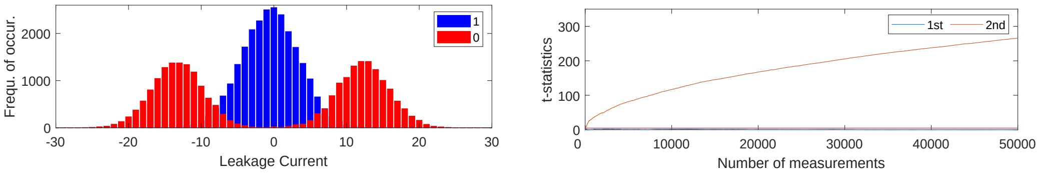
[Wie99] Michael J. Wiener, editor. *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*. Springer, 1999.

[XH17] J. Xu and H. M. Heys. Template attacks based on static power analysis of block ciphers in 45-nm cmos environment. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1256–1259, Aug 2017.

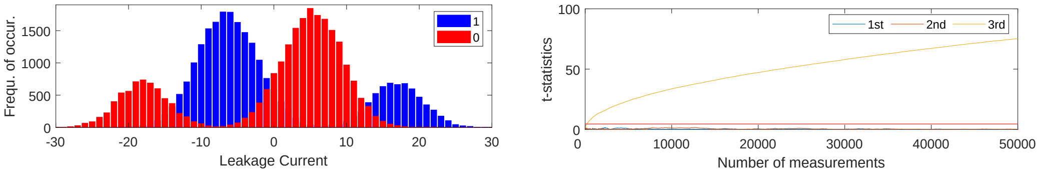
A Appendix 1



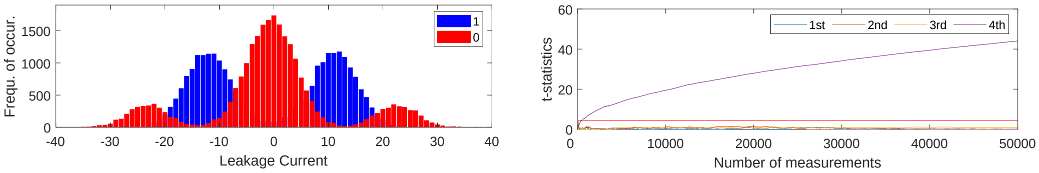
(a) histogram for 1 share (b) first-order *t*-test



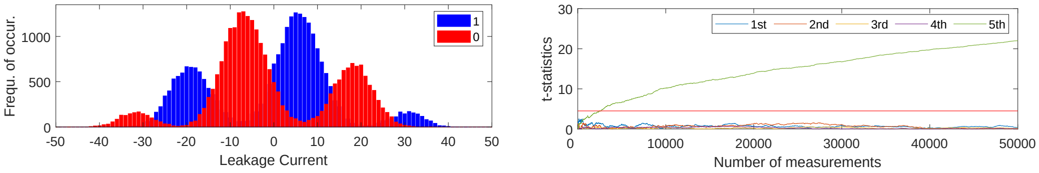
(c) histogram for 2 shares (d) first- to second-order *t*-test



(e) histogram for 3 shares (f) first- to third-order *t*-test

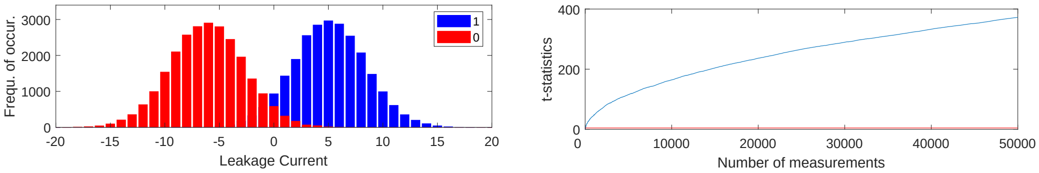


(g) histogram for 4 shares (h) first- to fourth-order *t*-test

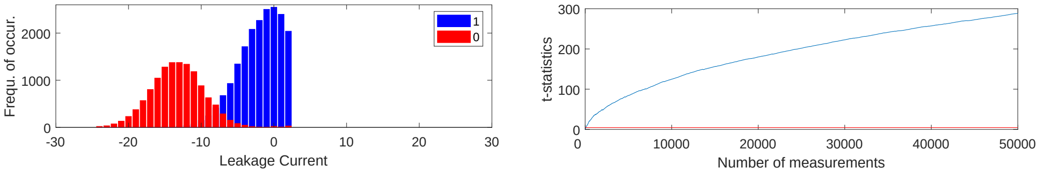


(i) histogram for 5 shares (j) first- to fifth-order *t*-test

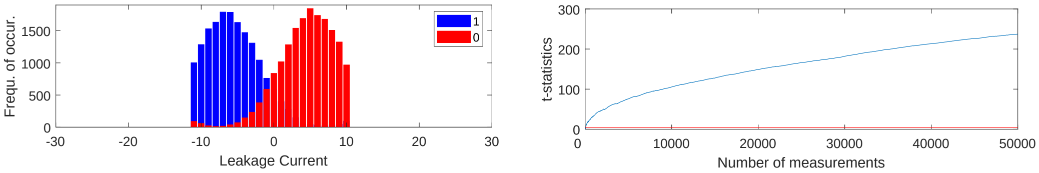
Figure 19: Same experiments as in Figure 8, but without stopping the clock and instead running an LFSR-based PRNG during the measurements.



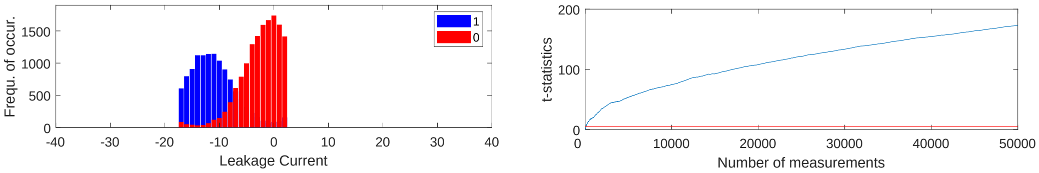
(a) selected slice of histogram for 1 share (b) first-order *t*-test



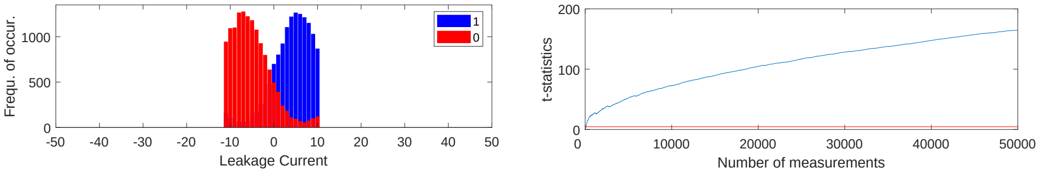
(c) selected slice of histogram for 2 shares (d) first-order *t*-test



(e) selected slice of histogram for 3 shares (f) first-order *t*-test

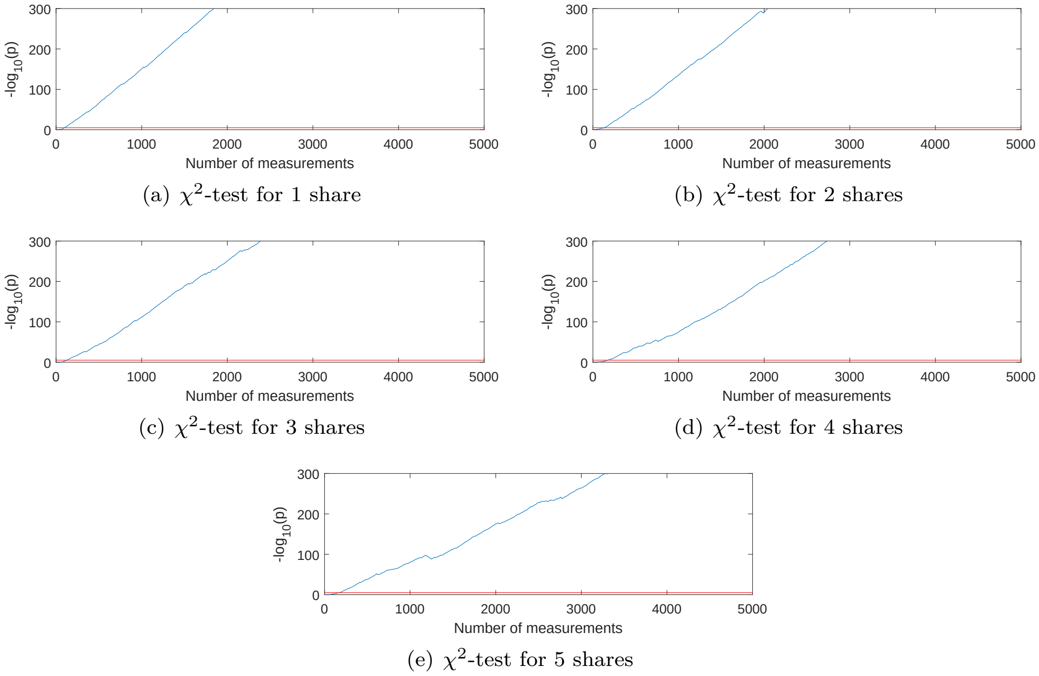


(g) selected slice of histogram for 4 shares (h) first-order *t*-test



(i) selected slice of histogram for 5 shares (j) first-order *t*-test

Figure 20: Same experiments as in Figure 10, but without stopping the clock and instead running an LFSR-based PRNG during the measurements.



(a) χ^2-test for 1 share

(b) χ^2-test for 2 shares

(c) χ^2-test for 3 shares

(d) χ^2-test for 4 shares

(e) χ^2-test for 5 shares

Figure 21: Same experiments as in Figure 11, but without stopping the clock and instead running an LFSR-based PRNG during the measurements.

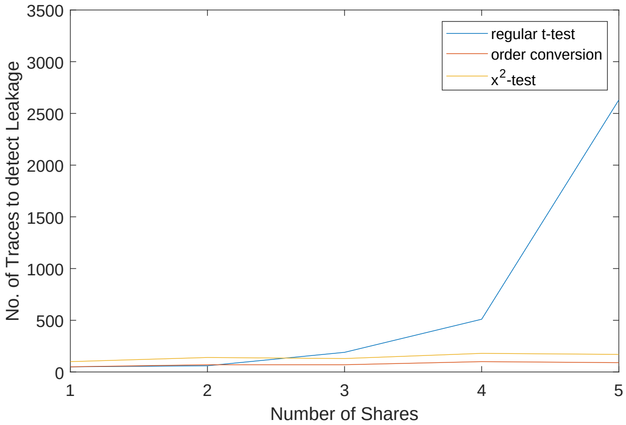


Figure 22: Number of traces to detect leakage for different number of shares using three different methods in Figures 19, 20 and 21.

3.3 Exploring the Effect of Device Aging on Static Power Analysis Attacks

Publication Data

> Naghmeh Karimi, Thorben Moos, and Amir Moradi. Exploring the effect of device aging on static power analysis attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):233–256, 2019

The acceptance rate for Volume 2019 of the IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES) was **19.6%** [Acca].

Content This work explores how the physical transformation of hardware circuits and their leakage behavior caused by device aging affects the vulnerability of cryptographic devices to static power side-channel attacks. The experimental results are based on the accelerated aging process of a 65 nm ASIC and a 150 nm ASIC in a climate chamber at a high temperature and significant over-voltage over a period of about 2 months. The data obtained from this practical investigation is compared to corresponding simulation results in order to conclude that device aging reduces the susceptibility of the hardware to static power attacks. It is also shown that input dependencies of combinatorial circuits may change significantly over time which additionally thwarts attacks.

Contribution The author of this thesis is a principal author of this publication. In particular, all practical experiments have been conducted and evaluated by the author of this thesis, who also contributed significantly to the writing and the presentation of the results. The author would like to thank both co-authors for their substantial contributions to the simulation results presented in this work.

Exploring the Effect of Device Aging on Static Power Analysis Attacks

Naghmeh Karimi[1], Thorben Moos[2] and Amir Moradi[2]

[1] University of Maryland, Baltimore County, USA
naghmeh.karimi@umbc.edu
[2] Ruhr University Bochum, Horst Görtz Institute for IT Security, Germany
firstname.lastname@rub.de

Abstract.
Vulnerability of cryptographic devices to side-channel analysis attacks, and in particular power analysis attacks has been extensively studied in the recent years. Among them, static power analysis attacks have become relevant with moving towards smaller technology nodes for which the static power is comparable to the dynamic power of a chip, or even dominant in future technology generations. The magnitude of the static power of a chip depends on the physical characteristics of transistors (e.g., the dimensions) as well as operating conditions (e.g., the temperature) and the electrical specifications such as the threshold voltage. In fact, the electrical specifications of transistors deviate from their originally intended ones during device lifetime due to aging mechanisms. Although device aging has been extensively investigated from reliability point of view, the impact of aging on the security of devices, and in particular on the vulnerability of devices to power analysis attacks are yet to be considered.

This paper fills the gap and investigates how device aging can affect the susceptibility of a chip exposed to static power analysis attacks. To this end, we conduct both, simulation and practical experiments on real silicon. The experimental results are extracted from a realization of the PRESENT cipher fabricated using a 65 nm commercial standard cell library. The results show that the amount of exploitable leakage through the static power consumption as a side channel is reduced when the device is aged. This can be considered as a positive development which can (even slightly) harden such static power analysis attacks. Additionally, this result is of great interest to static power side-channel adversaries since state-of-the-art leakage current measurements are conducted over long time periods under increased working temperatures and supply voltages to amplify the exploitable information, which certainly fuels aging-related device degradation.

Keywords: Leakage Current · Static Leakage Analysis · Side-Channel Analysis · Device Aging

1 Introduction

Two decades after its introduction to the public domain [KJJ99] and in view of numerous contributions of the scientific community towards a better understanding of its sources and mitigation mechanisms, side-channel analysis (SCA) attacks are known as a serious threat to devices which deal with cryptographic primitives. It can be considered as general knowledge that a cryptographic device, where a secret is stored and processed, is vulnerable to SCA attacks if it is not equipped with dedicated countermeasures. Amongst such attacks, power analysis attacks have attracted more attention due to their simplicity and high efficiency to recover the secrets.

Until recently, the dynamic power consumption has dominated the power behavior of CMOS integrated circuits and was, thus, the main source for their SCA leakage. Not surprisingly, almost the entire body of SCA research from the last 20 years, which consists of several 100s of publications, is based on the dynamic power consumption of the underlying circuit. However, through the continuous shrinkage of semiconductor technology, the static power consumption of CMOS circuits is becoming a major concern, while the dynamic power consumption (per logic unit) is decreasing due to smaller capacitances, shorter rising and falling times, and smaller supply voltages. Hence, it can be observed that in newer generations of CMOS devices the static power behavior contributes an increasingly large share to the overall power consumption [KAB$^+$03, Hel09, WERR13, Sha12]. As a consequence, SCA attacks based on dynamic power will become increasingly more difficult, which can be viewed as a positive development. Therefore, in a couple of works it has been attempted to estimate the feasibility of SCA attacks through the static power consumption by means of transistor-level simulations [GSST07, LB08, AGST09, AGST10]. It is noteworthy, that such SCA attacks are sometimes referred to as *leakage current* power analysis attacks. Afterwards, such experiments have been conducted in practice demonstrating successful key recoveries by means of the static power side channel [Mor14, PSKM15] under certain conditions. Further, the effectiveness of a masking countermeasure [NRS11] to mitigate dynamic power analysis attacks is practically compared to those based on static power [MMR17].

In contrast to traditional power analysis attacks, measuring the static power consumption requires a sophisticated setup and faces several engineering challenges. For example, the adversary should have control over the clock signal to force the target device into an idle state (i.e., no activity or change in the state of the circuit) to be able to measure the DC shift of its current consumption, i.e., its leakage current. This process also needs to be performed in a temperature-controlled environment, i.e., a heating chamber, since the leakage current is extremely sensitive to temperature variations. In addition, due to its very low amplitude, the leakage current signal needs to be amplified with a high gain and low-pass filtered to achieve measurements suitable for SCA attacks. Recently, a study on the effect of various elements of a measurement setup on the success of such SCA attacks has been published in [MMR18].

From another perspective, with the rapid scaling of process technology, aging-related degradation of integrated circuits has become one of the main challenges in nano technologies. Due to aging, electrical behavior of transistors deviates from the originally intended one, leading to performance degradation in the underlying device, and ultimate device failure [SKR$^+$13, KHHC11]. Generally speaking, device aging leads to an increase of the threshold voltage of the transistors over time. Therefore, the gates exhibit longer propagation delays compared to their original state. Further, recent works show that the static power of a CMOS circuit reduces by aging. This has been examined by transistor-level simulation [RTY$^+$17]. Although aging mechanisms and related mitigation schemes have received the lion's share of attention from reliability perspective in recent years [LK11, EKD$^+$03], their impact on the security of devices, in particular cryptographic devices, is yet to be investigated.

Our Contribution. In this work we examine the effect of aging on the exploitability of information leakage through the static power consumption of cryptographic devices as a side channel. We conduct both, simulations and a practical analysis, to study such an effect. Our practical investigations are based on analyses conducted on two samples of an ASIC chip which we have designed and fabricated in a 65 nm technology node, as well as on an available prototype in 150 nm technology. For transistor-level simulations of the 65 nm ASICs we model the post-layout netlist of the target core by means of SPICE models provided by the foundry that manufactured the ASIC prototype.

It is noteworthy, that in order to observe the effect of aging, we followed an aging-acceleration process by operating the device at a high temperature, a high supply voltage, and a dynamic workload [JED16]. Our analyses demonstrate that, although the concrete nature of the data dependency may shift due to aging, the amount of information leaked through the static power consumption is reduced when the device is aged. This highlights a common issue in static power analysis attacks. Since the amount of leakage current is increased in higher temperatures and for higher supply voltages, static power analysis attacks are usually conducted while the device is operated at a high temperature, e.g., 90 °C, and an increased supply voltage, which leads to more easily exploitable leakages, i.e., lower number of traces for successful attacks [BCS+17, DBST17, MMR18]. However, during such a special measurement condition the device is aged faster, and as result of our research the exploitability of its leakage current is steadily decreased. This causes a mismatch between the samples collected at different measurement phases. Of course, this is only of concern if the measurement process takes a long time, e.g., at least a couple of weeks.

2 Preliminaries

In this section, we first present a background on device aging and its effect on circuit's characteristics, followed by a discussion on how aging affects the leakage currents in modern CMOS technology. Afterwards, we express the design architecture of the case study which we consider in our investigations.

2.1 Device Aging

Device aging results in performance degradation and eventual failure of digital circuits over time [Kim15]. Aging mechanisms include Negative Bias Temperature-Instability (NBTI), Positive Bias Temperature-Instability (PBTI), Hot Carrier Injection (HCI), Time Dependent Dielectric Breakdown (TDDB), and Electro-Migration (EM).

In practice, BTI (including NBTI and PBTI effects) is one of the major causes of threshold voltage increase in transistors during their lifetime. NBTI and PBTI occur in PMOS and NMOS transistors, respectively. In practice, the impact of NBTI is more dominant than PBTI beyond the 45 nm technology node. However, with the introduction of high-k gate dielectrics and metal gate transistors, PBTI effects have also received significant attention [ZKN+06, CPC+05]. NBTI occurs in a PMOS transistor when a negative voltage is applied to its gate. In this mechanism, positive interface traps are generated at the Si-SiO_2 interface. As a result, the threshold voltage increases and the PMOS transistor becomes slower and fails to meet timing constraints. In contrast to NBTI, PBTI occurs when a positive voltage is applied to the gate of an NMOS transistor. This results in generating traps at the interface of gate oxide and channel, and in turn increasing the transistor threshold voltage.

HCI occurs when hot carriers are injected into the gate dielectric during transistor switching and remain there. HCI mainly affects NMOS transistors, degrades the underlying circuit by shifting the threshold voltage and the drain current of transistors under stress [RFFT14a]. TDDB relates to the creation of an electrical current conduction path through the gate oxide in the device-under-stress. It degrades the isolation properties of gate dielectric, increasing the tunneling current across the transistor gate terminal, and ultimately results in device breakdown [NBRR13]. On the other hand, high density currents result in EM aging. The currents create electron winds that cause metal atoms to migrate over time, gradually removing metal atoms from wires, thereby increasing interconnect resistance, and eventually resulting in an open circuit [Miz08].

Among all aging mechanisms, BTI and HCI are two leading factors in degradation of digital circuits [KDLG16]. Both mechanisms result in increasing switching and path delays in the circuit under stress [KGD18, KDG18]. What follows discusses these aging mechanisms in more detail.

2.1.1 NBTI Aging

NBTI affects a PMOS transistor when a negative voltage (i.e. $V_{gs} < V_t$) is applied to its gate. In fact, a PMOS transistor experiences two phases of NBTI depending on its operating condition. The first phase, so-called stress phase, occurs when the transistor is on, i.e., when a negative voltage ($V_{gs} < V_t$) is applied to its gate. In this case, positive interface traps are generated at the Si-SiO$_2$ interface which lead to an increase of the threshold voltage of the transistor. The second phase, so-called recovery phase, occurs when a positive voltage ($V_{gs} > V_t$) is applied to the PMOS transistor's gate. As a result, the threshold voltage drift that occurred during the stress phase will partially recover.

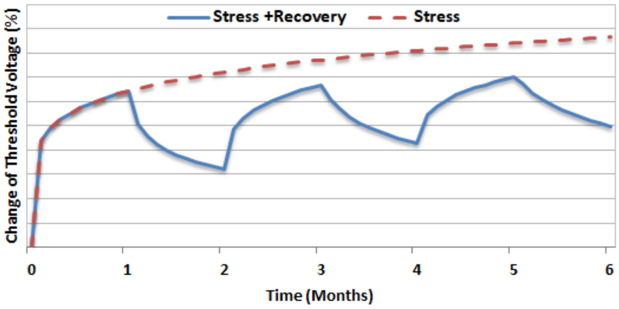


Figure 1: Threshold voltage shift of a PMOS transistor under NBTI effect.[1]

Threshold voltage drifts depend on the physical parameters of the transistor, supply voltage, temperature, and stress time [AKVM07, KCC+05]. The last three parameters (so-called external parameters) are generally used as acceleration factors of aging process. Figure 1 shows the threshold voltage drift of a PMOS transistor that is continuously under stress for 6 months and a PMOS transistor that alternates stress/recovery phases every other month. As shown, NBTI effect is high in the first couple of months but the threshold voltage tends to saturate for long stress times. The impact is exacerbated with thinner gate oxide and higher operating temperature [AKVM07, MSVK06].

Two prevalent theories, Reaction-Diffusion (R-D) and Trapping-Detrapping (T-D), have been proposed in literature to explain NBTI. The R-D model explains the NBTI phenomenon as the breaking and rebonding of hydrogen-silicon bonds at the silicon-gate dielectric interface of PMOS devices [Sch07, CCLM14]. The T-D model considers a number of defect states with different energy levels, and capture and emission time constants. In the T-D model, the threshold voltage increases when a trap captures a charge carrier from the channel of a PMOS device [SVRC15].

According to the R-D model proposed in [WYB+10], the NBTI-related increase in the threshold voltage of a PMOS transistor in the stress phase is evaluated as follows [TT08].

$$\Delta V_{th_{\mathrm{st}}} = A_{\mathrm{NBTI}} \cdot t_{ox} \cdot \sqrt{C_{ox}(V_{dd} - V_{th})} \cdot e^{\left(\frac{V_{dd}-V_{th}}{t_{ox} \times E_0} - \frac{E_a}{k \times T}\right)} \cdot {t_{\mathrm{st}}}^{0.25}, \qquad (1)$$

where t_{ox} is the oxide thickness, and C_{ox} the gate capacitance per unit area. The constants E_0 and E_a stand for device-dependent parameters, A_{NBTI} is a technology-dependent

---

[1] The Y axis has not been shown to make the graph generic for different technologies.

constant, and k the Boltzmann constant. T represents the temperature, and t_{st} the stress time.

As discussed, the threshold voltage drift of a PMOS transistor is partially recovered if the transistor is placed in the recovery phase. The following equation expresses the final change in the threshold voltage of a PMOS transistor [TT08].

$$\Delta V_{th_{\text{NBTI}}} = \Delta V_{th_{\text{st}}} \times (1 - \sqrt{\eta \frac{t_{\text{rec}}}{t_{\text{rec}} + t_{\text{st}}}}), \tag{2}$$

where η is equal to 0.35, and t_{st} and t_{rec} represent the stress and recovery time durations, respectively.

2.1.2 HCI Aging

Hot carriers refer to the electrons or holes in the substrate that attain energies above the average [RRFT16a]. These high energetic carriers, which are the result of high electric fields in the drain region of a transistor are injected into the gate oxide and form interface states and eventually result in performance degradation in the transistor under stress. HCI mainly affects NMOS transistors and has become more severe as the transistor features continue to shrink [CPS08].

HCI results in the change of the threshold voltage of the device under stress. Besides increasing the threshold voltage, HCI reduces the mobility of a device, which leads to a decrease in drain current. Unlike NBTI, there is no recovery for HCI.

HCI effect is due to the switching between '0' and '1' on an NMOS transistor. Thereby, HCI is highly sensitive to the number of transitions that occur in the gate input of the transistor under stress. In fact, the threshold voltage changes sublinearly with the number of transitions that occur in the input of an NMOS transistor. In practice, HCI has a sublinear dependency on the clock frequency, usage time, and the activity factor of the transistor under stress, where activity factor represents the ratio between the number of cycles the transistor is doing transitions and the total number of cycles the device is utilized. In addition, HCI effects depend on the operating temperature [OT12]. The equation given below evaluates the HCI-induced threshold voltage shift [WYB+10, TT08].

$$\Delta V_{th_{\text{HCI}}} = A_{\text{HCI}} \cdot \alpha \cdot f \cdot e^{\frac{V_{dd} - V_{th}}{t_{ox} \cdot E_1}} \cdot t^{0.5}, \tag{3}$$

where t stands for time, α and f for the activity factor and the frequency, respectively. In addition, t_{ox} is the oxide thickness, and E_1 depends on the device specifications, the temperature, and V_{dd}. Further, A_{HCI} is a technology-dependent constant.

As expressed with more details in Section 3, in order to extract the simulation results we deployed HSpice MOSRA (MOS Reliability Analysis) [Syn16] to evaluate the impact of NBTI and HCI on a circuit under stress. MOSRA uses the Reaction-Diffusion (R-D) model discussed in [WYB+10].

2.2 Leakage Currents in MOSFETs

Metal-oxide-semiconductor field-effect transistors (MOSFETs) are by far the most common transistors in digital circuits and have experienced an aggressive scaling process during the last decades, mainly fueled by two forces. First, the increasing demand to put more computation capability onto a single chip, second the potential enablement of faster computation. Indeed, decreasing channel lengths not only reduces area, but also allows electrons (or holes) to pass through transistors faster, and decreasing distances between gates minimize the cell-to-cell travel time [Key05]. Also, due to the shrinkage of transistor and cell dimensions, capacitances decrease in size and, thus, can be charged faster. Besides improving the overall capability and speed of integrated circuits, a further implication of the scaling process is a decrease in the dynamic power dissipation per logic unit due to smaller capacitances, shorter signal propagation delays and the allowance for lower supply voltages in general [Key05]. However, all these improvements come at the price of undesired side-effects like, for example, the static power consumption, a.k.a. leakage current.

Over the years it has become more and more apparent that nanometer-scaled MOSFETs conduct a significant current between V_{dd} and ground even when being switched off. This current, which is often called leakage current or off-current, is not only limited to drain to source leakage, but can occur as leakage through the gate insulator and leakage from drain to bulk and source to bulk as well [Hel09]. Multiple physical effects responsible for these phenomena have been pointed out and investigated in the literature, e.g. subthreshold leakage, drain-induced barrier lowering (DIBL, a.k.a. punchthrough current), gate tunneling, hot carrier effects, diode currents and gate-induced drain leakage (GIDL) [Hel09]. For our investigations mainly the subthreshold conduction is of interest, since it constitutes the dominating source of drain to source leakage and exhibits an exponential dependency on the threshold voltage of transistors. An increase in the threshold voltage reduces undesired effects like the subthreshold leakage, but also degrades the performance of transistors in terms of signal propagation delays. A decrease in the threshold voltage on the other hand has the opposite effect, namely allowing transistors to switch faster while having a higher static power consumption. This trade-off between speed and average power consumption of transistors has fueled the introduction of multi-threshold voltage cell libraries [Hel09]. Indeed, logic cells in advanced technologies usually exist in multiple versions composed of transistors with either low, high or medium voltage thresholds. For signal delay optimizations in the critical path of a circuit cells with a low voltage threshold (LVT) can be selected, while in all paths of a design which are not timing critical, cells with a high voltage threshold (HVT) are chosen to minimize the overall leakage.

2.3 Data-Dependent Leakage Currents in CMOS Logic

CMOS logic gates are constructed in such a way that there exists at least one switched-off MOSFET (i.e., the transistor is *not* forming a conducting channel between source and drain) in any path between V_{dd} and ground during an idle state. This design decision was made in order to ensure that such logic gates only draw a significant current during switching behavior, enabling low power circuitry. However, due to the currents passing through individual inactive, i.e., switched-off, MOSFETs, there is also a perceptible current leaked by CMOS logic gates constructed from them. The magnitude of the leakage current exhibited by a CMOS logic cell depends on the type and formation of switched-off MOSFETs in the path between V_{dd} and ground and the different electric potentials across them. For example, considering a simple CMOS `NOT`, i.e., inverter, gate such as depicted in Figure 2, it can be observed that depending on the input signal either the NMOS or the PMOS transistor is inactive. When assuming that the active, i.e., switched on, transistor creates a perfect conduction channel (which basically means replacing them by ideal wires), one obtains the resulting schematics on the right side of Figure 2 for different inputs. Usually, PMOS and NMOS transistors have different leakage currents, leading to the fact that the static power consumption of this `NOT` gate depends on its input, i.e., on the processed data. This dependency becomes even more obvious in gates with multiple input lines, such as the two-input `NAND` gate in Figure 3. Here, four different formations of inactive transistors can be observed, depending on the input. Clearly, these four cases lead to different leakage currents exhibited by the `NAND` gate. Comparing the two cases (A=0, B=0) and (A=0, B=1), for example, it is obvious that the former, where two switched-off NMOS transistors are connected in series, has a significantly smaller leakage current than the latter. In particular, connecting inactive transistors in series causes a so-called stacking

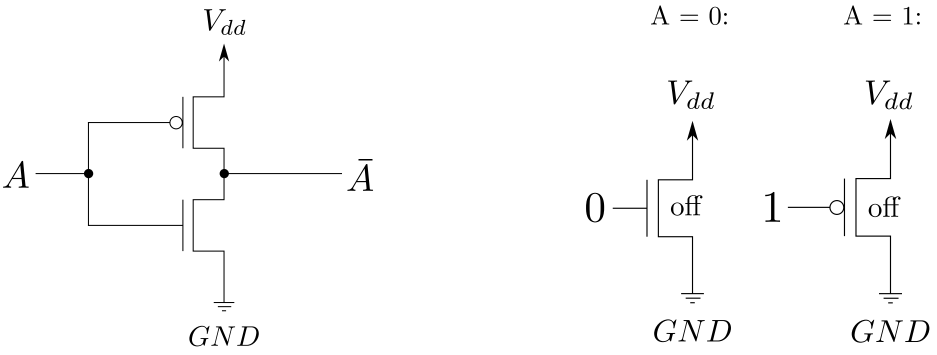


Figure 2: CMOS `NOT` (or inverter) gate (left) and formation of inactive transistors across the power supply path for different inputs (right), assuming perfect conduction for active transistors.

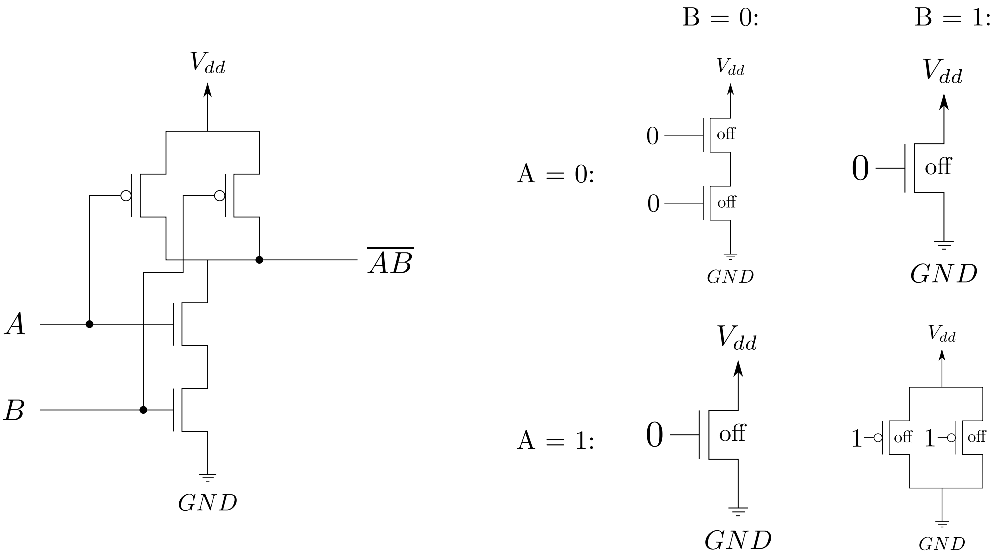


Figure 3: CMOS `NAND` gate (left) and formation of inactive transistors across the power supply path for different inputs (right), assuming perfect conduction for active transistors.

effect [RMMM03]. This effect reduces the current flowing through a stack of two inactive transistors by one order of magnitude compared to a single inactive one [RMMM03]. For this reason, transistor stacking is often used as a leakage current mitigation technique. The largest current is leaked by a CMOS `NAND` gate, however, when the input combination (A=1, B=1) is applied, since two inactive PMOS transistors, connected in parallel, are present between V_{dd} and ground, whose individual leakage currents cumulate.
Of course, assuming perfect conduction for the active transistors in the previous examples is meant as a simplification. In practice, even the two cases (A=0, B=1) and (A=1, B=0), which look identical in Figure 3, would lead to different leakage currents, caused by the difference in the electric potentials across them[1]. Thus, in reality the data dependency is even stronger than in the simplified scenario.

---

[1] This depends for instance on whether a terminal is pulled up/down by another transistor or not.

2.4 Impact of Device Aging on Leakage Currents in CMOS Circuits

As described before, the magnitude of the threshold voltage of a transistor is increased when this transistor is subject to an aging process, due to effects such as BTI and HCI. Since any increase in the threshold voltage of transistors translates to a decrease of the subthreshold currents in the device under test, it has to be expected that aging affects the leakage currents in CMOS devices quite significantly. Indeed, it can safely be assumed that device aging reduces the overall cumulative leakage current of a digital CMOS circuit. Its impact on the exploitability of the static power consumption through side-channel attacks, however, is much harder to predict. In fact, it is *not* the case that the leakage currents of a circuit are reduced by a fixed factor for all possible inputs, which would maintain its input dependency. Caused by multiple factors, the data dependency of the static power consumption of a combinatorial circuit (such as a non-linear substitution box of a block cipher), may shift significantly. For example, two inputs to such a circuit, whose leakage currents are clearly distinguishable in the original state of the device may become less distinguishable due to the aging process. But the opposite situation might occur as well[2]. This depends on the concrete netlist of the circuit, the cells which are used, and whether HCI or NBTI are the predominant aging mechanism in the device under test, influencing NMOS or PMOS transistors respectively. Since HCI is caused by the active switching of NMOS transistors, but NBTI by stable values at the input of PMOS transistors (and partially recoverable), it also depends on the type of aging that is performed on the given circuit (i.e., low or high workload, random or biased input data, etc.). Considering Figures 2 and 3 it is clear that depending on whether the threshold voltage increase is more significant in some transistors than others, e.g. more dominant in NMOS than in PMOS transistors, the data dependency of the gate can change significantly in both directions (i.e., decreasing or increasing). Even two identical transistors (i.e., both NMOS or both PMOS) face a different leakage current reduction depending on the data they have processed in the past. This difference in the impact of aging on specific (types of) transistors is what makes it difficult to anticipate its influence on the vulnerability of devices to static power attacks. Thus, an accurate simulation of the impact of aging mechanisms on the data-dependency of the power consumption can be crucial to evaluate the long-term security of cryptographic devices before the tape-out of a design.

2.5 Targeted Device

The main target for our analysis is a (plain/unprotected) implementation of the PRESENT block cipher synthesized by a 65 nm CMOS standard cell library. PRESENT has been introduced in 2007 as an SPN-based ultra-lightweight block cipher for ubiquitous computing environments with extremely constrained resources [BKL+07]. It consists of 31 rounds and makes use of a 4-bit S-box. Its block size is 64 bits and two different key lengths, 80 and 128 bits, are supported. In this work we consider the 80-bit version, which is called PRESENT-80. In [RPLP08] a serialized hardware architecture of PRESENT-80 is proposed and synthesis results by a 350 nm standard cell library are claimed to require as few as 1000 gate equivalences (GE) for the full cipher. In order to achieve such an area-optimized implementation the data paths are serialized to 4-bit words (one nibble) and only one 4-bit S-box is physically implemented, which thus needs to be shared between the data path and the key schedule. Obviously, this area-driven optimization comes at the price of a greater delay in terms of clock cycles, i.e., less throughput. This particular implementation requires 563 clock cycles to encrypt one plaintext. A block diagram of the serialized architecture is shown in Figure 4.

---

[2] Such a change in the data dependency is usually not to be expected when influencing the (global) operating conditions of a device.

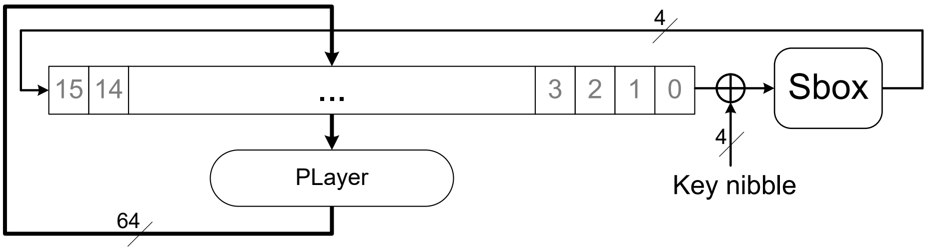


Figure 4: Nibble-serial architecture of the PRESENT block cipher, key schedule not shown.

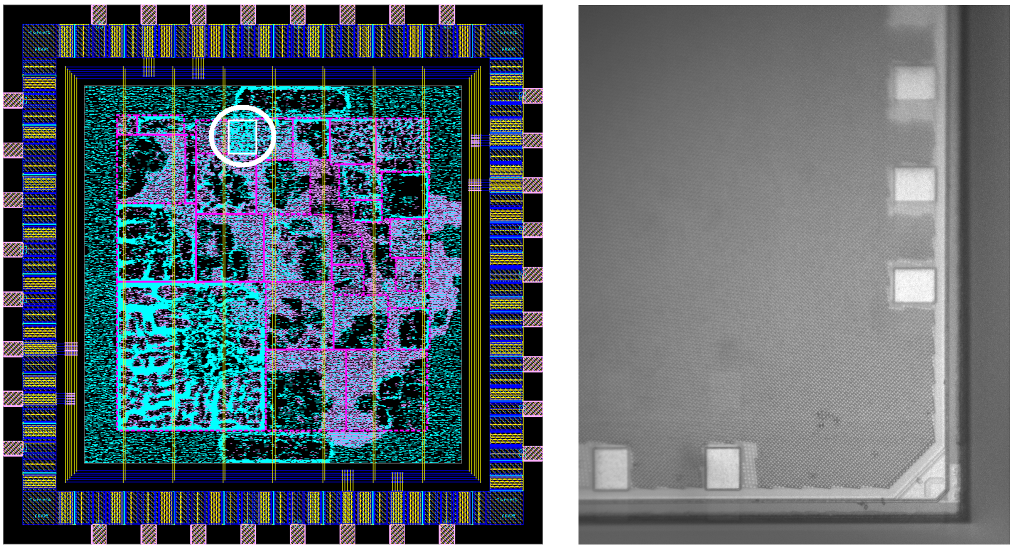


Figure 5: Layout of the 65 nm ASIC with the PRESENT-80 core highlighted in white (left) and a microscopic photograph of the bottom right corner of the chip (right).

We have implemented this PRESENT-80 architecture as one module of the 65 nm CMOS ASIC chip that we developed to serve as a state-of-the-art *device under test* (DUT) for advanced side-channel and fault injection evaluations. The fabricated ASIC features 27 different cipher cores (which are partially equipped with countermeasures against physical attacks) and combines them in a sophisticated configuration- and control-framework in order to operate them according to different application scenarios and to provide fast and easy communication with the outside. A picture of the layout of the chip and a microscopic photograph of the bottom right corner of the fabricated die can be seen in Figure 5. As it is apparent from the photo on the right side, no structural information (apart from the position of the bond pads) can be identified from microscope images of the chip due to the metal fill that is distributed over the whole top layer. However, to highlight where the targeted PRESENT-80 core is located on the chip we have circled and framed it in the layout schematic on the left side of Figure 5. This depiction of the layout was taken from the place-and-route software *Synopsys IC Compiler (Version 2016.12)* during the design process prior to fabrication.

For the tape-out a low-power, multi-threshold voltage, 65 nm CMOS library was used. We have applied the multi-threshold voltage power optimization technique in order to reduce the overall leakage current of the design. In particular, since we did not put any tight timing constraints on the PRESENT-80 core (indeed it was constrained to a clock frequency of 35 MHz) and as the combinatorial circuits (e.g., the S-box) are rather small in this design, mostly standard cells with a high threshold voltage (HVT) and a low drive strength have been selected by the synthesis tool (*Synopsys Design Compiler (Version 2016.12)*). Cells with a low drive strength are smaller in terms of area, but cannot drive large output loads. Cells with a high voltage threshold have larger propagation delays, but exhibit a smaller off-current. Even though we have applied rather loose timing constraints, the architecture requires almost 90% more area (1873 GE) in our post-layout netlist than what is claimed in the original proposal [RPLP08]. Apart from the obviously different libraries that are used, this can on one hand be caused by the fact that in many synthesis scripts for area estimation no timing constraints at all are applied (i.e., the reported numbers always tend to be smaller compared to a manufacturable post-layout netlist). On the other hand, we have prevented the tool from performing specific optimization techniques that remove module boundaries or optimize paths across those boundaries[3].

3 Simulation Results

Our aim is to investigate the impact of aging on the success of static power analysis attacks through conducting both, simulations and silicon measurements. As detailed before, this impact on the exploitability of a device is hard to anticipate. A reduction of the overall leakage current of a device is expected, but the concrete influence on the data dependency and the available information in a side-channel attack needs to be investigated in terms of simulations and practical experiments. Since SCA attacks commonly target the leakage of combinatorial logic, which is also the case in static power analysis attacks [LB08, MMR17], our simulations only consider an S-box module of the target device described in Section 2.5. We deployed the same netlist used for the chip fabrication in the same 65 nm technology node. Note that as explained earlier, the design contains only one instance of the S-box module.

We used Synopsys HSpice for the transistor-level simulations and deployed the HSpice built-in MOSRA Level 3 model to assess the effect of NBTI and HCI aging [Syn16]. Static current values were extracted for the original circuit as well as the aged ones. The effect of aging was evaluated for 8 weeks of device operation in time steps of one week. We considered the operating temperature as 90°C during the aging process and 20°C during the measurement. The supply voltage (V_{dd}) was set to 1.416 V during the aging process and 1.2 V during the measurement. Please note that such operating conditions during aging (90°C and 1.416 V) accelerate the process of device degradation by a factor of about 80 (this is technology dependent) [agi17]. Thus, 8 weeks of aging under these conditions correspond to approximately 640 weeks of normal device operation, which is more than 12 years.

The netlist was fed with randomly generated stimuli during the aging simulation process. With respect to the data dependency of the static power consumption, the amount of leakage current of a combinatorial circuit solely depends on its input value, i.e., there is no dependency on the former input or on the transition between consecutive inputs. Thereby, in the measurement phase we simulated the target circuit (either the original or the aged one) for all possible input values, i.e., 16 cases due to the 4-bit input width of the PRESENT S-box.

Figure 6 depicts the simulation results representing the static current change in an original (0-week age) target circuit as well as the cases where it has been used between 1 and 8 weeks under the above explained aging conditions. In this figure, the static current has been shown when the circuit is fed with different input stimuli. As expected, the magnitude of the static current decreases during the device lifetime regardless of the input stimuli applied during the measurement. Yet, the factor by which the current is reduced

---

[3] We excluded such kinds of optimization in order to not accidentally corrupt the (masking) countermeasures that have been applied to the other cores in the same ASIC.

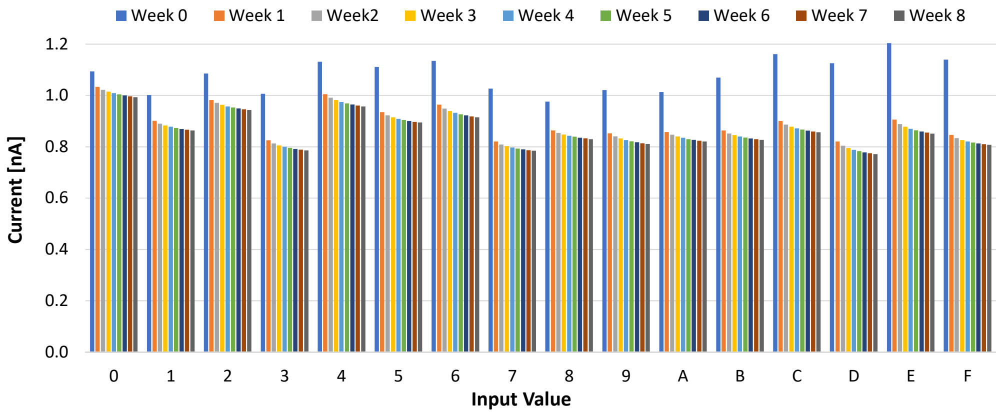


Figure 6: Simulated static current change for different aging duration and input stimuli.

for each respective input differs significantly. In other words, the pattern of the data dependency changes.

As apparent from Equations (1) to (3) in Section 2, the influence of the stress time on the threshold voltage increase is strong in the beginning but declining quickly (asymptotically to the square root, respectively quartic root function). Increasing the threshold voltage in turn reduces subthreshold conduction exponentially [Hel09]. Thus, the decrease of the static current is dominant in the first week of aging ($\approx 24.6\%$). After that it continues to decrease, yet with a slower rate.

3.1 Metrics

In the following, we apply essentially two metrics to assess the side-channel leakage of the PRESENT-80 ASIC implementation in simulation and practice. The first one is the non-specific Welch's t-test which has been proposed as a leakage assessment tool for cryptographic primitives in [GJJR11, CDG$^+$13] and further developed in [SM15, RGV17]. The second one is the Correlation Power Analysis (CPA) introduced in [BCO04]. The former one aims at being independent of any concrete attack scenarios, targeted intermediate values, and hypothetical leakage models. It can therefore be used to quantify the information leakage that is exhibited by a device through a particular side channel without the need to test many different parameter combinations and to perform an actual key-recovery attack. However, the drawback of this method is that, in case a leakage is detected, it does not provide any information about the hardness of an attack, the intermediate value that should be targeted, or the model that can be applied for a successful key recovery. It may even occur that the detected leakage is not related to any sensitive (key-dependent) intermediate value at all. In other words, a leakage that is detected by a t-test does not necessarily point to a vulnerability in the implementation. However, if no leakage can be detected by several non-specific t-tests, most likely the device does not exhibit any exploitable leakage. In this work, we do not only rely on the t-test as an evaluation metric, but additionally perform a CPA on the intermediate value that is processed by the single S-box instance of the design in order to recover (a part of) the key. In a CPA the adversary usually compiles a hypothetical power consumption for each recorded trace, by applying a hypothetical model to an intermediate value that partially depends on a known input and to the other part on a small chunk of the secret. By guessing the secret part, compiling the hypothetical power model under this guess and finally correlating the hypothetical leakage to the measured leakage traces a correctly guessed secret can often be recognized. This is true in case a large enough number of side-channel measurements has been collected, and

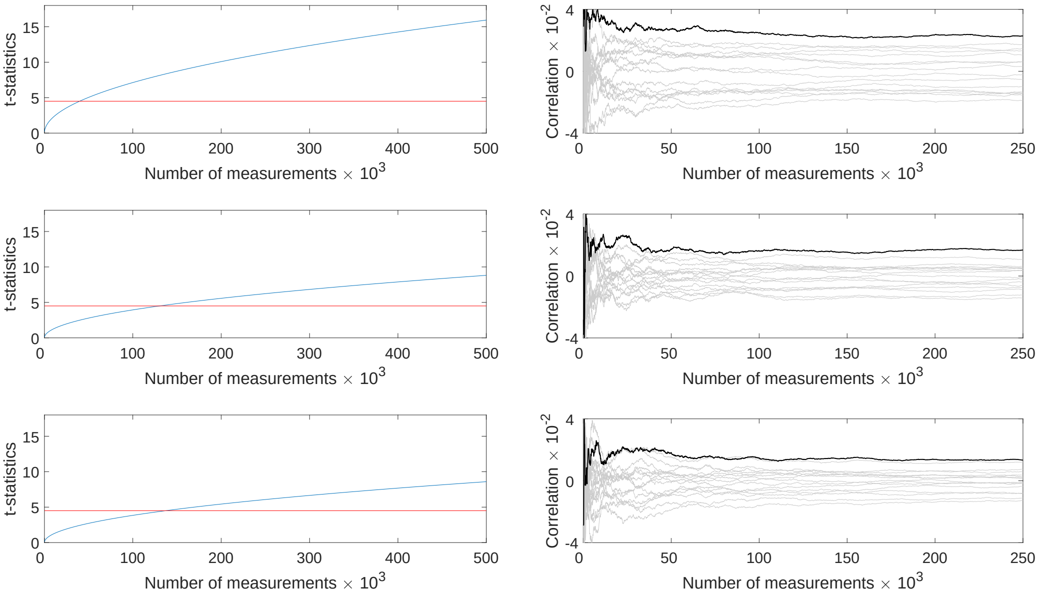


Figure 7: Non-specific t-test and CPA results on simulated leakage measurements from the 65 nm post-layout netlist of the PRESENT S-box after 0, 4 and 8 weeks of aging (top to bottom). The CPA targets a key nibble using the HW of the S-box output.

the chosen leakage model reflects the reality in a sufficiently accurate manner.

3.2 Analysis

In order to determine the influence of aging on the vulnerability of the PRESENT S-box circuit, we apply the previously introduced metrics. In this regard, we have sampled 500,000 values by adding Gaussian distributed noise (with standard deviation of 15×10^{-10}) to the simulated data in Figure 6, suitable for a non-specific Welch's t-test. To be more precise, half of those values have been sampled for randomly chosen inputs, the other half for one fixed input nibble[4]. The t-test is then used to decide whether the two groups, i.e., fixed vs. random, can confidently be distinguished. It is a usual practice to set the threshold for a successful distinguishability to a value of ± 4.5, since this corresponds to a confidence level of 99.999 % to reject the null hypothesis [SM15]. We further made use of half of the collected measurements (those with random associated input) to conduct a CPA attack. To this end, we applied a Hamming weight (HW) model based on the S-box output targeting a key nibble. The corresponding results are depicted in Figure 7. Here, we considered three cases of the simulated static currents: (1) the original circuit, (2) the circuit aged for 4 weeks, and (3) the circuit aged for 8 weeks. As shown later in Section 4, we considered exactly these three cases in our practical investigations.

3.2.1 Discussion

The graphics show that the values for the t-statistic are reduced significantly by aging the circuit. In particular, the original circuit showed a t-value of about 16 after 500,000 measurements, while both, the 4-weeks and 8-weeks aged circuits show a value of around 9. This result confirms not only that aging reduces the data-dependency of the measured currents, but also that the most significant reduction takes place in the first weeks of aging.

---

[4] Note that the order of giving fixed or random inputs is also randomized [CDG+13].

The correlation coefficient of the CPA, targeting the Hamming weight (HW) of the S-box output, reduces by 26.32% after 4 weeks of aging and further 20.22% after another 4 weeks of aging (8 weeks in total)[5]. Please note that the concrete values depend on the sampling of the Gaussian noise and thus may be subject to change when repeating the process. Yet, since we perform the same kind of analysis in the practical experiments, we chose to apply these metrics here.

Interestingly, in Figure 6, we can notice both of the following situations. On one hand, considering the leakage currents of the S-box prior to aging (week 0) for inputs `0x7` and `0xD` for example, it is obvious that these inputs can be distinguished easily by their leakage currents. However, already after the first week of aging, their corresponding leakage currents become very similar and continue to stay in the same range throughout the aging process. On the other hand, when taking a look at the leakage currents for inputs `0x1` and `0x3` for example, one may notice that prior to aging (week 0) they are very similar. But then, after aging the circuit the leakage currents become clearly distinguishable. Although the examples of the former kind (i.e., worse distinguishability after aging) are predominant, it is noteworthy that the latter kind (i.e., improved distinguishability after aging) exists as well. In general it can be concluded that the absolute leakage current difference between the individual input classes decreases by aging the circuit. This, in fact, corresponds to a decrease of the signal [MMR18] in the signal-to-noise ratio (SNR) [MOP07] frequently applied in side-channel analysis. However, due to the existence of the second kind of examples (i.e., improved distinguishability after aging) it can be assumed that it is possible to find combinatorial circuits that have a balanced static power consumption (i.e., showing no/small input dependency) before aging, which then in turn becomes imbalanced after aging the circuit. Thus, we presume that a (naive) leakage current balancing technique for combinatorial circuits is not an appropriate countermeasure against static power analysis attacks, since it can potentially be defeated by aging. Additionally this example demonstrates that template attacks will be difficult to carry out in such a scenario, since the input dependence, used to build the templates, changes over time.

4 Practical Analysis

The main objective of this work is the practical verification of the effects that device aging has on the exploitability of the data-dependent leakage currents in cryptographic hardware. To this end we first introduce the dedicated static power measurement setup that we used and the procedure that we follow to quantify the amount of information that is leaked through the static power side channel by our targeted ASIC implementations. As a next step we describe how the aging process of the device is artificially accelerated by controlling its operating environment. Afterwards, we compare the initial measurements to the ones recorded after the chips are aged for several weeks in order to determine the impact of the aging-related degradation on our ability to extract the secret key via the static power side channel.

4.1 Measurement Setup

As detailed in all previous publications which report results based on experimental static power side-channel analyses [Mor14, PSKM15, MMR17, BCS+17, MMR18], to extract sensitive information from the acquired leakage traces, a dedicated measurement setup is required. In addition to a digital sampling oscilloscope, the main components for such a setup include a precisely controllable climate chamber, a low-noise DC amplifier and a low-pass filter. We have used a similar setup as the one reported in [MMR18] consisting of a CTS climate chamber of type C-40/100, a custom amplifier with a ×1000 gain and

---

[5] Concrete values are compared in Section 4.

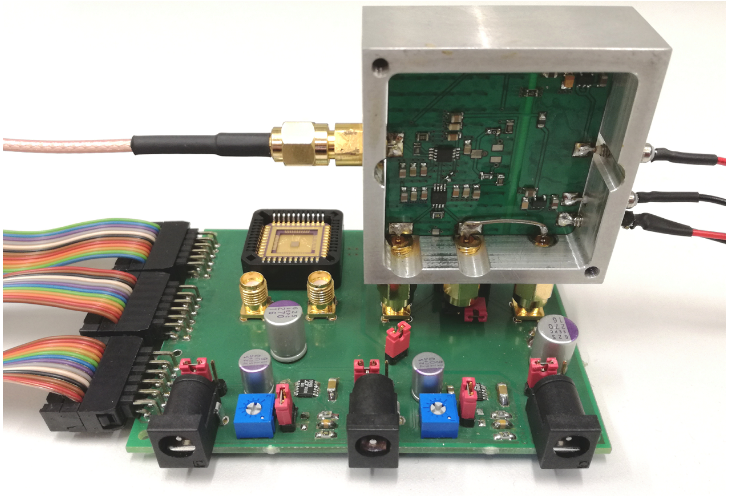


Figure 8: Custom measurement board carrying one of the 65 nm ASIC samples and the DC amplifier.

custom low-pass filter with a cutoff frequency of 100 Hz. For the sampling a LeCroy HRO 66 zi oscilloscope was used. As an interface for the communication between the PC and the ASIC we have used a *Basys 3* board which features an Artix-7 FPGA [DIG]. In contrast to the setup in [MMR18], the FPGA board was not placed inside the climate chamber. We have developed a custom measurement board which holds the ASIC in a PLCC-44 socket and provides SMA connectors for the DC amplifier to measure the voltage drop over a 1 Ω shunt resistor. This board, which can be seen in Figure 8, is powered by the *Basys 3* board and supplies the ASIC with two voltages via two linear voltage regulators, one for the core area (nom. 1.2 V) and one for the IO ring (nom. 3.3 V).

The custom board, together with the amplifier and the ASIC have been placed inside the climate controlled environment, while the remaining parts of the setup are placed outside of the chamber. For the static power measurements on the 65 nm ASIC, we have operated the core-region of the chip with nominal supply voltage of 1.2 V and a constant temperature of 20 °C.

4.2 Procedure

To measure the static power consumption of a device under test (DUT) the target implementation has to be kept in an idle state. More precisely, the attacker needs to control the clock signal in order to suspend the device at the desired cycle of the cryptographic operation and to measure its leakage current. This leakage current can be observed as a DC shift of the static signal, as soon as the effect of the dynamic power consumption is vanished. We follow the procedure introduced in [MMR18] to measure the DC shift that corresponds to the fixed state of the circuit in a particular targeted clock cycle. In our measurements, after the last edge of the clock we ignore the first 100 ms and average all values that are measured in the next 100 ms by the digital sampling oscilloscope to obtain a singular static power value. This procedure is depicted in Figure 9, where T_1 denotes the first 100 ms which are ignored, and T_2 denotes the next 100 ms which are averaged to obtain a quantitative value for the DC shift. The sampling rate was set to 10 MS/s, i.e., 1,000,000 samples were averaged over the 100 ms period. There was no explicit delay considered between the measurements. Yet, due to the communication between the PC and the *Basys 3* as well as between the *Basys 3* board and the ASIC a small implicit delay took place between the individual acquisitions.

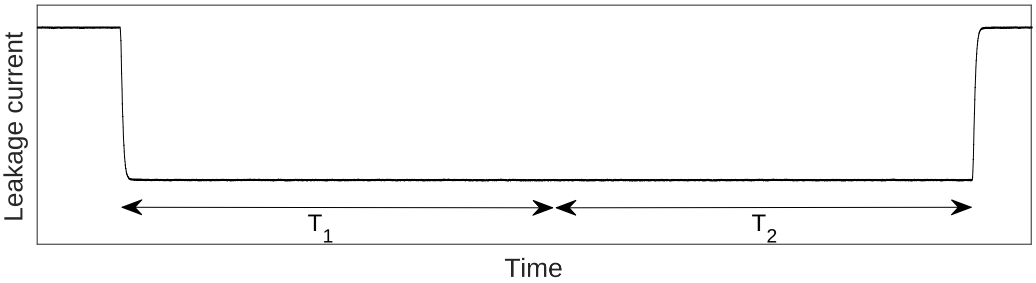


Figure 9: Sample measurement illustrating the procedure. At the beginning of T_1 the clock is stopped, all values in T_1 are ignored. Then all values measured in T_2 are averaged to a singular value representative for amount of static power.

4.3 Results 65 nm ASIC

For these experiments we used **two** completely fresh and unused samples of the developed 65 nm ASIC. In particular, we took two of the dies that have been shipped by the foundry, packaged them using a semi-automatic die bonder into a JLCC-44 package, and started the experiments on those two fresh samples which never performed a single computation before and were power-upped for the first time. Prior to proceeding with the aging acceleration process on the ASIC samples, we first acquired the corresponding reference values for the PRESENT implementations. To this end, we collected 500,000 measurements using the previously described measurement procedure for randomly interleaved fixed and random plaintexts at a constant temperature of 20 °C while the ASIC chips were powered by a 1.2 V supply voltage. The clock was stopped at the end of the first round, when the last state (plaintext ⊕ key) nibble is applied to the S-box circuit of the PRESENT-80 implementation. Acquiring those traces took roughly 3 days. After the first sample was finished we performed the same experiments on the second sample, which took another 3 days. Then we started the aging acceleration process on both chips in parallel.

In order to accelerate the device aging we have operated the ASIC chips for 4 consecutive weeks at a constant temperature of 90 °C, a supply voltage that has been increased by 18% (i.e., 1.416 V instead of 1.2 V) and a continuously high workload (i.e., constantly giving random input to the targeted PRESENT-80 encryption core). Thus, due to the high activity factor, HCI aging will contribute a lot to the device degradation, while the NBTI aging of PMOS transistors is constantly changing between recovery and stress times, due to the randomized input data. In theory, such conditions lead to a much faster device degradation than a normal operation (factor of about 80) [JED16, agi17]. After this period, both chips were disconnected from the power supply and rested for two full days at room temperature without any operation in order to cool down. Afterwards we started another set of measurements at 20 °C and 1.2 V on the first sample, while the second sample was still resting without being powered. Subsequently, we exchanged both chips so that the first ASIC was resting while the second one was measured. Please note that in order to avoid any influences in our measurements stemming from the aging of components in the measurement chain other than the targeted ASIC chips (e.g., the PCB and all electronic components, capacitors/resistors/voltage regulators/etc.), we have used different, but structurally identical, custom measurement boards for the aging acceleration process than the one which was used for the reference measurements. Furthermore, neither the shunt resistor nor the DC amplifier have been placed inside the climate chamber during the aging. In other words, none of the parts that have been present in the heating chamber during the aging acceleration process, with the exception of the ASIC chips themselves, are reused for the measurements, and equally none of the parts that are used in the measurements

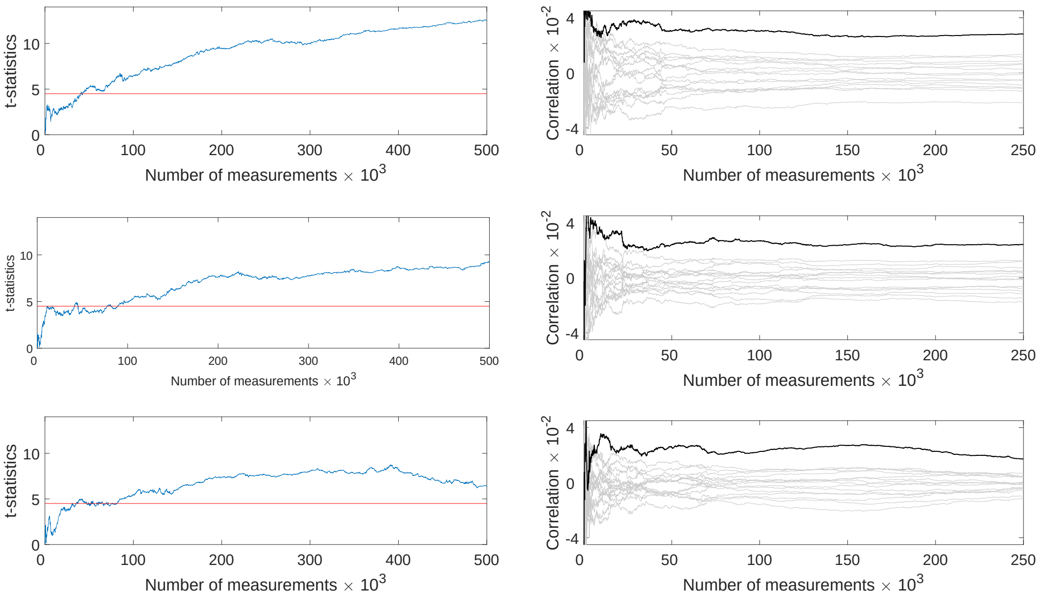


Figure 10: Non-specific t-test and CPA results on static power measurements from the first sample of the 65 nm ASIC after 0, 4 and 8 weeks of aging (top to bottom). The CPA targets a key nibble in the first round using the HW of the S-box output.

have ever been exposed to increased temperatures, supply voltages or to an extremely large workload over a long time period. For the same reason we did not perform the measurements on the chips at increased operating conditions, but instead at 20 °C and 1.2 V. Although, at higher temperatures and voltages the static power side-channel is more informative, we could not consider such conditions for our analysis. At higher temperatures or voltages the ASIC, the DC amplifier and the components on the measurement board would be subject to an accelerated aging process *during* the measurements, which certainly would influence the results and limit their comparability[6].

The described aging process and the subsequent measurements have been performed twice, which allows us to present results for 4 weeks and 8 weeks of intense aging respectively. These settings have been selected according to the simulation results shown in Section 3. For both, after 4-week and 8-week aging, we conducted the same evaluations as those performed on the simulated data, i.e., non-specific t-test and CPA based on the HW of an S-box output at the first round. The corresponding results, showing the development of the exploitability of the first sample over a period of 8 weeks of aging are shown in Figure 10. Similar to the simulation results it can be observed that the vulnerability of the implementation is reduced by aging the circuit. The t-test requires roughly twice as many measurements as on the unaged chip to overcome the 4.5 threshold when the device is aged, independent of whether 4 or 8 weeks of aging are considered. Furthermore, the t-test curves in the two aged cases behave similar up to approximately 400 000 measurements. After this limit the t-values start to decrease again in the 8-weeks aged case, due to some noise influence. Similar observations can be made for the correlation coefficient in the CPA attack on the traces measured for random inputs. Especially when comparing the results up to 400 000 traces (200 000 random ones) it is obvious that the most significant reduction of the exploitability occurs in the first aging period. However, this can still be observed after the whole set of traces in the t-test results, but with a smaller difference. In order to confirm these practical results we repeated the same analysis on the mea-

---

[6] Results for simultaneous aging and measuring of a chip are presented later in the section.

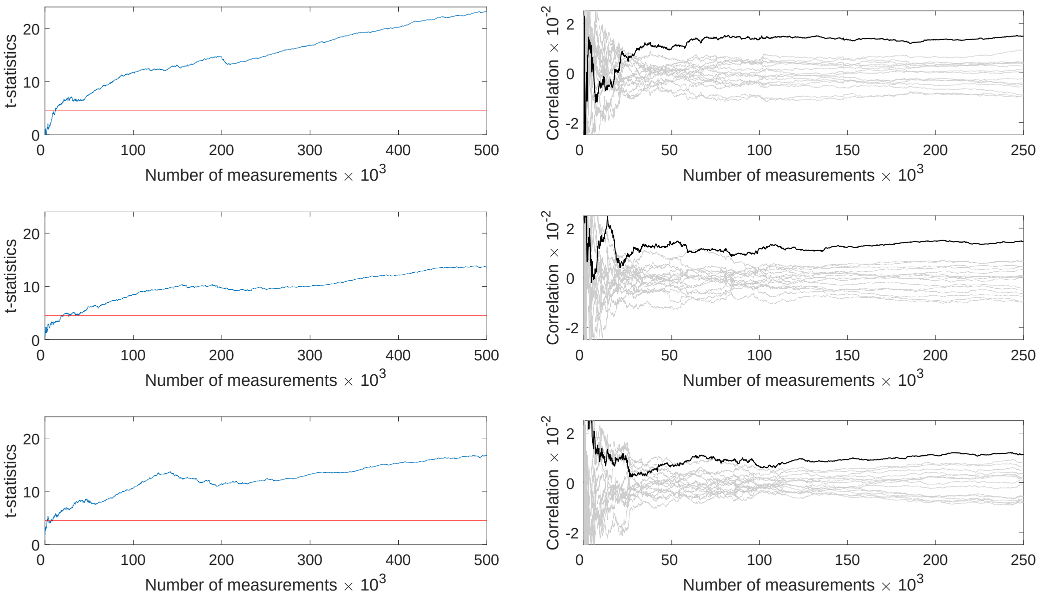


Figure 11: Non-specific t-test and CPA results on static power measurements from the second sample of the 65 nm ASIC after 0, 4 and 8 weeks of aging (top to bottom). The CPA targets a key nibble in the first round using the HW of the S-box output.

surements recorded from the second (identical) ASIC sample as well. The corresponding results are depicted in Figure 11. Interestingly, the t-values are larger than on the other sample, while the CPA performs worse than before. This already shows the impact of process variations on the comparability of side-channel measurements taken from two structurally identical devices. The distinguishability of the fixed and random groups in the t-test is not only (positively) influenced by the leakage of the S-box circuit of the PRESENT implementation (although it certainly has a large contribution) but also by the leakage of the state register, multiplexers and other cells in the design. The CPA on the other hand, as it actively targets the leakage of the S-box, is affected negatively by those leakages as they contribute to the algorithmic noise (when associated to non-targeted state nibbles). Thus, it can be assumed that the cells not belonging to the S-box have a larger data-dependent leakage current on this sample of the 65 nm ASIC than on the other one. Apart from that observation, the results also confirm that aging reduces the available information. However, it can be noticed that the t-values on the measurements recorded after 8 weeks of aging the chip are slightly larger in comparison to the 4 weeks aged chip. We assume this to be a random occurrence. As predicted by the simulations there should be no large difference in the distinguishability between 4 weeks and 8 weeks anyway and both values are significantly smaller in comparison to the original state of the circuit. The CPA results confirm that the S-box circuit is more difficult to attack after 8 weeks than after 4. For a simple comparison of the simulations and the practical results on both ASIC samples we have listed the t-test and correlation values in Table 1.

4.4 Results 150 nm ASIC

In addition to the aging experiments on our self-made 65 nm chip, we have also performed measurements on another ASIC prototype chip which has been manufactured in a less recent technology node, namely 150 nm. In view of recently published results where static power measurements were performed under an increased working temperature and supply voltage

Table 1: Comparison of the simulation and practical measurements on both 65 nm samples.

Experiment	Stage of aging	t-stat.	Corr. coeff.	Avg. total curr.
Simulation	Original device	15.941	0.02283	-
Simulation	4 weeks aged	8.818	0.01682	-
Simulation	8 weeks aged	8.590	0.01340	-
Measurements sample 1	Original device	12.514	0.02801	8.6 µA
Measurements sample 1	4 weeks aged	9.299	0.02410	8.0 µA
Measurements sample 1	8 weeks aged	6.359	0.01718	7.5 µA
Measurements sample 2	Original device	23.251	0.01472	7.5 µA
Measurements sample 2	4 weeks aged	13.647	0.01465	7.2 µA
Measurements sample 2	8 weeks aged	16.710	0.01147	6.9 µA

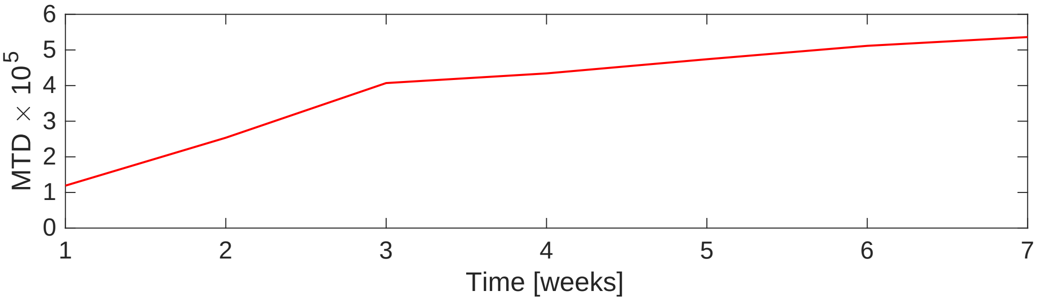


Figure 12: 150 nm ASIC chip, 90 °C, 10 % over voltage, average number of measurements to disclosure (MTD) for a successful key recovery over 7 consecutive weeks under aging process.

over a long time period to amplify the exploitable signal in a side-channel attack [MMR18] we were eager to investigate whether a simultaneous aging and measurement process also leads to reduced exploitable information in the power traces over time. In this regard we took the identical test chip as was used in [MMR18] (but a new, unaged sample) and performed the same kind of measurements on the PRESENT core (i.e., 90 °C, 10 % overvoltage and a 10 ms measurement interval) for a consecutive period of 7 weeks. In other words, during the measurement phase the device was continually aged. Afterwards, we have calculated average MTD (measurements to disclosure) values for each week respectively, corresponding to CPA results on disjoint subsets of the full measurement set per week. The results are presented in Figure 12. As shown, the average number of required measurements increases significantly in the first three weeks. Afterwards, it still increases, yet with a slower rate. A similar behavior can be observed regarding the correlation coefficients for the correct key candidate, as demonstrated in Figure 13. However, in this case the most significant reduction of the correlation coefficient can be observed between the first and the second week of simultaneous aging and measuring.

4.5 Discussion

Our practical analysis, based on real-silicon measurements, shows clearly that the information which is leaked through the static power side-channel is reduced when the circuit is aged. The transistor-level simulations have indicated that the predominant decrease of the exploitable data dependency occurs in the first week(s) of aging. The practical experiments on both 65 nm ASIC samples have confirmed this prediction, especially with respect to the t-test results.

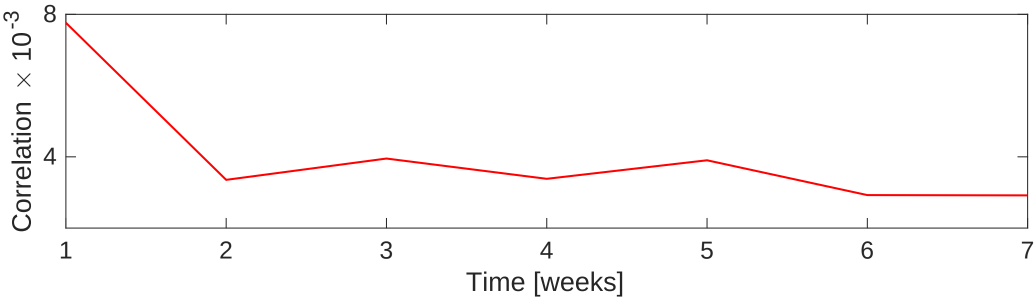


Figure 13: 150 nm ASIC chip, 90 °C, 10 % over voltage, correlation coefficient for the correct key candidate using 5 million traces in each week.

The experiments on the 150 nm ASIC revealed that throughout a process of 7 weeks of simultaneous aging and measuring, the number of traces for a successful key recovery is increased by a factor of roughly 5 and the correlation coefficient is decreased approximately by a factor of 4. However, in order to argue that it generally still seems to pay off to conduct leakage measurements at high temperatures and supply voltages, we refer to [MMR18], where it is shown that on a chip with the same 150 nm technology node the number of measurements to disclosure can be reduced by a factor of approximately 25 compared to room temperature and nominal supply voltage.

5 Conclusion

Due to so-called device aging, the electrical specifications of transistors embedded in integrated circuits change over their device lifetime. This causes the power consumption and timing characteristics of the device to alter over time. Hence, there seems to be a thorough need to examine its effect on the physical security of cryptographic devices. It is noteworthy that such analyses have previously been performed on delay-based PUFs [MvdL14, MHZ16, KDSG17, KDLG16, RRFT16b, RFFT14b, Qu09] and template attacks [KGD18, KDG18].

Here we investigated the effect of device aging on the success of side-channel analysis attacks through the static power consumption (i.e., leakage current). The transistor-level simulations and practical investigations based on real-silicon experiments that we demonstrated indicate that the amount of exploitable information in the leakage current is reduced when the device is aged. Consequently, the corresponding attacks on aged devices require more measurements for a key recovery. Since static power measurements are usually performed when the target device is being operated at a high temperature (and sometimes with high supply voltage), the device is being aged at the same time. We have shown that in such conditions, when the measurement process takes a couple of weeks, the samples collected at different measurement phases do not correspond to each other. Thus, we do believe that all future publications which report analysis results based on static power side-channel attacks need to explicitly state whether and for how long the corresponding measurements have been performed at aging-accelerating conditions. Additionally, it can be of interest to present information about the starting age of any device under test and the order of measurements in case they have been collected in multiple phases from the same chip.

The experiments we showed here were based on an unprotected implementation (i.e., no SCA-countermeasures have been applied). The effect of aging on static power analysis attacks is expected to be more destructive when higher-order leakages of an SCA-protected implementation need to be exploited, since the estimation of higher-order statistical moments is highly sensitive to the noise level. We plan to practically investigate such cases in the future. A further scope for future work was already mentioned in Section 3. It can be of interest to verify whether a (leakage) power-balanced combinatorial circuit can be made vulnerable again by aging the device.

Acknowledgments

The work described in this paper has been supported in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and through the project 271752544 "NaSCA: Nano-Scale Side-Channel Analysis".

References

[agi17] Reliability report 1h 2017, 2017. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/rr/rr.pdf.

[AGST09] Massimo Alioto, Luca Giancane, Giuseppe Scotti, and Alessandro Trifiletti. Leakage Power Analysis Attacks: Well-Defined Procedure and First Experimental Results. In *International Conference on Microelectronics - ICM 2009*. IEEE, 2009.

[AGST10] Massimo Alioto, Luca Giancane, Giuseppe Scotti, and Alessandro Trifiletti. Leakage power analysis attacks: A novel class of attacks to nanometer cryptographic circuits. *IEEE Trans. on Circuits and Systems*, 57-I(2):355–367, 2010.

[AKVM07] M. A. Alam, H. Kufluoglu, D. Varghese, and S. Mahapatra. A comprehensive model for PMOS NBTI degradation: Recent progress. *Microelectronics Reliability*, 47(6):853–862, 2007.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

[BCS+17] Davide Bellizia, Danilo Cellucci, Valerio Di Stefano, Giuseppe Scotti, and Alessandro Trifiletti. Novel Measurements Setup for Attacks Exploiting Static Power using DC Pico-ammeter. In *ECCTD 2017*, pages 1–4. IEEE, 2017.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems - CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

[CCLM14] S. Cha, C-C. Chen, T. Liu, and L. S. Milor. Extraction of threshold voltage degradation modeling due to negative bias temperature instability in circuits with I/O measurements. In *VLSI Test Symp. (VTS)*, pages 1–6, 2014.

[CDG+13] Jeremy Cooper, Elke De Mulder, Gilbert Goodwill, Joshua Jaffe, Gary Kenworthy, and Pankaj Rohatgi. Test Vector Leakage Assessment (TVLA)

Methodology in Practice. International Cryptographic Module Conference, 2013.

[CPC+05] F. Crupi, C. Pace, G. Cocorullo, G. Groeseneken, M. Aoulaiche, and M. Houssa. Positive bias temperature instability in MOSFETs with ultra-thin Hf-silicate gate dielectrics. *Microelectronic Engineering*, 80:130–133, 2005.

[CPS08] S. P. Ching, C. T. Ping, and Y. H. Sun. Studies of the critical ldd area for hci improvement. In *Semiconductor Electronics*, pages 622–625, 2008.

[DBST17] Milena Djukanovic, Davide Bellizia, Giuseppe Scotti, and Alessandro Trifiletti. Multivariate analysis exploiting static power on nanoscale CMOS circuits for cryptographic applications. In *Progress in Cryptology - AFRICACRYPT 2017*, volume 10239 of *Lecture Notes in Computer Science*, pages 79–94, 2017.

[DIG] DIGILENT. Basys 3 Artix-7 FPGA Trainer Board. `https://store.digilentinc.com`.

[EKD+03] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. Razor: a low-power pipeline based on circuit-level timing speculation. In *Microarchitecture 2003*, pages 7–18, 2003.

[GJJR11] Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. A testing methodology for Side channel resistance validation. In *NIST Non-invasive Attack Testing Workshop*, 2011.

[GSST07] Jacopo Giorgetti, Giuseppe Scotti, Andrea Simonetti, and Alessandro Trifiletti. Analysis of data dependence of leakage current in CMOS cryptographic hardware. In *Great Lakes Symposium on VLSI, GLSVLSI 2007*, pages 78–83. ACM, 2007.

[Hel09] Domenik Helms. *Leakage Models for High Level Power Estimation.* PhD thesis, Carl von Ossietzky Universität Oldenburg, 2009.

[JED16] JEDEC. JEP122H: Failure mechanisms and models for semiconductor devices, September 2016. `http://www.jedec.org/standards-documents/docs/jep-122e`.

[KAB+03] Nam Sung Kim, Todd M. Austin, David T. Blaauw, Trevor N. Mudge, Krisztián Flautner, Jie S. Hu, Mary Jane Irwin, Mahmut T. Kandemir, and Narayanan Vijaykrishnan. Leakage current: Moore's law meets static power. *IEEE Computer*, 36(12):68–75, 2003.

[KCC+05] A. T. Krishnan, C. Chancellor, S. Chakravarthi, P. E. Nicollian, V. Reddy, A. Varghese, R.B. Khamankar, and S. Krishnan. Material dependence of hydrogen diffusion: implications for nbti degradation. In *Electron Devices Meeting, IEDM 2005*, pages 688–691, 2005.

[KDG18] N. Karimi, J.-L. Danger, and S. Guilley. On the effect of aging in detecting hardware trojan horses with template analysis. In *On-Line Testing and Robust System Design, IOLTS 2018*, pages 1–6, 2018.

[KDLG16] Naghmeh Karimi, Jean-Luc Danger, Florent Lozach, and Sylvain Guilley. Predictive aging of reliability of two delay pufs. In *Security, Privacy, and Applied Cryptography Engineering - SPACE 2016*, volume 10076 of *Lecture Notes in Computer Science*, pages 213–232. Springer, 2016.

[KDSG17] Naghmeh Karimi, Jean-Luc Danger, Mariem Slimani, and Sylvain Guilley. Impact of the switching activity on the aging of delay-pufs. In *European Test Symposium - ETS 2017*, pages 1–2. IEEE, 2017.

[Key05] Robert W. Keyes. Physical limits of silicon transistors and circuits. *Reports on Progress in Physics*, 68(12):2701–2746, 2005.

[KGD18] Naghmeh Karimi, Sylvain Guilley, and Jean-Luc Danger. Impact of aging on template attacks. In *Great Lakes Symposium on VLSI, GLSVLSI 2018*, pages 455–458. ACM, 2018.

[KHHC11] Seyab Khan, Nor Zaidi Haron, Said Hamdioui, and Francky Catthoor. NBTI monitoring and design for reliability in nanoscale circuits. In *Defect and Fault Tolerance in VLSI and Nanotechnology Systems, DFT 2011*, pages 68–76. IEEE Computer Society, 2011.

[Kim15] K. K. Kim. On-chip delay degradation measurement for aging compensation. *Indian Journal of Science and Technology*, 8(8), 2015.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Advances in Cryptology - CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[LB08] Lang Lin and Wayne Burleson. Leakage-Based Differential Power Analysis (LDPA) on Sub-90nm CMOS Cryptosystems. In *International Symposium on Circuits and Systems - ISCAS 2008*, pages 252–255. IEEE, May 2008.

[LK11] Y. Lee and T. Kim. A fine-grained technique of NBTI-aware voltage scaling and body biasing for standard cell based designs. In *Asia and South Pacific Design Automation Conference - ASP-DAC 2011*, pages 603–608, 2011.

[MHZ16] Mohd Syafiq Mispan, Basel Halak, and Mark Zwolinski. NBTI aging evaluation of puf-based differential architectures. In *On-Line Testing and Robust System Design, IOLTS 2016*, pages 103–108. IEEE, 2016.

[Miz08] E. Mizan. Efficient fault tolerance for pipelined structures and its application to superscalar and dataflow machines. Ph.D. thesis, Electrical and Computer Engineering Dept., University of Texas At Austin, 2008.

[MMR17] Thorben Moos, Amir Moradi, and Bastian Richter. Static Power Side-Channel Analysis of Threshold Implementation Prototype Chip. In *Design, Automation, and Test in Europe - DATE 2017*, pages 1324 – 1329. IEEE, 2017.

[MMR18] Thorben Moos, Amir Moradi, and Bastian Richter. Static Power Side-Channel Analysis - A Survey on Measurement Factors. *IACR Cryptology ePrint Archive*, 2018:676, 2018.

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards.* Springer, 2007.

[Mor14] Amir Moradi. Side-Channel Leakage through Static Power — Should We Care about in Practice? In *Cryptographic Hardware and Embedded Systems - CHES 2014*, volume 8731 of *Lecture Notes in Computer Science*, pages 562–579. Springer, 2014.

[MSVK06] S. Mahapatra, D. Saha, D. Varghese, and P.B. Kumar. On the generation and recovery of interface traps in MOSFETs subjected to NBTI, FN, and HCI stress. *IEEE Transactions on Electron Devices*, 53(7):1583–1592, 2006.

[MvdL14] Roel Maes and Vincent van der Leest. Countering the effects of silicon aging on SRAM pufs. In *Hardware-Oriented Security and Trust, HOST 2014*, pages 148–153. IEEE Computer Society, 2014.

[NBRR13] C. Nunes, P. F. Butzen, A. I. Reis, and R. P. Ribas. BTI, HCI and TDDB aging impact in flip-flops. *Microelectronics Reliability*, 53(9–11):1355–1359, 2013.

[NRS11] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of nonlinear functions in the presence of glitches. *J. Cryptology*, 24(2):292–321, 2011.

[OT12] F. Oboril and M. B. Tahoori. Extratime: Modeling and analysis of wearout due to transistor aging at microarchitecture-level. In *Dependable Systems and Networks - DSN 2012*, pages 1–12, 2012.

[PSKM15] Santos Merino Del Pozo, Francois-Xavier Standaert, Dina Kamel, and Amir Moradi. Side-Channel Attacks from Static Power: When Should we Care? In *Design, Automation, and Test in Europe - DATE 2015*, pages 145–150. IEEE, 2015.

[Qu09] Gang Qu. Temperature-aware cooperative ring oscillator PUF. In *Hardware-Oriented Security and Trust, HOST 2009*, pages 36–42. IEEE Computer Society, 2009.

[RFFT14a] M. T. Rahman, D. Forte, J. Fahrny, and M. Tehranipoor. ARO-PUF: An aging-resistant ring oscillator PUF design. In *Design, Automation Test in Europe - DATE 2014*, pages 1–6, 2014.

[RFFT14b] Md. Tauhidur Rahman, Domenic Forte, Jim Fahrny, and Mohammad Tehranipoor. ARO-PUF: an aging-resistant ring oscillator PUF design. In *Design, Automation, and Test in Europe - DATE 2014*, pages 1–6, 2014.

[RGV17] Oscar Reparaz, Benedikt Gierlichs, and Ingrid Verbauwhede. Fast leakage assessment. In *Cryptographic Hardware and Embedded Systems - CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 387–399. Springer, 2017.

[RMMM03] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer cmos circuits. *Proceedings of the IEEE*, 91(2):305–327, 2003.

[RPLP08] Carsten Rolfes, Axel Poschmann, Gregor Leander, and Christof Paar. Ultra-lightweight implementations for smart devices - security for 1000 gate equivalents. In *Smart Card Research and Advanced Applications, CARDIS 2008*, volume 5189 of *Lecture Notes in Computer Science*, pages 89–103. Springer, 2008.

[RRFT16a] M. T. Rahman, F. Rahman, D. Forte, and M. Tehranipoor. An aging-resistant ro-puf for reliable key generation. *IEEE Trans. on Emerging Topics in Computing*, 4(3):335–348, July 2016.

[RRFT16b] Md. Tauhidur Rahman, Fahim Rahman, Domenic Forte, and Mark Tehranipoor. An aging-resistant RO-PUF for reliable key generation. *IEEE Transactions on Emerging Topics in Computing*, 4(3):335–348, 2016.

[RTY+17] D. Rossi, V. Tenentes, S. Yang, S. Khursheed, and B. M. Al-Hashimi. Aging benefits in nanometer CMOS designs. *IEEE Transactions on Circuits and Systems II*, 64(3):324–328, 2017.

[Sch07] Dieter K. Schroder. Negative bias temperature instability: What do we understand? *Microelectronincs Reliability*, 47(6):841–852, 2007.

[Sha12] Eitan N. Shauly. CMOS Leakage and Power Reduction in Transistors and Circuits: Process and Layout Considerations. *Journal of Low Power Electronics and Applications*, 2(1):1–29, 2012.

[SKR+13] O. Sinanoglu, N. Karimi, J. Rajendran, R. Karri, Y. Jin, K. Huang, and Y. Makris. Reconciling the IC test and security dichotomy. In *European Test Symposium - ETS 2013*, pages 1–6, 2013.

[SM15] Tobias Schneider and Amir Moradi. Leakage assessment methodology - A clear roadmap for side-channel evaluations. In *Cryptographic Hardware and Embedded Systems - CHES 2015*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.

[SVRC15] Ketul B. Sutaria, Jyothi B. Velamala, Athul Ramkumar, and Yu Cao. *Compact Modeling of BTI for Circuit Reliability Analysis*, pages 93–119. Springer New York, 2015.

[Syn16] Synopsys. HSPICE User Guide: Basic Simulation and Analysis, 2016.

[TT08] A. Tiwari and J. Torrellas. Facelift: Hiding and slowing down aging in multicores. In *Microarchitecture 2008*, pages 129–140, 2008.

[WERR13] A. Wiltgen, K. A. Escobar, A. I. Reis, and R. P. Ribas. Power consumption analysis in static cmos gates. In *Symposium on Integrated Circuits and Systems Design - SBCCI 2013*, pages 1–6, 2013.

[WYB+10] W. Wang, S. Yang, S. Bhardwaj, S. Vrudhula, F. Liu, and Yu Cao. The impact of NBTI effect on combinational circuit: modeling, simulation, and analysis. *IEEE Transactions on Very Large Scale Integration Systems*, 18(2):173–183, 2010.

[ZKN+06] S. Zafar, Y. Kim, V. Narayanan, C. Cabral, V. Paruchuri, B. Doris, J. Stathis, A. Callegari, and M. Chudzik. A comparative study of nbti and pbti (charge trapping) in sio2/hfo2 stacks with fusi, tin, re gates. In *Symposium on VLSI Technology 2006*, pages 23–25, 2006.

3.4 Unrolled Cryptography on Silicon

Publication Data

> Thorben Moos. Unrolled cryptography on silicon A physical security analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):416–442, 2020

The acceptance rate for Volume 2020 of the IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES) was **26,8%** [Acca].

Content This work analyzes the side-channel security of a fully-unrolled low-latency block cipher implemented on a 40 nm ASIC. In particular, it is investigated whether round-unrolling offers an intrinsic resistance to side-channel analysis attacks. When the unrolled circuit's input is reset to a random value between each operation a natural resistance to dynamic power attacks can indeed be observed. In fact, even 100 million measurements are insufficient to extract the secret key. However, it is also suggested and demonstrated that static power side-channel adversaries, especially those with clock control, are not negatively affected by the random reset as they do not exploit the leakage of a transition but rather that of a static state.

Contribution The author of this thesis is the sole author of this publication.

Unrolled Cryptography on Silicon

A Physical Security Analysis

Thorben Moos

Ruhr University Bochum, Horst Görtz Institute for IT Security, Germany
thorben.moos@rub.de

Abstract. Cryptographic primitives with low-latency performance have gained momentum lately due to an increased demand for real-time applications. Block ciphers such as PRINCE enable data encryption (resp. decryption) within a single clock cycle at a moderately high operating frequency when implemented in a fully-unrolled fashion. Unsurprisingly, many typical environments for unrolled ciphers require protection against physical adversaries as well. Yet, recent works suggest that most common SCA countermeasures are hard to apply to low-latency circuits. Hardware masking, for example, requires register stages to offer resistance, thus adding delay and defeating the purpose of unrolling. On another note, it has been indicated that unrolled primitives *without* any additional means of protection offer an intrinsic resistance to SCA attacks due to their parallelism, asynchronicity and speed of execution. In this work, we take a closer look at the physical security properties provided by unrolled cryptographic IC implementations. We are able to confirm that the nature of unrolling indeed bears the potential to decrease the susceptibility of cipher implementations significantly when reset methods are applied. With respect to certain adversarial models, e.g., ciphertext-only access, an amazingly high level of protection can be achieved. While this seems to be a great result for cryptographic hardware engineers, there is an attack vector hidden in plain sight which still threatens the security of unrolled implementations remarkably – namely the static power consumption of CMOS-based circuits. We point out that essentially all reasons which make it hard to extract meaningful information from the dynamic behavior of unrolled primitives are not an issue when exploiting the static currents for key recovery. Our evaluation is based on real-silicon measurements of an unrolled PRINCE core in a custom 40 nm ASIC. The presented results serve as a neat educational case study to demonstrate the broad differences between dynamic and static power information leakage in the light of technological advancement.

Keywords: Unrolled Cryptography · Low-Latency Cryptography · PRINCE · Side-Channel Analysis · Static Power SCA · SPSCA

1 Introduction

Physical security becomes a concern whenever cryptography is deployed in a field that puts the hardware responsible for executing cryptographic primitives in a potentially hostile environment. Years of academic and industrial research have revealed the unpleasant truth that no universal solution exists to protect cryptographic devices from key recovery attacks when they are forced to operate under permanent physical exposure to untrusted parties. Although significant advances have been made in developing dedicated protection mechanisms against this threat, there is still neither one guaranteeing full resistance, nor any that is universally applicable to all hardware and software implementations alike (without significant adjustments).

Glitch-Resistant Masking.

With respect to the protection of hardware implementations against passive and non-invasive physical attacks, glitch-resistant masking (a.k.a. hardware-based masking or hardware masking) has become one of the most promising research directions. This particular field was sparked by the introduction of threshold implementations (TIs) in 2006 [NRR06] and has been complemented by a number of further schemes (e.g., [RBN+15, CRB+16, GMK16, GMK17, GM17, GM18, GIB18, FGP+18]) that are summarized and analyzed in [MMSS19]. The common denominator between all of them is the one vital ingredient strictly required to achieve provable security in the presence of glitches, namely the correct instantiation of register stages (see [MMSS19]). Registers are a fundamental part of the approach as they prevent the propagation of glitches between combinatorial (sub-)circuits. Naturally, such a concept is not applicable to implementations where the inclusion of clocked memory elements contradicts a certain design goal, as it is the case for fully-unrolled low-latency circuits. In contrast to other common block cipher implementations strategies (e.g., round-based or serialized), an unrolled circuit is a fully combinatorial representation of the whole encryption (resp. decryption) function without any memory elements incorporated[1]. Clearly, such a design strategy has significantly higher demands in terms of area usage, as no part of the cipher, e.g., a substitution box (Sbox) or a round function, may be reused during the cryptographic operation. Yet, the unrolled implementation style enables the fastest possible execution as it avoids the additional delay to store or synchronize intermediate results. In summary, there is an inherent conflict between state-of-the-art hardware-based masking and the desired high-speed single-cycle execution property of low-latency ciphers. The difficulty of combining low-latency performance with glitch-resistant masking has been extensively discussed at Asiacrypt 2016 for FPGA platforms [MS16a]. Schneider *et al.* attempt to balance the trade-off between physical security and speed of execution in multiple different case studies. One of the considered variants is to mask only the outer rounds of a block cipher and leave the rounds in the middle unrolled and unprotected. Another is the realization of implementations equipped with hardware masking (e.g., TI) as asynchronous circuits, i.e., regular register stages are included but controlled in a clock-less and self-timed fashion. Yet, none of these options fully preserves the desired low-latency characteristic. The only scheme that enables secure masking in the presence of glitches without strictly requiring synchronization stages (and the ensuing latency penalty) has been proposed by Gross *et al.* at CHES 2018 [GIB18]. The general concept is based on the observation that no register stages are needed between consecutive masked non-linear operations, when skipping the share compression that is usually performed to reduce the number of output shares to its minimum. Yet, by applying this technique to mask a certain function, the number of shares required per intermediate result (and therefore the size of the circuit) grows exponentially in the number of subsequent non-linear operations [GIB18]. Naturally, a full cipher instantiation contains a large number of non-linear operations, making the technique rather impractical for a fully-unrolled block cipher. Without integrating at least a couple of resharing and compression stages, containing registers and demanding the addition of fresh randomness, the circuit size and the number of output shares would simply explode.

Unrolling as an Implicit Countermeasure.

We summarize that applying hardware masking to unrolled ciphers is neither trivial nor cheap. None of the available options actually preserves the low-latency property at a reasonable price. However, it has been argued before that such concepts may not even be required in order to achieve a proper level of resistance against side-channel attacks. In

---

[1] The concept of unrolled hardware implementations should not be confused with the common optimization strategy of (loop-)unrolling in software implementations.

fact, even before dedicated low-latency primitives became prominent in cryptography, it was pointed out that the nature of unrolling itself may serve as a decent countermeasure against classical passive attacks, especially when certain usage and design principles are respected. The underlying observation is simply that the fast, asynchronous and highly-parallel execution actively prevents an adversary from capturing the dissipation of the target part of the circuit in sufficiently high quality. Bhasin *et al.* proposed at CT-RSA 2010 to unroll hardware implementations of cryptographic primitives in order to prevent adversaries from learning sensitive data through physical measurables [BGSD10]. The crucial prerequisite is hereby that the data path is cleared between each two consecutive encryptions, which obviously halves the available throughput. In such a scenario the adversary can not predict the Hamming distance (HD) between consecutively processed values in the first round(s) and learns less information. While first experiments focused on the DES block cipher [BGSD10], similar results showing the effectiveness of unrolling against SCA attacks have been demonstrated on the AES as well [MMP11].

In 2012 the first dedicated low-latency block cipher was introduced by the name of PRINCE [BCG+12]. This primitive has been specifically developed to be implemented in a fully-unrolled fashion in order to encrypt and decrypt data efficiently in a single clock cycle. Such a lightweight and high-speed encryption engine is a crucial component for the secure communication in pervasive computing environments with real-time security needs. The intrinsic SCA resistance attributed to the unrolled implementation style may not have been the primary objective during the design process, but it certainly is a welcome side effect as pervasive computing solutions are often threatened by physical adversaries as well. In order to assess the physical security properties of unrolled primitives, several articles have analyzed PRINCE implementations regarding their susceptibility to side-channel attacks, e.g., [YHA15], [MS16a], [YHA17a], [YHA17b] and [CSR+19]. The main focus of these works is finding efficient ways to exploit the observable leakage despite the challenges that the single-cycle execution presents in that regard. The applied techniques range from frequency analysis [CSR+19] to the smart selection of Points-of-Interest (PoI) [YHA15]. Others have pointed out that unrolled ciphers are particularly susceptible in chosen-input scenarios [YHA17a]. These attempts emphasize the additional exploitation effort that has to be invested in order to analyze the security of unrolled implementations, but also the importance of considering different adversary models.

Static Power Side-Channel Analysis (SPSCA).

The landscape of power analysis attacks has changed significantly over the years. The continuous down-scaling of circuit technology has led to a decline of the dynamic power consumption per individual logic unit due to smaller capacitances and supply voltages involved. As a result, it becomes increasingly difficult to target the dissipation of small parts of a circuit in divide and conquer based power analysis attacks. The progressive decrease of propagation delays only benefits this development. At the same time, the static power consumption intensifies in newer technology generations and reaches a significant magnitude in sub-100 nm complementary metal-oxide-semiconductor (CMOS) technology [Moo19]. Hence, it is fair to wonder whether such contrary trends may lead to a shift of the primarily targeted side channel when considering implementations in advanced technology nodes. Numerous advances have been made in recent literature towards a better understanding of the static power consumption of CMOS-based circuits as a source of information leakage. At CHES 2019 it was demonstrated that the difference in susceptibility between two successive CMOS technology nodes can be as large as a 10-fold increase [Moo19]. It was also shown that exponential dependencies of this side channel on the temperature and the supply voltage can be exploited by adversaries to escalate the leakage of information [Moo19, MMR20]. Finally, it was discovered that SPSCA attacks can be performed without obtaining control over the clock signal of the device under test (DUT) when sensitive intermediates remain in the circuit after cryptographic operations and are not subject to an immediate modification [Moo19]. This is particularly relevant for our work, as unrolled circuits, due to the nature of their usual applications, are commonly deployed without any reset signal or key-removal mechanisms, as high performance is often the main criterion. Unrolled circuits which are instantiated without any considerations of this issue, are a prime example of implementations where the full state, containing all sensitive intermediates, remains in the circuit between any two consecutive encryptions. Yet, this side channel has never been considered as a complementary attack vector when evaluating the SCA security of unrolled circuits.

FPGA vs. ASIC.

Latency-optimized ciphers are primarily attractive for ASIC platforms. A cryptographic primitive, like any kind of computation, can unfold its full potential with respect to execution speed when realized in an advanced IC technology node as a semi-custom (standard-cell-based) or full-custom design. Hence, a striking issue with all the previously listed works, analyzing the physical security of unrolled PRINCE implementations, is simply that all of them are based on FPGA case studies. At first glance, this may not appear to be an overly limiting factor for the general validity of the reported results. FPGA case studies are frequently used to make generalized statements about hardware implementations. However, specifically for exceptional implementation styles such as unrolling, it is not always possible to transfer conclusions from FPGA to ASIC platforms in a meaningful way. To illustrate the discrepancy between the two hardware platforms in more detail, we refer to the so-called *cost of programmability* [KR07]. According to the seminal work by Kuon *et al.* [KR07], a fully combinatorial representation of a function (such as unrolled PRINCE) requires about **35** times as much area on an FPGA as on a standard-cell-based ASIC, due to the structure of the programmable fabric. Clearly, such a significant increase in the number of gates involved in the computation leads to a much higher power consumption and delay as well. In particular, the authors observed that regular logic designs are more than **4** times slower on an FPGA, while consuming **14** times as much dynamic power as an equivalent ASIC design in the considered 90 nm reference technology [KR07]. Obviously, a 300% faster circuit which consumes 93% less dynamic power is significantly harder to exploit via side-channel analysis. In summary, without an ASIC-based case study, an important benchmark is missing in order to understand how susceptible low-latency cryptography is towards attacks when implemented in its predestined environment.

1.1 Our contribution

We present an extensive analysis of the physical security level that an unrolled, latency-optimized, cryptographic primitive can provide when implemented in state-of-the-art ASIC technology. Surprisingly, no similar case studies seem to exist in public literature, despite their importance for the cryptographic community as well as the industry sector, showcased by the deployment of such primitives in real-world security produts[2]. By performing our analysis we contribute and discover a variety of novelties. In summary, we find that a few comparably inexpensive usage principles can greatly reduce the information leakage of unrolled primitives through dynamic circuit emanations. The static leakage on the other hand, due its different nature, remains informative and requires special care. Our observations can be used to guide the secure (and low-cost) implementation of unrolled cryptography on silicon and even the protocol design surrounding it. Furthermore, our case study is of educational value as it highlights the broad conceptual differences between

---

[2] The LPC55S microcontroller series by NXP semiconductors for example deploys PRINCE for memory encryption.

static and dynamic power information leakage in a simple and vivid manner. We express all contributions of this work in more detail in the following sub-categories:

Effectiveness of Unrolling as a Countermeasure.

For the first time in public literature, we perform a physical security analysis of an unrolled ASIC implementation of the low-latency cipher PRINCE. Our experiments on the custom 40 nm ASIC confirm that it is straightforward to extract parts of the secret key through side-channel attacks when the adversary obtains knowledge of consecutively encrypted plaintexts under a fixed key. In such a case, the Hamming distance (HD) between consecutively processed first-round Sbox outputs serves as an efficient distinguisher. However, it is claimed in [BGSD10] that clearing the data path between each two encryptions is an effective protection against this threat. We evaluate whether this claim holds for our target. Since *clearing* the data path is a rather vague description of the action to be performed, we test 4 different reset (i.e., clearing) methods and evaluate their worth against their cost. Our analysis shows that setting the plaintext input of the unrolled circuit to a random state (unknown to the adversary) between each two encryptions, while leaving the key constant, is most cost-effective and delivers a high level of resistance.
We also argue that unrolled cryptography, by nature, is extremely resistant to attacks when considering an adversary model with ciphertext-only access. In particular, it is very hard to exploit an unrolled implementation from the ciphertext side (i.e., targeting the last round(s)) when using the dynamic power consumption as a source of information leakage. The signal-to-noise ratio (SNR) degrades very quickly after the first round(s) of the implementation and the asynchronicity of signals in later rounds grows significantly. This resistance may be exploited by designers in such a way that unrolled primitives are used in protocols or modes of operation where an adversary may obtain the ciphertexts but not the plaintexts. Intuitively, such a scenario appears reasonably often in real-world applications, since the ciphertext is commonly transmitted over an insecure channel while the plaintext is often kept secret.

Impact of Static Power Leakage on Unrolled Crypto.

Both observations, the effectiveness of resetting the circuit to a random state between encryptions and the difficulty to exploit the later rounds, give reason for optimism regarding the SCA security of unrolled cryptography. It appears that unrolled circuits, if carefully used, can provide a high level of protection against common dynamic power SCA attacks at a comparably low cost. However, we demonstrate that this is not the case when analyzing the static power for key recovery. Obviously, the static power consumption does not leak information about a transition between consecutive states, but only about the current state of the circuit. Hence, any reset method is conceptually ineffective, as the previous state of the circuit is no part of the leakage function and does not have any impact on the static power consumption[3]. Additionally, a static power adversary can target each round of the unrolled block cipher with approximately the same effort due to the value-based information leakage. All logic gates leak at the same time about their inputs and the intensity of their leakage does not depend on their position in the circuit (e.g., how close to the input or output they are located). Accordingly, attacks with ciphertext access on the last round are not expected to be any more complex than attacks with plaintext access on the first round. This is a valuable asset for an adversary, as a common first-round attack on PRINCE can recover at most 64 bits of information about the 128-bit key. To retrieve more information, either a deeper hypothesis into the second round needs to be

---

[3] Reset methods can be effective it the adversary can *not* control the clock. This is discussed later in more detail.

made, or the last round has to be targeted. While this is not problematic for a static power adversary, an attacker exploiting the dynamic currents might struggle significantly.

Dynamic vs. Static Comparison.

We provide a detailed comparison between the two essential power consumption side channels, dynamic and static, with respect to a cryptographic primitive realized in 40 nm ASIC technology. Earlier comparisons between both side channels exist in the literature [PSKM15, MMR17]. However, our results are based on a more recent semiconductor technology node and exploit thermal as well as voltage dependencies for both, dynamic and static power attacks, to maximize the signal-to-noise ratio (SNR). Our observations clearly help to understand the current state of the technology-scaling-induced race between the two side channels.

Static Power Novelties.

This is the first work that presents static power side-channel attacks on an ASIC implementation of a cryptographic primitive in such an advanced technology node (40 nm). Previous results, which in part also exploit thermal and voltage dependencies, have been reported for ASICs manufactured in 65 nm [PSKM15, KMM19, Moo19], 90 nm [Moo19] and 150 nm [MMR20] technology. This is also the first work that reports SPSCA attacks on an unrolled cryptographic primitive on any platform.

2 Preliminaries

Before discussing the target and the results of our simulations and practical experiments, we shortly revisit a few concepts that are crucial for the understanding of this work. We conclude this section with a toy example for illustration purposes.

2.1 Useful and Useless Transitions in Logic Circuits

The main contributors to the dynamic power consumption of today's physical logic circuits are the charging, discharging and short-circuit currents which are consumed during the transition of a CMOS gate's output from low to high or vice versa [MOP07]. Such a transition of a logic gate's output, a.k.a. toggle, can either be of useful or useless nature. Useful transitions are required to ensure correct functionality, while useless ones are not. A sequence of two useless transitions, e.g., $0 \rightarrow 1 \rightarrow 0$ or $1 \rightarrow 0 \rightarrow 1$, is called a glitch. It has been known for a long time that such glitches are responsible for unnecessary energy loss in combinatorial logic circuits [LvMJ95]. The concrete number of glitches occurring in the evaluation of a combinatorial circuit mainly depends on the logic depth of the circuit, the fanout of each gate and how balanced the propagation delays are. In logic circuits with a large logic depth and a significant fanout per gate the power consumption caused by glitches can be immense. In 1995 already, the authors of [LvMJ95] have shown an example where 60% of the switching activity in an 8x8 array multiplier is caused by glitches and therefore unnecessary. In a 16x16 array multiplier even 77% of switching activity corresponds to glitches [LvMJ95]. In our unrolled PRINCE circuit, glitches account for 96% of all gate toggles on average when both inputs, key and plaintext, make a random transition. They still account for almost 92% on average when the key remains fixed and only the plaintext is changed from one random value to another.

Table 1: Input-dependent leakage current of a 2-input NAND gate in 45 nm technology taken from [AO14].

Input	Leakage Current [nA]
0,0	57.63
0,1	38.55
1,0	72.27
1,1	107.07

2.2 Data-Dependent Static Power Consumption

The static power consumption of CMOS-based circuits has become a relevant source of energy dissipation in more recent years due to the semiconductor technology moving towards nanometer dimensions [AO14]. It is well-known that this share of a circuit's power consumption depends on the logical values that are applied to the inputs of logic gates. In [KMM19] it is described how the structure of CMOS gates contributes to the severe input dependency of their leakage currents. To gain an impression of the magnitude and data dependency of the leakage currents conducted by individual gates we refer the reader to [AO14], where typical values for common gates in different nanometer-scaled technologies are presented. As an example, we provide the input-dependent leakage currents for a 2-input NAND gate in 45 nm technology in Table 1. It can be observed that a NAND gate conducts an almost three times larger current in a stable state for the most leaking input combination (1,1) than for the least leaky one (0,1). Such a difference can indeed be significant enough, especially when accumulating over multiple gates, to be exploited by adversaries to break cryptographic implementations. The practicality of such attacks has been demonstrated multiple times in literature [Mor14, PSKM15, MMR17, BCS+17, Moo19, MMR20].

2.3 A Toy Example

In order to emphasize the broad differences between the nature of static and dynamic power information leakage we have constructed a toy encryption circuit and evaluate its behavior for two exemplary input transitions. The circuit is depicted in Figure 1. It receives two plaintext inputs p_0, p_1, two key inputs k_0, k_1 and calculates two ciphertext outputs c_0, c_1. The numbers denoted inside each logic gate correspond to their propagation delays in time delay units (tdu). Table 2 presents the input-dependent leakage currents for all types of logic gates in the circuit.

For the first exemplary input transition, we assume that input vector $(p_0, k_0, p_1, k_1) = (0, 1, 0, 0)$ has fully propagated through the circuit and resulted in output $(c_0, c_1) = (0, 1)$. In that case, the circuit idles in the state that is illustrated at the top of Figure 2, where the green color corresponds to a signal value of '1' and the red color corresponds to '0'. As a

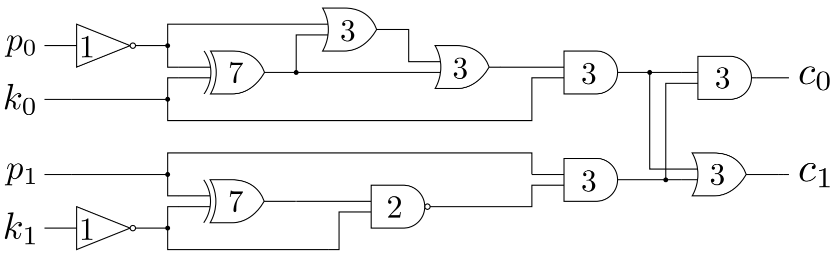


Figure 1: Toy encryption circuit with two plaintext inputs p_0, p_1, two key inputs k_0, k_1 and two ciphertext outputs c_0, c_1. The numbers within the logic gates denote their propagation delay in time delay units (tdu).

Table 2: Fictional input-dependent leakage currents of different logic gates.

Input	$\mathbf{I_{\text{INV}}}$ **[nA]**	$\mathbf{I_{\text{XOR}}}$ **[nA]**	$\mathbf{I_{\text{OR}}}$ **[nA]**	$\mathbf{I_{\text{NAND}}}$ **[nA]**	$\mathbf{I_{\text{AND}}}$ **[nA]**
0,0	33.5	278.1	156.2	54.6	98.5
0,1	39.8	239.2	155.8	41.5	81.1
1,0	-	239.5	87.9	75.3	112.6
1,1	-	287.8	68.3	112.1	143.4

next step, we assume that another input vector, namely $(p_0, k_0, p_1, k_1) = (1, 1, 0, 0)$, arrives at the input and propagates through the circuit which still holds the state corresponding to the previous input. Only one bit, namely p_0, makes a transition, while the remaining values are identical in both of the consecutive input vectors. This single-bit transition causes the sequence of gate toggles that is depicted by the timeline at the bottom of Figure 2. The timeline can be interpreted as a depiction of the dynamic power consumption caused by gate toggles. For simplicity it does not distinguish between $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions. As listed in Table 3 the input transition causes a total of 10 gate toggles, whereby 8 of them are useless, i.e., 4 glitches occur. The time to propagate the correct values to the output and keep them stable is 17 time delay units.

Unlike its dynamic counterpart, the static power consumption is not caused or affected by a transition between states. In fact, static power is consumed whenever the circuit's gates are connected to a power supply. Its magnitude, however, depends on the logic values applied to the inputs of such powered logic gates. Hence, two different stable leakage values can be observed, one before and one after the transition. These leakage values are calculated in Table 4.

While the dynamic power consumption is significant for this first input transition, due to the amount of gate toggles caused, the static power consumption shows only a small difference between both states as most of the transitions were caused by glitches and reverted back to their old state.

For the second exemplary input transition, we now assume that input vector $(p_0, k_0, p_1, k_1) = (1, 1, 1, 0)$ has fully propagated through the circuit and resulted in output $(c_0, c_1) = (1, 1)$. This state is depicted at the top of Figure 3.

Similar to the previous example, we now propagate a second input vector $(p_0, k_0, p_1, k_1) = (1, 1, 0, 0)$, which in fact is the same as before, through the circuit which still holds the previous state. As before, only one input bit makes a transition, but this time it is p_1. This

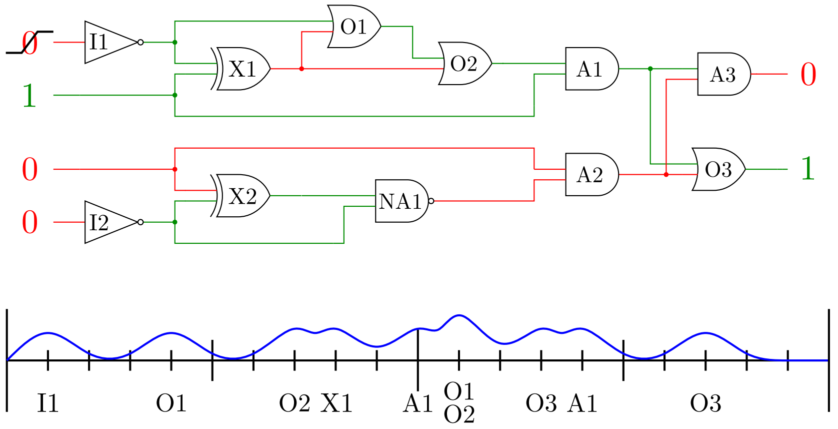


Figure 2: Circuit behavior for exemplary input transition $(p_0, k_0, p_1, k_1) = (0, 1, 0, 0) \rightarrow (1, 1, 0, 0)$. The timeline shows the occurrence of gate toggles over time.

Table 3: Number of gate toggles and glitches caused by the exemplary input transition and the total propagation time.

Input Trans. (p_0, k_0, p_1, k_1)	**Toggles**	**Glitches**	**Prop. time [tdu]**
$(0 \rightarrow 1, 1, 0, 0)$	10	4 (8 Trans.)	17

Table 4: Leakage currents exhibited by all circuit gates for the stable inputs before and after the input transition.

Input	$\mathbf{I_{I1}}$ **[nA]**	$\mathbf{I_{I2}}$ **[nA]**	$\mathbf{I_{X1}}$ **[nA]**	$\mathbf{I_{X2}}$ **[nA]**	$\mathbf{I_{O1}}$ **[nA]**	$\mathbf{I_{O2}}$ **[nA]**	$\mathbf{I_{O3}}$ **[nA]**	$\mathbf{I_{A1}}$ **[nA]**	$\mathbf{I_{A2}}$ **[nA]**	$\mathbf{I_{A3}}$ **[nA]**	$\mathbf{I_{NA1}}$ **[nA]**	$\mathbf{I_{SUM}}$ **[nA]**
(0, 1, 0, 0)	33.5	33.5	287.8	239.2	87.9	87.9	87.9	143.4	98.5	112.6	112.1	1324.3
(1, 1, 0, 0)	39.8	33.5	239.2	239.2	155.8	68.3	87.9	143.4	98.5	112.6	112.1	1330.3

toggle causes the sequence of gate transitions that is depicted at the bottom of Figure 3. The interesting observation is here that, although the same input is propagated through the same circuit, the behavior and dissipation of the circuit is vastly different in this second example. In particular, as listed in Table 5, only 4 gate toggles and 0 glitches occur. Furthermore, this second input transition leads to an execution time of only 6 *tdu* until the correct result is stable at the output (although further toggles occur in intermediate gates until 9 *tdu*). For the previous exemplary input transition, the output gates' last toggle occurred after 17 *tdu*.

This significant difference in the number of toggles and the execution time clearly highlights that the currently encrypted plaintext is only one part of the leakage function and can barely be correlated to the dynamic power dissipation of the circuit without knowledge of the initial state. Also, it underlines the fact that gates at similar stages of the circuit (e.g. directly at the output) are commonly evaluated at completely different moments in time depending on the transition at the input.

Finally, the static power consumption of the circuit before and after the transition is calculated in Table 6. In this scenario, the difference in the static power consumption is much larger than before, since more useful transitions occurred and affected the final state of the circuit. However, due to the lack of glitches the dynamic power consumption is much lower than for the first exemplary transition. In summary, the dynamic power

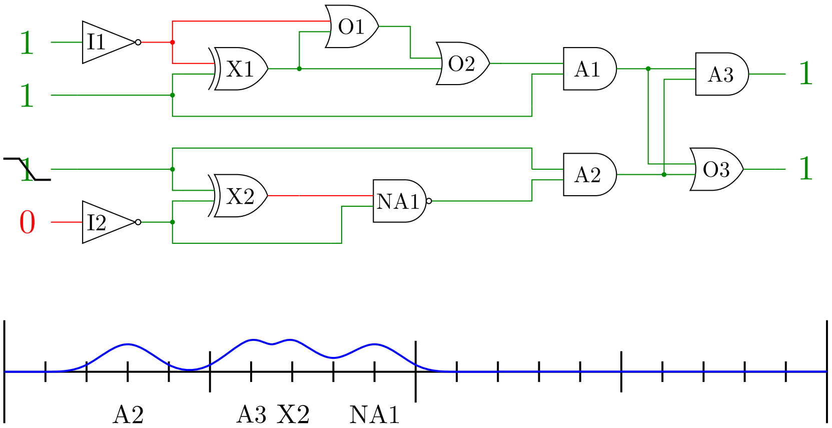


Figure 3: Circuit behavior for exemplary input transition $(p_0, k_0, p_1, k_1) = (1, 1, 1, 0) \rightarrow (1, 1, 0, 0)$. The timeline shows the occurrence of gate toggles over time.

Table 5: Number of gate toggles and glitches caused by the exemplary input transition and the total propagation time.

Input Trans. (p_0, k_0, p_1, k_1)	**Toggles**	**Glitches**	**Prop. time [tdu]**
$(1, 1, 1 \rightarrow 0, 0)$	4	0 (0 Trans.)	6

Table 6: Leakage currents exhibited by all circuit gates for the stable inputs before and after the input transition.

Input	I_{I1} **[nA]**	I_{I2} **[nA]**	I_{X1} **[nA]**	I_{X2} **[nA]**	I_{O1} **[nA]**	I_{O2} **[nA]**	I_{O3} **[nA]**	I_{A1} **[nA]**	I_{A2} **[nA]**	I_{A3} **[nA]**	I_{NA1} **[nA]**	I_{SUM} **[nA]**
(1, 1, 1, 0)	39.8	33.5	239.2	287.8	155.8	68.3	68.3	143.4	143.4	143.4	41.5	1364.4
(1, 1, 0, 0)	39.8	33.5	239.2	239.2	155.8	68.3	87.9	143.4	98.5	112.6	112.1	1330.3

consumption of an unrolled cryptographic primitive, like our toy cipher, is to a large part determined by the state of the combinatorial circuit before the actual input arrives. The following sections highlight how this situation can be exploited by designers to increase the side-channel security of unrolled circuits. The static power consumption, however, is not affected by any previous state and therefore inherently immune against countermeasures based on this fact.

3 Target

The target for our practical evaluations is a 40 nm CMOS ASIC prototype which is depicted in Figure 4. The chip is fully digital and 1.92 mm × 1.92 mm in size. It contains 8 metal layers for routing and has a nominal core voltage of 1.1V, as well as a nominal IO voltage of 2.5V. The ASIC has been developed for physical security evaluations and contains a number of different cipher cores which are all clock gated in order to make sure that they do not influence each other during evaluations. The position of the unrolled PRINCE implementation is indicated by the red circle on the left side of Figure 4. It occupies an area of 10 036 gate equivalents (GE), which corresponds to an on-chip-size of about 97.2 µm × 97.2 µm considering the utilization of about 75%. We have performed fully SDF annotated post-layout gate level simulations of the PRINCE core with picosecond accuracy at typical conditions. As stated in Section 2 already, the vast majority of gate toggles during the execution of the PRINCE core can be attributed to glitches. When both, the plaintext and the key are changed from one random value to another, the 9 169 logic gates in the circuit perform on average 114803 output transitions, while 96% of that are glitches. When only the plaintext is exchanged, 56920 gate transitions are caused, 92% of which correspond to glitches.

Our timing simulations further allow us to visualize the differences in execution time, defined as the time until the last output gate toggle occurs, for different input transitions and scenarios. Obviously such differences can not easily be observed in the power measurements presented in Section 4. Figure 5 shows the results of our simulations. We consider 2 different cases here and provide distributions for the occurrence of the first and the last output toggle respectively. We focus on the red distributions first. Here, we have simulated the unrolled PRINCE core for random plaintext transitions under a fixed key. In this scenario, the first output gates toggle after 2.1 to 2.4 nanoseconds. The final output toggle, before the output is stable and can be read from the crypto core, occurs after 4.2 to 4.5 nanoseconds. This already highlights the fact that gates corresponding to the last round of the cipher are evaluated at vastly different points in time, depending on the transition at

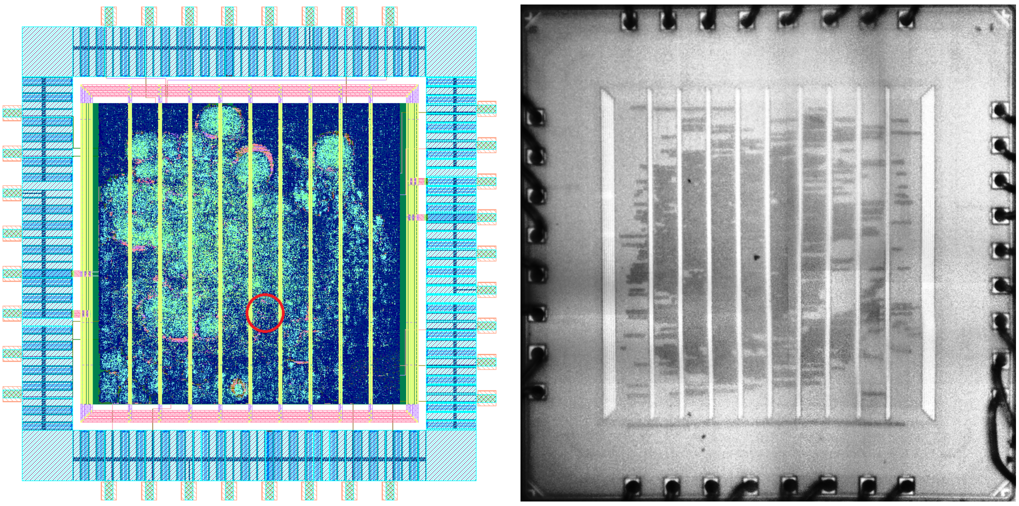


Figure 4: Layout schematic of the 40 nm ASIC on the left and microscope photography of the fabricated chip on the right.

the input. In that regard, it is important to note that differences of up to 2 nanoseconds can also be observed when monitoring the last toggle of a single output gate for different input transitions. Hence, dynamic power traces of unrolled primitives are inherently misaligned (asynchronous) at later stages of the circuit. It is also noteworthy that the quickest paths to the output require only half the amount of time that the longest ones require. The blue distributions in Figure 5 have been acquired when both, the plaintext *and* the key, are subject to random transitions. In that case the first toggles occur as early as 0.6 to 1.0 nanoseconds after the start. This is caused by the fact that the round key is propagated to all rounds at the same time and the path from the last round to the output is obviously shorter than the path from plaintext to output. The last toggle occurs later than for a fixed key due to the larger number of glitches that are caused when changing the key as well.

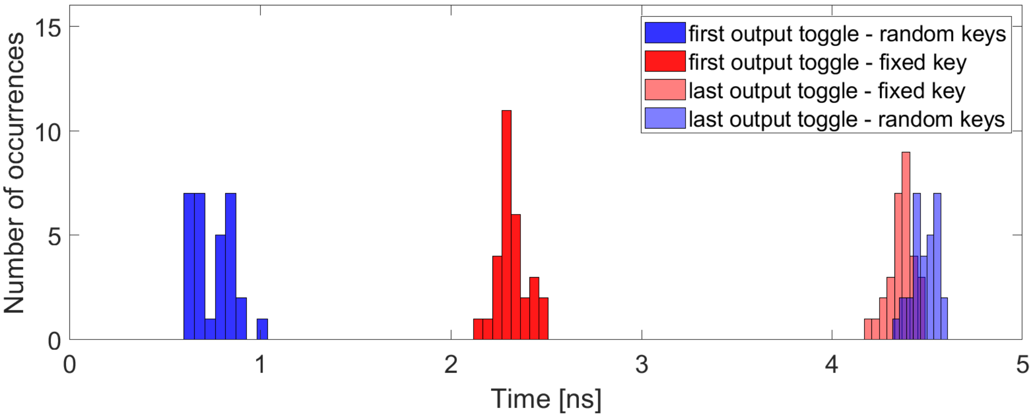


Figure 5: Timeline showing the distribution of first and last output toggles when simulating the unrolled PRINCE netlist for random plaintext transitions and either a fixed key (red) or random key transitions (blue).

4 Experimental Results

In this section we present our practical analysis of the physical security level of unrolled PRINCE implemented on a 40 nm ASIC prototype. At first, we analyze the dynamic dissipation of the circuit in 5 different usage scenarios. Then, we compare those scenarios against each other in terms of security provided and overhead spent. Finally, we present attack results exploiting the static power consumption of the circuit and discuss the dangers of ignoring this security threat.

4.1 Dynamic Power Attacks

Previous analyses of the SCA security provided by unrolled PRINCE implementations have targeted the dynamic power consumption on FPGA platforms. As described in Section 1, an ASIC implementation is expected to have a much higher speed of execution, a higher asynchronicity as well as lower power consumption footprint. All of those differences should increase the difficulty to perform attacks on the primitive, especially in an advanced semiconductor technology node due to the even smaller delays and even lower power consumption per logic unit. In the following experiments, we analyze the level of difficulty to mount successful attacks on our unrolled circuit, while taking different usage principles and adversary models into account. We distinguish a number of cases, as our conclusions about the security level of unrolled PRINCE on silicon entirely depend on how the primitive is instantiated and used.

4.1.1 Measurement Details.

In order to provide a meaningful and fair analysis, we have taken several measures to guarantee the highest possible quality of results in our trace acquisition. First of all, in order to capture the dissipation of a primitive whose execution takes only a couple of nanoseconds (see Figure 5), a high bandwidth and a high sampling rate are required from the oscilloscope that is utilized in the measurement process. In that regard we chose a *Teledyne LeCroy WaveRunner 8254M* [Wav20], which features a bandwidth of 2.5 GHz and a sampling rate of 40 GS/s. The vertical resolution of the scope is 8 bit in normal operation and up to 11 bit with enhanced resolution (ERES) which we used in our measurements. As a next step, we evaluated whether electromagnetic emanation (EM) or power measurements were favorable for the analysis. We found that power measurements led to a higher signal-to-noise ratio (SNR) than EM measurements which were recorded on the front side of the chip directly above the PRINCE core using a *Langer EMV ICR HH150-27* near-field probe with a bandwidth of up to 6 GHz. It may be counterintuitive that the power consumption of our target is supposed to be more informative than the electromagnetic radiation. Typically, when sampling a very short signal which potentially carries a lot of temporal information, EM measurements are the method of choice. Indeed, the precise point in time when a certain glitch occurs may carry valuable information for an adversary. Yet, we could not observe any benefit on our target when performing EM measurements and will shortly discuss the potential reasons for that. First of all, many logic gates in the underlying CMOS node have propagation delays as short as 10-20 picoseconds. Thus, one is naturally limited by the probe's and oscilloscope's bandwidth and sampling rate to adequately capture the timing of individual intermediate transitions. Bandwidths of 50 GHz and beyond would be required to enable the proper sampling of fast glitches in the output lines of such logic gates and common lab equipment rarely supports such high frequencies[4].

Additionally, we know from the simulations presented in Section 3 that up to 115 000

---

[4] Oscilloscopes with a bandwidth above 50 GHz do exist, e.g., the Teledyne LeCroy LabMaster 10 Zi-A series [Lab20], but come at the cost of multiple hundreds of thousands of dollars.

gate transitions occur in a span of less than 5 nanoseconds when executing the targeted PRINCE core. Thus, at each point of the execution a large number of gate transitions occurs simultaneously, making it difficult to effectively sense minor temporal differences related to the processing of a small part of the circuit. Finally, the active transistor area is covered by 8 metal layers (as seen from the front side of the die) which are densely utilized. While not all routed signals above the targeted area will transport information at the time of measurement, some certainly do. Others are responsible for supplying current to the core area of the chip. Hence, the same voltage fluctuations that are seen in the power measurements will also affect any EM measurements from the front side. In fact, from a purely visual standpoint EM measurements of this target's execution have a similar appearance as power traces and do not show any sharper distinction of different calculation parts. Measurements from the back side, targeting a thinned sample of the ASIC, may potentially improve the quality of results, but we consider this out of scope for this work. Hence, we concentrate on power analysis attacks in the following.

Since the static power results, presented later in this section, were acquired inside a climate chamber at an increased temperature and supply voltage, we examined whether those parameters have an impact on the dynamic power results as well. Our observations are that neither a constant nor a lower or higher temperature led to noticeably improved results, in terms of attack success and SNR. Thus, we recorded all dynamic power traces outside the chamber at room temperature. Increasing the supply voltage from 1.1V to 1.6V (45.45% overvoltage) on the other hand raised the first-round SNR by about 10-15%. Hence, we chose to exploit this effect in our trace acquisition. For completeness, we also evaluated a decreased supply voltage of 0.8V (27.27% undervoltage). A smaller supply voltage in theory leads to higher propagation delays and an overall slower execution of the primitive which may result in a better sampling of the signal. However, the results did not confirm any potential improvement for lower voltages, probably due to the generally decreased dissipation, but showed a reduced SNR by about 7% and a negative effect on the attack success.

4.1.2 No Reset (Highest Performance, Lowest SCA Security).

The most straightforward manner to implement unrolled PRINCE and similar cryptographic primitives does not include any data path reset or key-removal mechanism. Each plaintext or ciphertext input is processed as soon as it arrives and no cleanup of any kind is performed after the cryptographic operation. This is the desired mode of operation when high performance is the primary objective, since it guarantees that each PRINCE instance has a throughput of 64 bits per clock cycle and delivers the result of a new encryption after exactly one clock cycle. However, these deliverables demand that a key is constantly applied to the circuit, unless it is somehow possible to fetch it from a secure storage without any additional latency. Furthermore, it implies that the circuit is not reset to a determined state after an encryption, causing the final state of the previous encryption to remain in the cryptographic circuit until a new plaintext arrives. Hence, each new plaintext causes the circuit to transition from the previous input to the new one, and the dissipation depends on both of those values in the same manner. While an implementation of that kind delivers the overall highest performance, it is the weakest in terms of SCA protection.

In order to demonstrate this, we have measured 500 000 traces for random plaintexts processed by the unrolled PRINCE encryption on the 40 nm ASIC which took less than 15 minutes to acquire. Figure 6 depicts an overlay of 30 sample measurements on the top and the result of a Correlation Power Analysis (CPA) [BCO04] attack on the bottom. The leakage model for the CPA is the Hamming distance (HD) between two first-round Sbox outputs based on consecutively encrypted plaintexts. To be more precise, the model can be expressed as $\mathrm{HD}(\mathrm{S}(p_{i-1,j} \oplus \hat{k}_j), \mathrm{S}(p_{i,j} \oplus \hat{k}_j))$, where $\mathrm{HD}(\cdot)$ is a function calculating the

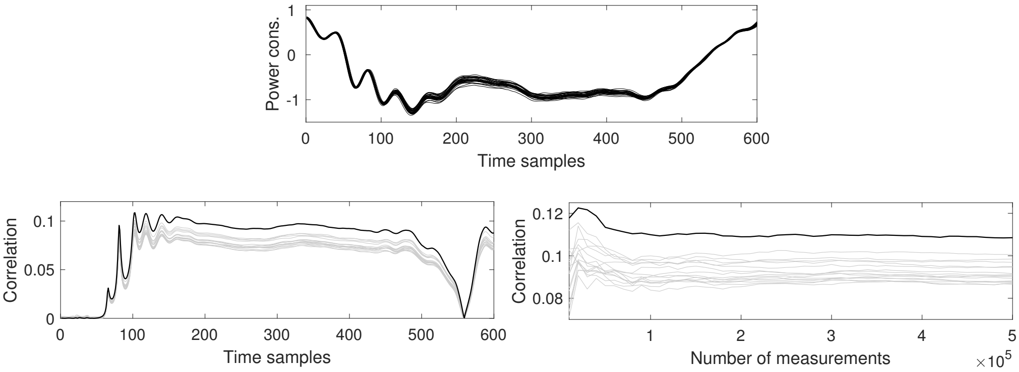


Figure 6: Overlay of 30 sample traces and a CPA attack using the Hamming distance of two first-round Sbox outputs based on consecutive known plaintexts.

Hamming distance, S(·) is the substitution box (Sbox) of the PRINCE cipher and $p_{i,j}$ is the j-th plaintext nibble of the i-th plaintext that is encrypted. The term $\hat{k}_j$ corresponds to the j-th nibble of $\hat{k}$, with $\hat{k}$ being defined as $\hat{k} = k_0 \oplus k_1$. The whitening key k_0 and the round key k_1 are defined in [BCG+12]. This model requires knowledge of consecutively encrypted plaintexts under the same key. When an adversary possesses such knowledge, all key nibbles can be recovered with the available amount of traces. To be more precise, in our experiments, the lowest number of traces required to recover a key nibble was 3 000, the median was 43 000 and the highest number was 350 000. This is already a significant amount of samples required to perform a key recovery on an essentially unprotected implementation. Obviously there are multiple reasons for this observation, including the small power consumption footprint and high speed of the 40 nm ASIC technology, the highly-parallel implementation style and the asynchronicity of the signals. As we will see in the following, the exploitation effort is even much higher without knowledge of consecutively encrypted plaintexts under the same key.

Please note, that we do not consider chosen-input scenarios in this work. It is obvious and has been demonstrated in [YHA17a], that such a scenario allows to increase the signal-to-noise ratio (SNR), especially in the first round(s), significantly by toggling only the targeted input bits instead of encrypting random plaintexts. We also do not consider dedicated filtering and pre-processing approaches (as e.g. presented in [CSR+19]) or template attacks in this work. Instead, we concentrate on more generic evaluation metrics in this work in order to keep the analysis as universally valid as possible and make as few assumptions about an attackers capabilities as possible. We are aware that for any concrete implementation a highly specialized and optimized attack method will usually outperform such generic approaches. Yet, this is a valid statement for all attacks (dynamic and static) presented in this work and should not significantly impact the interpretation of our comparison. Also, we would like to stress that we did not perform a TVLA analysis [GJJR11, SM15] on the measurements in the *no reset* scenario, as this is not straightforward. Without a reset between encryptions, the dissipation depends not only on the currently encrypted plaintext, but also on the previous one. Since the measurements for fixed and random plaintexts in a non-specific t-test should be recorded in a randomly interleaved fashion [SM15], the following 4 transition groups would occur, (i) fix → fix, (ii) fix → ran, (iii) ran → fix, (iv) ran → ran. Hence, 4 different distributions would need to be distinguished instead of the usual 2 and the transition from fixed to fixed would always have zero dissipation, as no gate toggles are caused. Comparing only the two groups (iii) ran → fix and (iv) ran → ran is no solution for this problem either, as it would lead to a false sense of security. The random component in the (iii) ran → fix transition group is artificially introduced by the methodology and would not be a limiting factor for the adversary in an actual attack. For this reason, we refrain from defining a new TVLA methodology for this special case. However, for all experiments presented in the following we are able to provide TVLA results due to the reset methods applied.

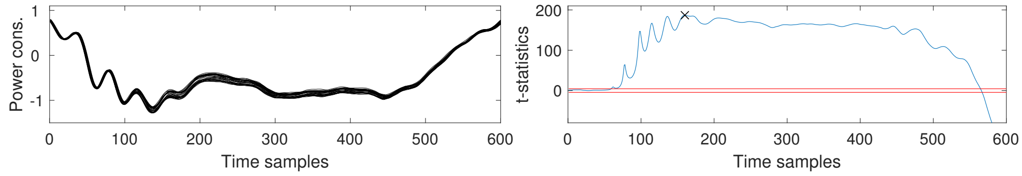
4.1.3 Reset Methods (Lower Performance, Higher SCA Security).

As our example on the toy cipher in Section 2 has demonstrated, the dynamic power consumption of an unrolled cipher during an encryption substantially depends on the initial state of the circuit. In particular, the leakage function of the circuit is mainly transitional. An attacker who obtains knowledge of consecutive inputs (processed under a fixed key) can easily correlate the dissipation of the circuit to his transitional hypothesis under the correct key guess. However, the authors of [BGSD10] claimed that *clearing* the data path between encryptions successfully prevents an adversary from performing such an attack. It is unclear from [BGSD10] how exactly such a reset (i.e., clearing) procedure should be executed, although it is described as *"propagating random values without interference from the key"*. In our experiments we test 4 different methods which come at different costs in terms of required randomness per cycle, total power consumption and delay. Yet, they all halve the available throughput, since useful data can only be encrypted every other clock cycle. Hence, in order to achieve the same performance as an unrolled primitive without reset method applied, twice as many unrolled PRINCE instances need to be implemented. This overhead is not unacceptable when compared to other side-channel countermeasures, such as dual-rail logic or masking. The 4 different reset strategies that we evaluate in the following are:

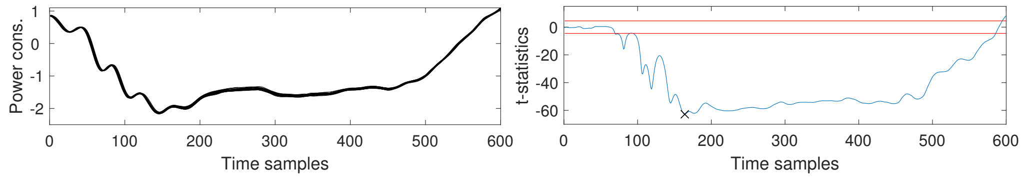
1. resetting the plaintext to zero
2. resetting the plaintext and the key to zero
3. resetting the plaintext to a random value
4. resetting the plaintext and the key to a random value

As described before, we are able to provide TVLA results in all 4 of these scenarios. Figure 7 shows an overlay of 30 sample traces on the left and results of a non-specific t-test for 10 000 randomly interleaved measurements for fixed and random inputs on the right. It can be observed that the voltage drop which is measured in the scenarios where both, the key and the plaintext, are reset is significantly larger. Figure 8 depicts the histograms of the fixed and random groups at the most leaking time sample (marked by an 'x' in Figure 7). It also shows the evolution of the maximum t-value over the number of traces on the right side for all 4 scenarios. Although the t-test reports leakage with a high confidence in all 4 scenarios, clear differences in the statistic's magnitude can be observed in Figure 7. Yet, the absolute magnitude of the t-value in a leakage detection scenario only expresses the confidence that the two input distributions can be distinguished (by their means in case of a first-order t-test) and can not be used as an assessment of the security level of an implementation. However, the histograms and the number of traces required to reach a certain t-value, as shown in Figure 8, confirm that the concrete fixed and random distributions shown here are more difficult to distinguish in case of the random reset scenarios. In order to asses the actual security level provided by the different usage scenarios we perform key recovery attacks in the following.

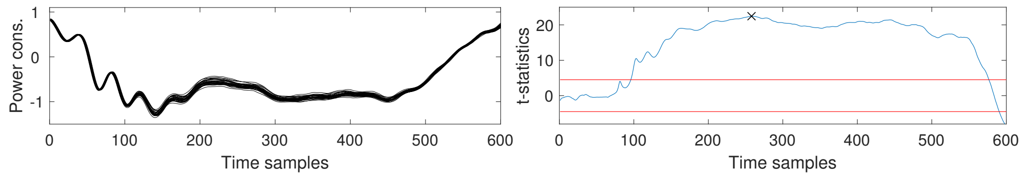
We chose to apply two different classes of key recovery attacks here, first Correlation Power Analysis (CPA) based on a power model and second collision-based SCA attacks which are independent of leakage models. As power models for the CPA we have tested the Hamming weight of Sbox outputs, the Hamming distance between Sbox outputs (of two consecutive encryptions) and all corresponding single-bit models (transition- and value-based). As



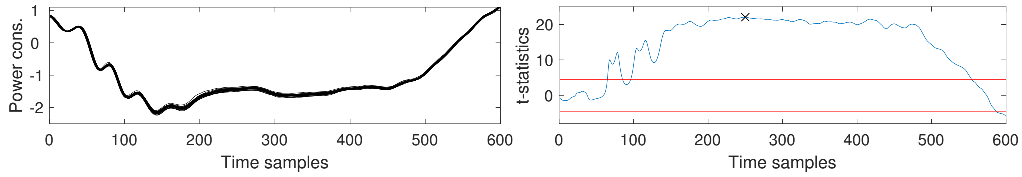
(a) plaintext reset to zero



(b) plaintext and key reset to zero



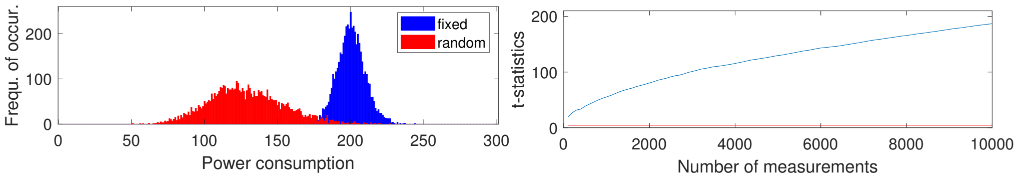
(c) plaintext reset to random



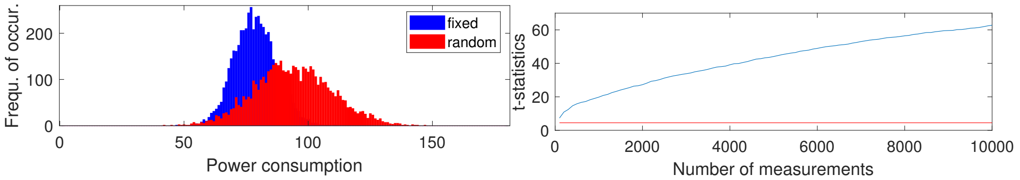
(d) plaintext and key reset to random

Figure 7: Overlay of 30 sample traces and t-test results using 10 000 traces each for 4 different reset methods.

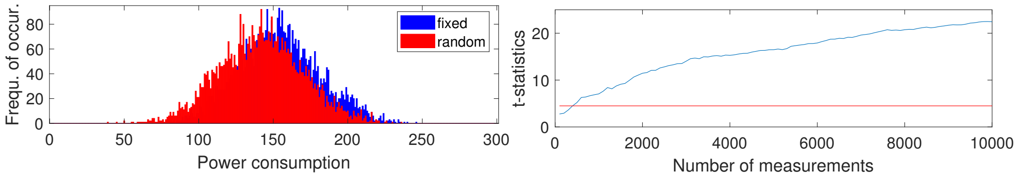
collision-based SCA attacks we have evaluated the leakage-model-independent approaches introduced as Correlation-Enhanced Collision in [MME10] and Moments-Correlating DPA (MCDPA) in [MS16b]. Table 7 states the most successful of our tested attacks for all 5 scenarios (including *no reset*). The collision-based SCA attacks recovered less information about the key than simple CPA. This is reasonable, as collision-based SCA attacks are based on the assumption that a module is (time-)shared between multiple computations, or, in the parallel case, that multiple physical instances of a module have similar leakage characteristics. In our standard-cell-based unrolled PRINCE circuit, all Sboxes are realized as a unique composition of gates and therefore are expected to have different power characteristics and time of evaluation depending on the input transition. Thus, the circuit does not meet the requirement to successfully apply collision-based SCA, at least not to recover significant portions of the key. This is already a major difference compared to FPGA-based results, where collision-based attacks were shown to be effective [MMP11]. According to Table 7, the *plaintext reset to zero* already provides an increased security level compared to the *no reset* scenario. The leakage function depends on fewer variable inputs that are predictable for the adversary, which makes the attack less powerful. To be more precise, the power model reported in Table 7 for the *no reset* scenario depends on $p_{i-1,j}$ and $p_{i,j}$, while for the *plaintext reset to zero* case $p_{i-1,j}$ is replaced by constant 0. As there are 16 possible values each for $p_{i-1,j}$ and $p_{i,j}$, the adversary can distinguish



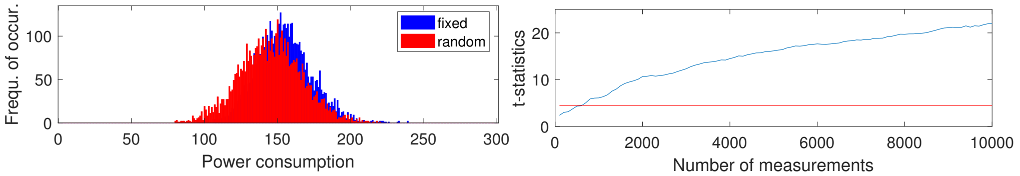
(a) plaintext reset to zero



(b) plaintext and key reset to zero



(c) plaintext reset to random



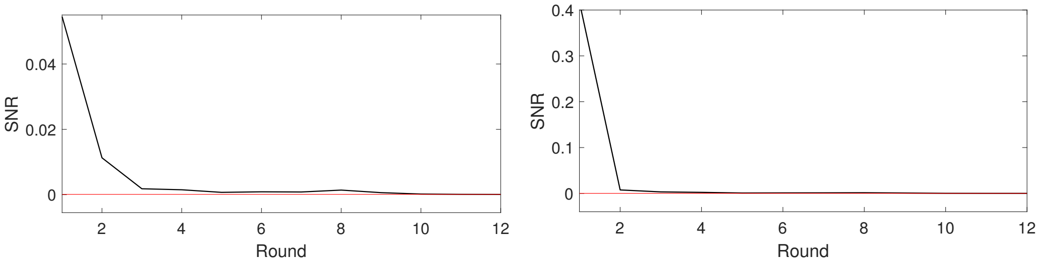
(d) plaintext and key reset to random

Figure 8: Histograms of the fixed and random groups at the most leaking time sample and development of the maximum t-value over the number of traces for 4 different reset methods.

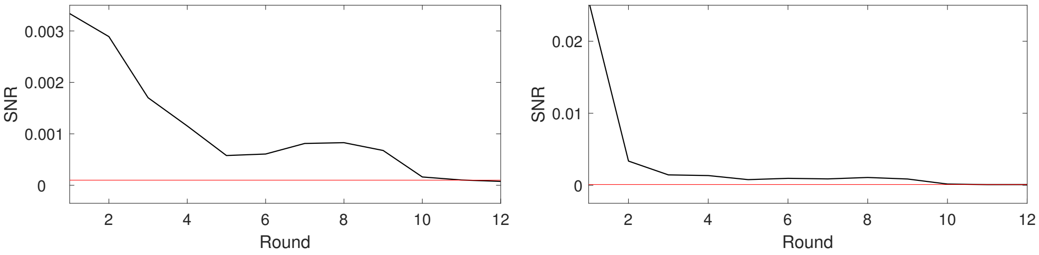
$16 \cdot 16 = 256$ cases in the *no reset*, but only 16 in the *plaintext reset to zero* scenario. This is sufficient to reduce the number of recovered nibbles by more than the half. However, the encryption is still fully deterministic, as for any other common unprotected block cipher implementation.

The method where both, the key and the plaintext, are reset to zero apparently provides further security. This can be explained by the noise that is induced due to the fact that all rounds receive the round key at the same time at the start of the encryption. Hence, all rounds begin to toggle before the new plaintext even propagated that far (see Section 3). When the random resets are used, the encryption is non-deterministic, which results in a much higher noise level, as reflected in the t-test and attack results. Also, the adversary can not easily make transitional hypotheses anymore. In these cases, a significant part of the leakage function is unknown to the adversary. It still has some value-based leakage component, thus the detectable first-order leakage, but the evaluation shows that most key nibbles in the first round can not be recovered.

For a final comparison between the 4 different reset methods we evaluated the maximum signal-to-noise ratio (SNR) based on the 16 input nibbles of each of the 12 block cipher rounds. The results are depicted in Figure 9. The first observation that has to be made is that the SNR degrades very quickly after the first round for all 4 methods. This is caused



(a) plaintext reset to zero (b) plaintext and key reset to zero



(c) plaintext reset to random (d) plaintext and key reset to random

Figure 9: Maximum (nibble wise) signal-to-noise ratio (SNR) computed for all 12 round inputs for 4 different reset methods.

by the nature of unrolling. As already demonstrated by the toy example in Section 2 and the simulation results in Section 3, logic gates in later rounds are evaluated at completely different moments in time for different input transitions. The signals arrive at a different time at the gates depending on which path they have taken. This is what we call the asynchronicity of signals. Additionally, since the gates corresponding to the last two rounds are located towards the end of the circuit, their computational result does not affect as many further gates as the output of earlier rounds. The red line in Figure 9 is the border for statistical insignificance, which we experimentally determined as 0.0001. In all 4 cases the maximum SNR for round inputs 11 and 12 is below this threshold, indicating that an attack on the last two rounds is not expected to succeed given the available amount of traces. Nevertheless, we attempted different attacks from the ciphertext side, but indeed none were successful. Another interesting observation that can be made in Figure 9 is that the methods where the key is reset as well have a significantly higher first-round SNR. As reported in Section 3, the amount of gate toggles caused when changing the key in addition to the plaintext is more than twice as large. This can also be observed in the larger voltage drop in Figure 7. Taking all evaluation metrics into account (TVLA, CPA, SNR) we come to the conclusion that the *random plaintext reset* is the most preferable choice. It delivers the best SCA security and is cheaper than the *random key and plaintext*

Table 7: Summary of the optimal attack results (among all tested ones) for the 5 dynamic power scenarios respectively with a maximum of 500 000 traces.

Reset Type	**Attack**	**Best Power Model Found**	**Rec. Nib.**
no reset	CPA	$\mathrm{HD}(\mathrm{S}(p_{i-1,j} \oplus \hat{k}_j),\, \mathrm{S}(p_{i,j} \oplus \hat{k}_j))$	16/16
plain zero	CPA	$\mathrm{HD}(\mathrm{S}(0 \oplus \hat{k}_j),\, \mathrm{S}(p_{i,j} \oplus \hat{k}_j))$	7/16
plain and key zero	CPA	$\mathrm{HD}(\mathrm{S}(0 \oplus 0),\, \mathrm{S}(p_{i,j} \oplus \hat{k}_j))$	5/16
plain random	CPA	$\mathrm{HW}(\mathrm{S}(p_{i,j} \oplus \hat{k}_j))$	2/16
plain and key random	CPA	$\mathrm{HW}(\mathrm{S}(p_{i,j} \oplus \hat{k}_j))$	3/16

reset, since it consumes a less power and requires only a third of the randomness per clock cycle (32 instead of 96 bit).

In order to provide a benchmark with respect to the security level that the *random plaintext reset* scenario provides we have measured 100 million traces which took about 24 hours. Then we conducted the same CPA attack that proved to be most successful for the smaller amount of traces (see Table 7) on all key nibbles and obtained the results depicted in the Appendix in Figure 13. The first row shows the results targeting key nibbles 0 to 4, while the last row corresponds to nibbles 12 to 15. Surprisingly, the attack with 100 million traces is not any more successful than with 500 000 traces, as also 2 key nibbles can be recovered (numbers 1 and 15)[5]. We also attempted single-bit models, but none of them succeeded on more than two nibbles either. For the sake of completeness we finally performed collision-based Moments-Correlating DPA (MCDPA) [MS16b] on the 100 million traces with an offset of 0. The results are depicted in the Appendix in Figure 14. Here, the first row shows the results targeting key differences 0-1 to 3-4, while the last row corresponds to differences 12-13 to 15-0. As shown in the figure, only 1 key difference can correctly be recovered[6].

In conclusion, the *random plaintext reset* is a viable and effective protection against side-channel attacks targeting the dynamic circuit emanations of unrolled primitives. In our case study it was not possible to extract a notable portion of the key even with a huge amount of available traces. Furthermore, even if a larger part of the key could be extracted from the first round, an attacker would still need to perform further attacks with a deeper hypothesis into the second round or target the last round. Both strategies have small likelihood of success based on the acquired SNR results. This analysis shows that unrolled cryptography on silicon can provide a high level of resistance against dynamic power SCA attacks at a low cost, if certain usage principles are carefully respected.

4.2 Static Power Attacks

As a next step, we analyze the susceptibility of unrolled ciphers in state-of-the-art ASIC technology towards attacks exploiting the static power consumption. Static power side-channel analysis (SPSCA) has been a growing field in recent years due to its emergence in nanometer-scaled CMOS technologies [Moo19]. Its nature is entirely different from dynamic power analysis, since it does not exploit a momentary transitional effect that can be observed for a finite period of time only. Instead, it is based on observing a static phenomenon that can be quantified for as long as no transition occurs in the targeted circuit part. As demonstrated on the toy example in Section 2, the static power consumption is fully independent of a potential previous state of the circuit and exhibits a deterministic leakage behavior for any given input. Hence, the aforementioned tricks and usage principles are not expected to be effective against this kind of adversary. In the end of our analysis, we discuss under what circumstances one source of information leakage is preferable over the other (from an adversarial standpoint) and which guidelines need to be observed to provide protection against both.

4.2.1 Measurement Details.

Our setup for the static power side-channel attacks differs from the one used for the dynamic power experiments in several regards. Most notably, we have used a different oscilloscope. Since a high bandwidth and sampling rate are not primarily important for static power measurements, we rather chose a scope that has a high vertical resolution.

---

[5] Of course, it remains unclear whether these 2 successful recoveries are indeed reliable or simply a consequence of the probability of 1/16 for each key candidate to produce the highest correlation when no significant correlation is found for any candidate.

[6] As for the CPA, this may well be a statistical incident and no true recovery.

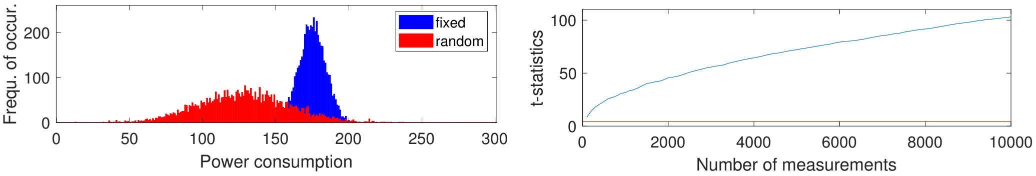


Figure 10: Histograms of the fixed and random groups and development of the maximum t-value over the number of traces for the static power measurements.

In particular, we have used a *Teledyne LeCroy HRO 66Zi* [HRO20], which features a bandwidth of 600 MHz, a sampling rate of 2 GS/s and a vertical resolution of 12 bit in normal operation and up to 15 bit with enhanced resolution (ERES). Furthermore, due to the leakage-enhancing effects of higher temperatures reported in [Moo19, MMR20], we have performed the measurements in a climate chamber at 90°C. The core supply voltage has been set to 1.6V instead of the nominal 1.1V (45.45% overvoltage), just as for the dynamic power measurements. Additionally, our setup utilizes a DC amplifier and a low-pass filter, similar to what has been suggested in [MMR20].

4.2.2 Results.

In the following, we apply the same evaluation metrics that have been introduced for the dynamic power analysis. First of all, we have measured the static power consumption of the circuit for a randomly interleaved sequence of 10 000 fixed and random inputs in order to perform a non-specific t-test. The results are shown as a histogram on the left and as t-value over the number traces on the right side in Figure 10. It is important to note that the measurements have been recorded while the global clock of the ASIC was active and other unrelated computations have been executed on the chip. In particular, an LFSR-based PRNG was running during the measurement and constantly computed on random values. We introduced this additional obstacle in order to clarify that there is not *necessarily* any relevant increase in the difficulty of measuring the static power consumption just because other parts of the circuit are consuming dynamic power at the same time[7]. This has already been observed in [Moo19] and was confirmed by our experiments. It is only required that the unrolled circuit itself is idle during the measurements and contains the sensitive intermediates. In this regard, it becomes clear that an unrolled circuit without a reset method or key-removal mechanism is susceptible to static power attacks, even if the adversary can not influence the clock signal of the circuit. Before moving to key recovery attacks, we take a look at the signal-to-noise ratio for all 12 round inputs. The result is compared to the dynamic power scenarios in Figure 11. It can be observed that the SNR does not degrade significantly after the first round, much unlike the dynamic scenarios. It stays approximately in the same range for all 12 round inputs. Furthermore, the SNR is consistently higher than in all dynamic power measurements for rounds 2-12. Even in the first round, the static power SNR is significantly larger than that for the *random plaintext reset* scenario. Clearly, the static power side-channel leakage contains more information about the later stages of the combinatorial circuit than the dynamic dissipation. In this regard we are able to perform attacks from the plaintext *and* the ciphertext side. Two exemplary results are depicted in Figure 12 and their success is compared in Table 8. Acquiring the 500 000 traces took approximately 70 hours. While the trace acquisition is orders of magnitude slower than for the dynamic power measurements in our laboratory experiments, the difference is expected to be significantly smaller when examining real

---

[7] Clearly, this does not mean that any kind of unrelated parallel workload is irrelevant for the attack success. Many counterexamples can be imagined. It simply means that it is no strict requirement that the whole chip is idle during measurement.

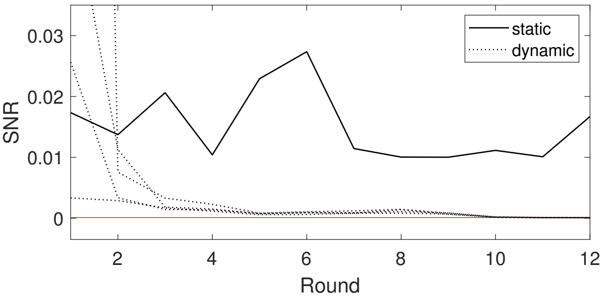
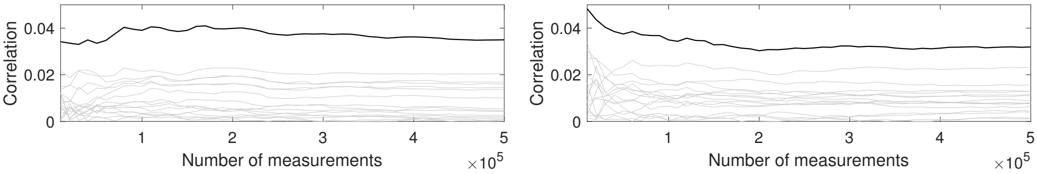


Figure 11: Maximum (nibble wise) signal-to-noise ratio (SNR) computed for all 12 round inputs for the static power measurements.



(a) first round

(b) last round

Figure 12: Two CPA attacks on the static power measurements, one on the first round of PRINCE using the LSB of the Sbox output and one on the last round of PRINCE using the LSB of the inverse Sbox input as a leakage model.

world devices. In our experience it is often not possible to acquire hundreds of millions of dynamic power traces per day when analyzing an actual product for its physical security. Additionally, we have not attempted to optimize the static power measurements in terms of measurement time in these experiments. Previous articles have outlined the trade-off between time interval spent and the quality of static power measurements [MMR20].

The CPA with ciphertext knowledge on the last round leads to the best results. The lowest number of traces required to recover a key nibble is 1 000, the median is 21 000 and the highest number is 420 000. 14 key nibbles can be recovered with less than 50 000 traces. In contrast to the dynamic power analysis, these numbers are independent of a potentially applied reset method and independent on having access to plaintexts. Hence, especially in the somewhat *protected* use cases, the static power consumption is a much more severe threat to the SCA security than the dynamic power. While a fully unprotected (i.e., *no reset*) unrolled implementation of PRINCE still provides high protection against dynamic power analysis attacks on the last rounds (with ciphertext-only knowledge), this is not true against static power adversaries, even if clock control is not an option[8]. Furthermore, in case clock control *is* obtained by a static power adversary, all reset methods are conceptually ineffective to thwart attacks. The adversary can simply stop the clock whenever a user-supplied input is applied to the combinatorial circuit (i.e., before the reset is performed) and measure its static leakage without any influence from a previous state. However, when the random reset of the circuit between each two encryptions is performed immediately in the next clock cycle after each valid encryption and the attacker can not influence the clock signal, SPSCA attacks are not informative and an adversary has to rely on the dynamic currents to extract information.

---

[8] Unless the circuit is never idle and encrypts data all the time. Then, any kind of SPSCA fails.

Table 8: Summary of the optimal attack results (among all tested ones) for the static power scenario with a maximum of 500 000 traces.

Round	Attack	Best Power Model Found	Rec. Nib.
first	CPA	$\mathrm{LSB}(\mathrm{S}(p_{i,j} \oplus \hat{k}_j))$	15/16
last	CPA	$\mathrm{LSB}(\mathrm{S}(c_{i,j} \oplus \hat{k}'_j))$	16/16

5 Conclusion

In this work we have analyzed the physical security level that can be provided by the unrolled low-latency cipher PRINCE when it is implemented in state-of-the-art semiconductor technology. We have realized the primitive in a fully round-unrolled fashion in a 40 nm CMOS node as a semi-custom standard-cell-based design with a latency of less than 5 nanoseconds. Our observations regarding its vulnerability are manifold. First of all, performing a *full* key-recovery attack (revealing all 128 key bits) on an unrolled ASIC implementation of PRINCE is always hard when observing its dynamic behavior. Even in the best case for the adversary it is difficult to extract more than 64 bit of information about the key. The extremely fast execution, the high level of parallelism and its asynchronicity make the life of adversaries difficult. Recovery of even small parts of the key requires a huge amount of observations when the adversary can not obtain the encrypted plaintexts, but rather is in possession of the ciphertexts only[9]. It is also extremely challenging to extract key parts when the state of the combinatorial encryption circuit is reset to a value unpredictable for the adversary between any two user-driven encryptions. This can be achieved by propagating a random plaintext through the circuit, while leaving the key constant, and simply ignoring the output of the computation. The cost of this method is obviously that the throughput is halved, which can be compensated by putting twice as many instances on the chip. Furthermore, it requires 64 bits of randomness every other clock cycle (i.e., 32 bits per cycle), which need to be generated by a PRNG. In such a case, only the currently encrypted plaintext is known to the attacker, while the previous state of the circuit can not be predicted. In that case, even encrypting the same plaintext twice leads to two vastly different power consumption (or electromagnetic emanation) footprints due to the difference in the initial state of the circuit. In other words, the encryption engine has a non-deterministic behavior and dissipation, which leads to a high level of protection. In our experiments, straightforward first-round attacks could not recover the key with up to 100 million traces.

However, there exists another attack vector with a remarkable impact on the physical security of unrolled cryptography on silicon, namely the static power consumption. Our results indicate that the static power side channel is a convenient source of information leakage for adversaries against unrolled cryptographic primitives in advanced technologies. Its independence of the execution speed, asynchronicity and glitching behavior of the circuit is a favorable advantage that leads to effective attacks. In our experiments, targeting the static power was clearly the best choice when trying to extract the full 128-bit key. Any round of the block cipher can be targeted with roughly the same effort and reset methods are useless if the adversary can stop the clock signal of the input register feeding the combinatorial circuit. The static dissipation is always deterministic, due to its independence of any previous state of the circuit. Hence, the increased security level achieved by certain usage principles does not translate to the static power side channel due to its different nature. The best chance to thwart this kind of attack is to ensure that the clock signal of the unrolled primitive is generated on silicon and can not be stopped without causing the circuit to lose its state. Yet, this is not always an option. Even without control over the

---

[9] This is only true when the adversary can not observe the physical leakage of the decryption of the ciphertext on the communication partner's side too.

clock, static power adversaries remain dangerous. To protect implementations one should always ensure that a reset of the full circuit is performed immediately in the next clock cycle after the result of a previous operation has been saved in order to not leave sensitive information behind. Only if that is guaranteed, the implementation can provide reasonable security against this type of attacker. Nevertheless, an adversary with full physical access to the target should never be underestimated. It is unclear whether possibilities exist to stop the propagation of the clock signal to a targeted cipher core, even when the oscillator is implemented on silicon and precautions are in place.

Acknowledgments

The author would like to thank the TCHES 2020 reviewers for their constructive feedback and Amir Moradi for his input and support. The work described in this paper has been supported in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and through the project 271752544 "NaSCA: Nano-Scale Side-Channel Analysis".

References

[AO14] Zia Abbas and Mauro Olivieri. Impact of technology scaling on leakage power in nano-scale bulk CMOS digital standard cells. *Microelectronics Journal*, 45(2):179–195, 2014.

[BCG+12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2012.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

[BCS+17] Davide Bellizia, Danilo Cellucci, Valerio Di Stefano, Giuseppe Scotti, and Alessandro Trifiletti. Novel measurements setup for attacks exploiting static power using DC pico-ammeter. In *2017 European Conference on Circuit Theory and Design, ECCTD 2017, Catania, Italy, September 4-6, 2017*, pages 1–4. IEEE, 2017.

[BGSD10] Shivam Bhasin, Sylvain Guilley, Laurent Sauvage, and Jean-Luc Danger. Unrolling cryptographic circuits: A simple countermeasure against side-channel attacks. In Josef Pieprzyk, editor, *Topics in Cryptology - CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings*, volume 5985 of *Lecture Notes in Computer Science*, pages 195–207. Springer, 2010.

[CRB+16] Thomas De Cnudde, Oscar Reparaz, Begül Bilgin, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Masking AES with d+1 shares in hardware. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 194–212. Springer, 2016.

[CSR+19] Nikhil Chawla, Arvind Singh, Nael Mizanur Rahman, Monodeep Kar, and Saibal Mukhopadhyay. Extracting side-channel leakage from round unrolled implementations of lightweight ciphers. In *IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2019, McLean, VA, USA, May 5-10, 2019*, pages 31–40. IEEE, 2019.

[FGP+18] Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and François-Xavier Standaert. Composable masking schemes in the presence of physical defaults & the robust probing model. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):89–120, 2018.

[GIB18] Hannes Groß, Rinat Iusupov, and Roderick Bloem. Generic low-latency masking in hardware. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):1–21, 2018.

[GJJR11] Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. A testing methodology for sidechannel resistance validation. In *NIST non-invasive attack testing workshop*, 2011. `https://csrc.nist.gov/csrc/media/events/non-invasive-attack-testing-workshop/documents/08_goodwill.pdf`.

[GM17] Hannes Groß and Stefan Mangard. Reconciling d+1 masking in hardware and software. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 115–136. Springer, 2017.

[GM18] Hannes Groß and Stefan Mangard. A unified masking approach. *J. Cryptographic Engineering*, 8(2):109–124, 2018.

[GMK16] Hannes Groß, Stefan Mangard, and Thomas Korak. Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. In Begül Bilgin, Svetla Nikova, and Vincent Rijmen, editors, *Proceedings of the ACM Workshop on Theory of Implementation Security, TISCCS 2016 Vienna, Austria, October, 2016*, page 3. ACM, 2016.

[GMK17] Hannes Groß, Stefan Mangard, and Thomas Korak. An efficient side-channel protected AES implementation with arbitrary protection order. In Helena Handschuh, editor, *Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings*, volume 10159 of *Lecture Notes in Computer Science*, pages 95–112. Springer, 2017.

[HRO20] *Teledyne LeCroy HRO Series Data Sheet.* `http://cdn.teledynelecroy.com/files/pdf/hro-12bit_datasheet.pdf`, accessed July 5th, 2020.

[KMM19] Naghmeh Karimi, Thorben Moos, and Amir Moradi. Exploring the effect of device aging on static power analysis attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):233–256, 2019.

[KR07] Ian Kuon and Jonathan Rose. Measuring the gap between fpgas and asics. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 26(2):203–215, 2007.

[Lab20] *Teledyne LeCroy HRO Series Data Sheet.* `http://cdn.teledynelecroy.com/files/pdf/labmaster-10zi-a-datasheet.pdf`, accessed July 5th, 2020.

[LvMJ95] Jeroen A. J. Leijten, Jef L. van Meerbergen, and Jochen A. G. Jess. Analysis and reduction of glitches in synchronous networks. In *1995 European Design and Test Conference, ED&TC 1995, Paris, France, March 6-9, 1995*, pages 398–403. IEEE Computer Society, 1995.

[MME10] Amir Moradi, Oliver Mischke, and Thomas Eisenbarth. Correlation-enhanced power analysis collision attack. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 125–139. Springer, 2010.

[MMP11] Amir Moradi, Oliver Mischke, and Christof Paar. Practical evaluation of DPA countermeasures on reconfigurable hardware. In *HOST 2011, Proceedings of the 2011 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 5-6 June 2011, San Diego, California, USA*, pages 154–160. IEEE Computer Society, 2011.

[MMR17] Thorben Moos, Amir Moradi, and Bastian Richter. Static power side-channel analysis of a threshold implementation prototype chip. In David Atienza and Giorgio Di Natale, editors, *Design, Automation & Test in Europe Conference & Exhibition, DATE 2017, Lausanne, Switzerland, March 27-31, 2017*, pages 1324–1329. IEEE, 2017.

[MMR20] Thorben Moos, Amir Moradi, and Bastian Richter. Static power side-channel analysis - an investigation of measurement factors. *IEEE Trans. VLSI Syst.*, 28(2):376–389, 2020.

[MMSS19] Thorben Moos, Amir Moradi, Tobias Schneider, and François-Xavier Standaert. Glitch-resistant masking revisited or why proofs in the robust probing model are needed. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):256–292, 2019.

[Moo19] Thorben Moos. Static power SCA of sub-100 nm CMOS asics and the insecurity of masking schemes in low-noise environments. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):202–232, 2019.

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards.* Springer, 2007.

[Mor14] Amir Moradi. Side-channel leakage through static power - should we care about in practice? In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 562–579. Springer, 2014.

[MS16a] Amir Moradi and Tobias Schneider. Side-channel analysis protection and low-latency in action - - case study of PRINCE and midori -. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 517–547, 2016.

[MS16b] Amir Moradi and François-Xavier Standaert. Moments-correlating DPA. In Begül Bilgin, Svetla Nikova, and Vincent Rijmen, editors, *Proceedings of the ACM Workshop on Theory of Implementation Security, TIS@CCS 2016 Vienna, Austria, October, 2016*, pages 5–15. ACM, 2016.

[NRR06] Svetla Nikova, Christian Rechberger, and Vincent Rijmen. Threshold implementations against side-channel attacks and glitches. In Peng Ning, Sihan Qing, and Ninghui Li, editors, *Information and Communications Security, 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006, Proceedings*, volume 4307 of *Lecture Notes in Computer Science*, pages 529–545. Springer, 2006.

[PSKM15] Santos Merino Del Pozo, François-Xavier Standaert, Dina Kamel, and Amir Moradi. Side-channel attacks from static power: when should we care? In Wolfgang Nebel and David Atienza, editors, *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, DATE 2015, Grenoble, France, March 9-13, 2015*, pages 145–150. ACM, 2015.

[RBN+15] Oscar Reparaz, Begül Bilgin, Svetla Nikova, Benedikt Gierlichs, and Ingrid Verbauwhede. Consolidating masking schemes. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 764–783. Springer, 2015.

[SM15] Tobias Schneider and Amir Moradi. Leakage assessment methodology - A clear roadmap for side-channel evaluations. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.

[Wav20] *Teledyne LeCroy WaveRunner 8000 Series Data Sheet.* `http://cdn.teledynelecroy.com/files/pdf/waverunner8000-datasheet.pdf`, accessed July 5th, 2020.

[YHA15] Ville Yli-Mäyry, Naofumi Homma, and Takafumi Aoki. Improved power analysis on unrolled architecture and its application to PRINCE block cipher. In Tim Güneysu, Gregor Leander, and Amir Moradi, editors, *Lightweight Cryptography for Security and Privacy - 4th International Workshop, LightSec 2015, Bochum, Germany, September 10-11, 2015, Revised Selected Papers*, volume 9542 of *Lecture Notes in Computer Science*, pages 148–163. Springer, 2015.

[YHA17a] Ville Yli-Mäyry, Naofumi Homma, and Takafumi Aoki. Chosen-input side-channel analysis on unrolled light-weight cryptographic hardware. In *18th International Symposium on Quality Electronic Design, ISQED 2017, Santa Clara, CA, USA, March 14-15, 2017*, pages 301–306. IEEE, 2017.

[YHA17b] Ville Yli-Mäyry, Naofumi Homma, and Takafumi Aoki. Power analysis on unrolled architecture with points-of-interest search and its application to PRINCE block cipher. *IEICE Transactions*, 100-A(1):149–157, 2017.

A Appendix

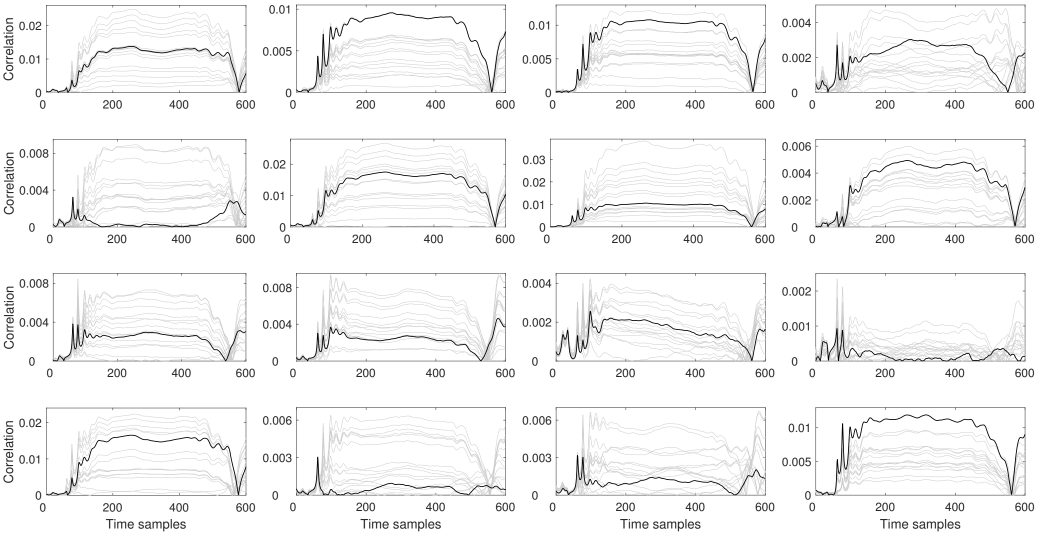


Figure 13: CPA attack on unrolled PRINCE using 100 million traces when the plaintext is reset to a random state between encryptions. Results for all 16 key nibbles are presented and the power model is the Hamming weight of the first-round Sbox output.

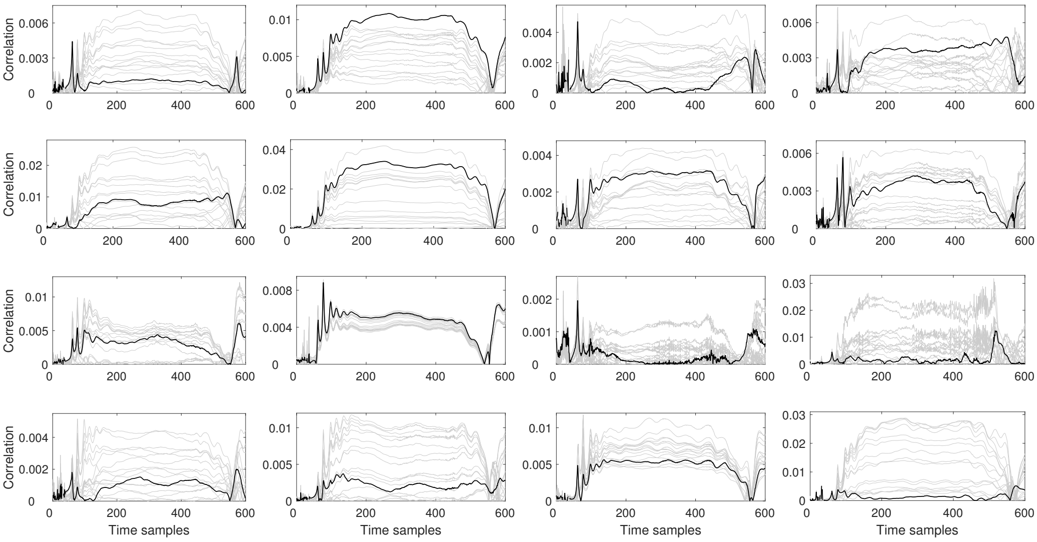


Figure 14: MCDPA attack on unrolled PRINCE using 100 million traces when the plaintext is reset to a random state between encryptions. Results for 16 key differences (0-1, 1-2, ..., 14-15, 15-0) are presented.

3.5 Countermeasures against Static Power Attacks

Publication Data

> Thorben Moos and Amir Moradi. Countermeasures against static power attacks - comparing exhaustive logic balancing and other protection schemes in 28 nm CMOS -. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):780–805, 2021

The acceptance rate for Volume 2021 of the IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES) was **31,2%** [Acca].

Content This work investigates the effectiveness of multiple hiding and masking countermeasures, as well as their combination, to protect cryptographic implementations against static power side-channel analysis. In more detail, a total of eleven different PRESENT-80 hardware co-processors with different levels of protection applied are implemented on a 28 nm CMOS ASIC and evaluated for their resistance to attacks. As a part of that analysis, this work also suggests a new countermeasure, called Exhaustive Logic Balancing (ELB), as the first ever standard-cell-based balancing scheme that achieves perfect data-independence in a simplified model. While this countermeasure comes at a significant implementation overhead it provides the best protection among all eleven co-processors when combined with a first-order masking scheme. Finally, its analysis provides an interesting case study to understand the limits of balancing techniques in general.

Contribution The author of this thesis is the principal author of this publication.

Countermeasures against Static Power Attacks

– Comparing Exhaustive Logic Balancing and Other Protection Schemes in 28 nm CMOS –

Thorben Moos and Amir Moradi

Ruhr University Bochum, Horst Görtz Institute for IT Security, Bochum, Germany
firstname.lastname@rub.de

Abstract. In recent years it has been demonstrated convincingly that the standby power of a CMOS chip reveals information about the internally stored and processed data. Thus, for adversaries who seek to extract secrets from cryptographic devices via side-channel analysis, the static power has become an attractive quantity to obtain. Most works have focused on the destructive side of this subject by demonstrating attacks. In this work, we examine potential solutions to protect circuits from silently leaking sensitive information during idle times. We focus on countermeasures that can be implemented using any common digital standard cell library and do not consider solutions that require full-custom or analog design flow. In particular, we evaluate and compare a set of five distinct standard-cell-based hiding countermeasures, including both, randomization and equalization techniques. We then combine the hiding countermeasures with state-of-the-art hardware masking in order to amplify the noise level and achieve a high resistance against attacks. An important part of our contribution is the proposal and evaluation of the first ever standard-cell-based balancing scheme which achieves perfect data-independence on paper, i.e., in absence of intra-die process variations and aging effects. We call our new countermeasure Exhaustive Logic Balancing (ELB). While this scheme, applied to a threshold implementation, provides the highest level of resistance in our experiments, it may not be the most cost effective option due to the significant resource overhead associated. All evaluated countermeasures and combinations thereof are applied to a serialized hardware implementation of the PRESENT block cipher and realized as cryptographic co-processors on a 28 nm CMOS ASIC prototype. Our experimental results are obtained through real-silicon measurements of a fabricated die of the ASIC in a temperature-controlled environment using a source measure unit (SMU). We believe that our elaborate comparison serves as a useful guideline for hardware designers to find a proper tradeoff between security and cost for almost any application.

Keywords: Static Power · Side-Channel · SPSCA · Countermeasures · Shuffling · SDRL · QuadSeal · Exhaustive Logic Balancing · Threshold Implementation

1 Introduction

Complementary Metal-Oxide-Semiconductor (CMOS) technology is the predominant standard for integrated circuit (IC) fabrication since about 40 years now. One of the main reasons for its continued dominance is the conceptually guaranteed low idle power dissipation. In contrast to other logic families, CMOS gates do not dissipate any energy in stable states unless leakage currents occur. Leakage currents are defined as the undesired transfer of electrical energy across a boundary which is technically viewed as insulating. The main example relevant in this context is the flow of current across a transistor which is in the *off* state. While these leakage currents have been negligibly small in former generations of CMOS technology, the aggressive down-scaling of the physical feature size

Table 1: Estimated leakage current of a 2-input `NOR` gate in a 22 nm CMOS technology for different input values [AO14].

A1	A2	Leakage Current [nA]
0	0	172.16
0	1	173.44
1	0	62.96
1	1	38.42

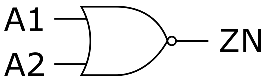
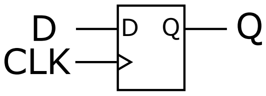


Table 2: Estimated leakage current of a `D-flip-flop` in a 22 nm CMOS technology for different input and output values [AO14].

D	CLK	Q	Leakage Current [nA]
0	0	1	421.79
0	1	0	446.39
0	1	1	370.71
1	0	0	376.11
1	0	1	441.54
1	1	0	437.42
1	1	1	386.61



throughout the past decades has led to a significant increase of their magnitude and therefore to a rise of the overall static power consumption of CMOS-based devices.

Nowadays, leakage currents are a crucial quantity to observe during the IC design flow. Modern Electronic Design Automation (EDA) tools devote a high effort towards reducing the overall leakage current of a circuit in order to keep devices suitable for battery-powered applications. Hereby, the leakage current conducted through a single standard cell in a stable state is modeled and characterized in a fairly simple manner, namely as a function of the logical signals applied. Indeed, the input values currently applied to a logic gate play a significant role in determining the leakage current conducted through the cell. The total magnitude of the current leaked by a circuit is then estimated as the sum of the individual leakage currents of all gates in the circuit. Design libraries which are used to estimate the timing, noise and power consumption during the synthesis and implementation stages of the IC design flow are typically characterized for one fixed set of temperature, supply voltage and process corner. All three of those parameters affect the leakage currents conducted by the cells globally. Yet, for one fixed set of conditions, the one local factor considered in the estimation of the current leaked is indeed the vector of logical input values applied to a logic cell. For each possible combination of inputs, one characterized magnitude of the current leaked is given by the design libraries. For memory cells like flip-flops the logical output value(s) and the clock input are also considered in the idle power estimation. For clarification, see the two exemplary leakage tables, one for a 2-input `NOR` gate in Table 1 and the other for a `D-type flip-flop` in Table 2, estimated for a 22 nm bulk CMOS process by the authors of [AO14]. It is obvious that any data value which is stored or currently processed by a standard-cell-based circuit has a direct impact on the total leakage current conducted. Therefore, it is no surprise that this quantity can be exploited via statistical analysis to learn details about the secret internals of cryptographic chips.

1.1 Related Work

The information leakage through the static power consumption of CMOS-based circuits has been identified in 2007 for the first time as a potential security threat for cryptographic hardware [GSST07]. It took until CHES 2014 before the first experimental analysis on the subject was made available in public literature [Mor14]. This work analyzed the data dependency of the leakage currents of different elements in the programmable fabric of modern FPGA devices manufactured in different nanometer-scaled technology generations. For the first time, the feasibility of such attacks was demonstrated in practice and the first implementations equipped with side-channel countermeasures had been evaluated against this new kind of analysis. In the following years, several practical case studies have been reported targeting both, programmable hardware [BCS+17] and dedicated ASIC chips [PSKM15, MMR17, KMM19, Moo19, MMR20, Moo20]. The general procedure of performing an attack based on the idle power consumption remained largely unchanged from the beginning. During the execution of the first (or last) round of a block cipher, the adversary halts the global clock signal of the device and therefore artificially creates an idle state that allows to measure the current flowing through the device without any ongoing computations for an extended period of time. Thus, the ability to halt or pause the clock signal of the device under test (DUT) is typically viewed as a requirement for this type of adversary. At CHES 2019 it was pointed out that sensitive information is often left behind by cryptographic co-processors after their operation, which allows the extraction of secret data even without any clock control abilities [Moo19]. The measurement setups used in previous practical case studies have mostly utilized an oscilloscope as the central measurement instrument for data acquisition [PSKM15, Mor14, MMR17, KMM19, Moo19, MMR20, Moo20], sometimes together with a differential probe with internal amplification [Mor14], sometimes together with custom DC amplifiers and low pass filters [MMR17, KMM19, Moo19, MMR20, Moo20]. To the best of our knowledge, the only work that used a commercial instrument dedicated for high-precision low-current measurements, namely a picoammeter, has been presented in [BCS+17]. Due to sensitive dependencies of the leakage currents on the supply voltage and the temperature, static power measurements typically require a little more care, setup-wise, to obtain the leakages in sufficient quality. In that regard, the experiments are often performed in temperature-controlled environments such as climate chambers [MMR17, KMM19, Moo19, MMR20, Moo20]. However, it was quickly discovered that the strong dependencies on environmental factors can be used in favor of the adversary to escalate the leakage of information [Moo19, MMR20]. While it appears to require a little more effort to build a setup and perform such experiments in practice (compared to dynamic power experiments), the implications can be significant once an adversary succeeds. In particular, even implementations that are typically less susceptible to passive SCA attacks or come with dedicated side-channel protections in place may be vulnerable to this kind of attack due to its different leakage mechanisms. This has been demonstrated especially with respect to dedicated logic styles [DGS+11, ABD+14], masking schemes [Mor14, MMR17, Moo19] and recently also unrolled implementations [Moo20].

With respect to dedicated countermeasures against static power attacks, the first obvious solution that comes to mind could be to build devices in such a manner that it is infeasible for an adversary to influence (esp. reduce the frequency or entirely halt) the clock signal of the circuit under analysis. Then, if the designer has also taken care that no sensitive intermediate values remain in the circuit while not currently computed upon, performing such attacks becomes virtually impossible. However, protecting the clock signal against exterior influences is easier said than done. Adversaries may employ invasive methods to stop the operation of a circuit part for some time and measure the current flowing. Or, even more importantly, depending on the functionality of a device it may be required to hold sensitive data in the circuit for an extended period of time without actively computing on it. Thus, protecting the clock signal against adversarial access is often not sufficient. From a security designer's point of view it is generally undesirable for a circuit to silently leak information about the stored data, even in the absence of computation. Therefore, it is often preferable to apply dedicated protections against this kind of attack to the sensitive circuit parts. Different kinds of countermeasures have been proposed for this purpose over the years [NYH13, HMY13, ZZL13, ZZL14, JIA+15, PR16, YK17b, YK17a, YW18, FMM20]. In this work we analyze primarily the two standard-cell-based solutions introduced in [ZZL13, ZZL14] and [JIA+15].

1.2 Our Contribution

For the first time in literature we perform a practical analysis of dedicated countermeasures against static power side-channel attacks (SPSCA) on real hardware. We have developed a prototype chip in 28 nm CMOS technology containing 11 cryptographic coprocessors with different levels of SCA protection applied and analyze the effectiveness of the countermeasures by performing practical attacks on the fabricated chip inside a climate chamber. We use a source measure unit (SMU) as power supply and high precision current measurement instrument simultaneously. Compared to previous works on dedicated test chips [MMR17, KMM19, Moo19, MMR20, Moo20], the 28 nm node constitutes the most advanced CMOS technology generation. We also make a contribution in the area of SPSCA countermeasures by proposing the first ever standard-cell-based balancing scheme that provides perfect data independence under the assumption that multiple instances of the same standard cell on the same chip have the same exact leakage characteristics. Of course, in reality this assumption can not hold due to the existence of (intra-die) process variations and aging-related degradation effects [KMM19]. Yet, our scheme, which we call exhaustive logic balancing (ELB), is likely as close as one may get towards achieving a fully data-independent static power consumption. Hence, the evaluation of this scheme gives insight about the practical limits of balancing techniques in general. Like we do for most of the hiding-based SPSCA countermeasures in this work we combine ELB with provably secure hardware masking in order to amplify the noise and show that the resulting circuit provides a high level of resistance against attacks. However, considering the very significant resource overhead of this method, some of the other countermeasure we evaluate here may be preferable from a cost efficiency standpoint. In general, our results can be used as a guideline for hardware designers to find a tradeoff between security and cost when trying to protect circuits from leaking information through the static power.

2 Countermeasures

In this section we introduce the hardware countermeasures which are implemented and practically evaluated throughout this work. Each countermeasure is applied to the serialized PRESENT-80 [BKL+07] block cipher implementation depicted in Figure 1. This area-optimized architecture has been proposed in [PMK+11].

2.1 High Threshold Voltage (HVT) [AE03]

Multi-Threshold Voltage CMOS (MTCMOS) is a popular technique available in most nanometer CMOS technology generations to reduce the leakage power of CMOS circuits while maintaining high performance. For this optimization strategy, standard cells exist in multiple versions with different threshold voltages. Cells with a lower threshold voltage (LVT) switch faster in response to their input signals and therefore are typically selected for gates in the critical path of a circuit. Cells with a higher threshold voltage (HVT) switch slower but consume a lower standby power [AE03]. In consequence, such cells are typically

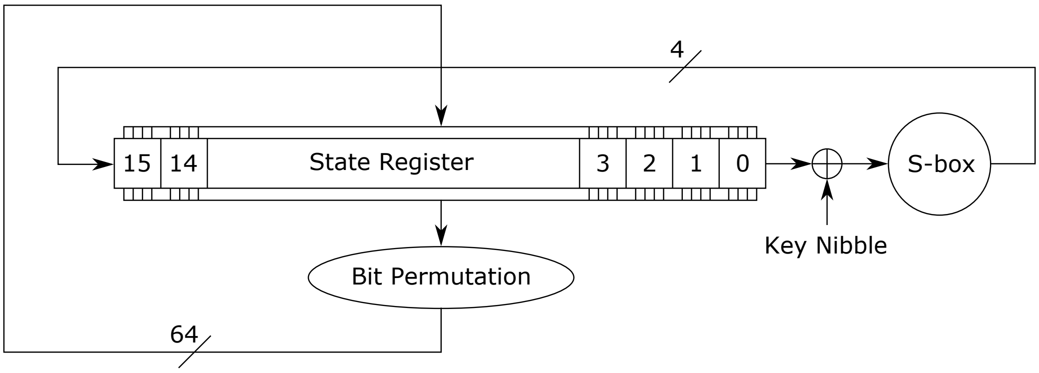


Figure 1: Architecture of the serialized PRESENT-80 hardware implementation. The key schedule is not shown.

selected for any path in a circuit where the timing constraints are not violated by the reduced performance of the cells. Although not explicitly proposed as a countermeasure against static power attacks yet, it is reasonable to assume that implementing a cryptographic primitive using only HVT cells with minimum drive strength will reduce the exploitable signal available to a static power adversary in relation to the noise level at the cost of a reduced performance of the circuit. In this work we verify whether this assumption holds by comparing two circuits derived from identical RTL code, one implemented for maximum performance and one implemented for minimum leakage current.

2.2 Random Start Index Shuffling (RSIS) [VMKS12]

Randomly changing the execution order of independent operations in a cryptographic algorithm is called shuffling and has been used as a side-channel countermeasure for many years. In modern symmetric block ciphers it is common to apply a non-linear substitution box (S-box) piecewise to the entire cipher state during the computation of each cipher round. These substitution boxes come in different sizes, but typical examples include 8-bit boxes like the AES S-box [DR98] and 4-bit boxes like the PRESENT S-box [BKL$^+$07]. The substitution functions are applied to each byte or nibble of the state independently and the order of their execution, if executed sequentially, may be randomly reshuffled in each round or cipher iteration without affecting the outcome (when implemented correctly). In both examples, AES-128 and PRESENT-80, 16 consecutive S-box evaluations are performed in each cipher round whose order may be reshuffled. Essentially, there are two common methods to implement such a shuffling. Either a Random Permutation (RP) is chosen from all $16! \approx 2^{44.25}$ permutations or a Random Start Index (RSI) is chosen from 16 possible start indices [VMKS12]. First applications of both methods have focused on software implementations [HOM06, RPD09]. Later, the RSI method in particular has also been applied to hardware circuits [MMP11]. The idea behind both shuffling techniques is simple. When observing the execution of an unprotected cipher implementation, the adversary typically knows exactly at which point in time which part of the secret key is processed and, even more importantly, that in multiple executions of the same cipher the same key parts are processed at the same points in time. When shuffling is applied and it can safely be assumed that the adversary is unable to predict the permutation or the start index chosen then there are 16 possible positions where a certain targeted key part might be processed in a cipher iteration. For the most trivial side-channel attacks this translates to a reduction of the correlation between hypothesis and leakage recorded by a factor of about 16. The authors of [VMKS12] mention that this factor can be reduced to $\sqrt{16} = 4$ when

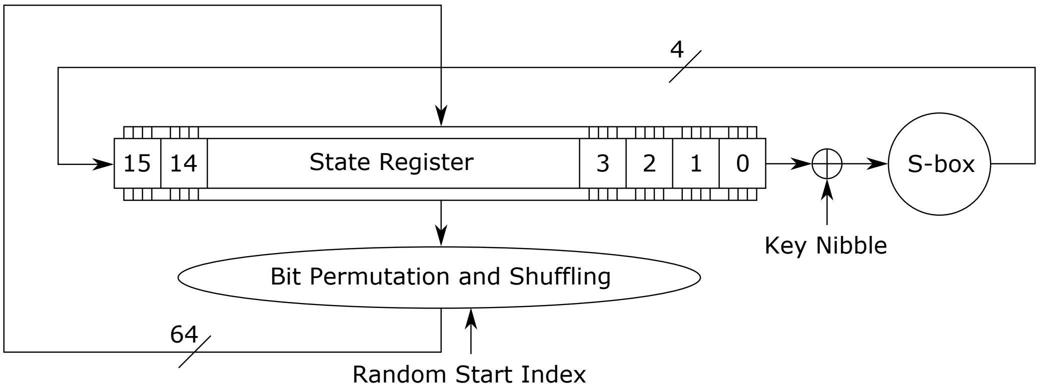


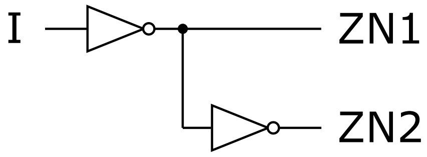
Figure 2: Architecture of the serialized PRESENT-80 implementation with Random Start Index Shuffling (RSIS). The key register, which also needs to be shuffled, is not shown.

so-called integrated DPA (or sliding window DPA) is used. The authors also voice some concerns regarding the perceived effectiveness of shuffling methods in general and Random Start Index (RSI) shuffling in particular when information about the chosen permutation or start index is leaked. However, this is less of a concern for hardware implementations where the randomness generation, reshuffling and cipher execution can be performed in parallel.

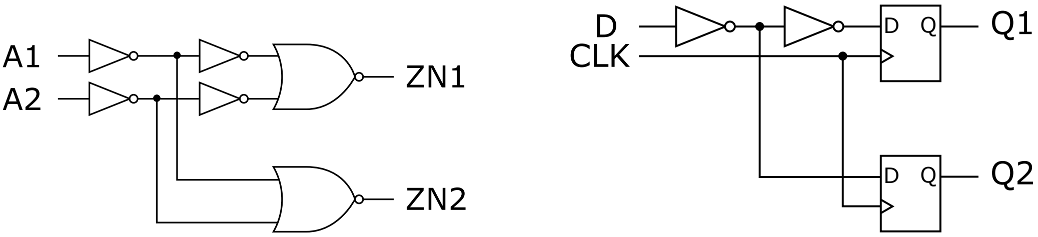
In this work we consider the RSI approach and apply it to a serialized PRESENT hardware implementation. A schematic of the result can be seen in Figure 2. The bit permutation operation, which previously could be realized purely through wire routing without any logic components, now receives the 4-bit random start index which determines by how many nibbles (0 to 15) the state register should be rotated. Clearly, this adds logic for the multiplexing, but since it is realized fully combinatorial, this change has no impact on the number of clock cycles required per encryption. Please note, that the same multiplexing logic is also required to rotate the current round key. While shuffling has primarily been proposed as a countermeasure against dynamic power or radiation side-channel attacks, it seems reasonable to expect that it also increases the difficulty of static power side-channel attacks. Especially when considering that the typical SPSCA adversary does not record a trace over time, but rather takes a single snapshot of the current state in the circuit. Thus, there are some qualitative differences between the impact of shuffling on the success probability of static power and dynamic power attacks which are discussed in Section 4.

2.3 Symmetric Dual-Rail Logic (SDRL) [ZZL13, ZZL14]

Symmetric Dual-Rail Logic (SDRL) has been proposed in [ZZL13, ZZL14] as the first standard-cell-based balancing technique dedicated to counteract static power attacks. The concept is very simple. In order to reduce the correlation between input vector applied and the leakage current of a certain cell, each standard cell is duplicated and the duplicated cell receives the inverted input vector. The general concept is illustrated for three exemplary cells in Figure 3. Please note, that outputs of gates which are not required can be left unconnected. Yet, a designer needs to make sure that the EDA tools do not remove gates whose output is not connected. From a high-level perspective, the inverter (or buffer) gate is perfectly balanced since each inverter receiving a logical ’0’ is accompanied by a second inverter receiving a logical ’1’. Under the assumption that both inverters are instantiations of the same standard cell (including drive strength, threshold voltage, etc.) and that identical standard cells have identical leakage characteristics, the total leakage current should be indistinguishable regardless of which inverter receives the logical ’0’ and which



(a) `INV` (or `BUF`)



(b) `NOR`

(c) `D-flip-flop`

Figure 3: `INV` (or `BUF`), `NOR` and `D-flip-flop` in Symmetric Dual-Rail Logic (SDRL)

receives the logical '1'. Please note, that in contrast to dual-rail logic styles which are used as a countermeasure against dynamic power attacks, timing differences through imbalanced routing or other implementation specifics are not a concern here, since the timing of signals does not affect the static power consumption measured in a stable state. Hence, the inverter (or buffer) circuit should have a data-independent static power consumption. In reality, this is not entirely true, since two instances of the same CMOS standard cell never have exactly the same physical and electrical characteristics due to (intra-die) process variations [DGS$^+$11, ABD$^+$14] and aging-related degradation of transistors [KMM19]. Therefore, a small data-dependency of the leakage current typically remains. Regarding the other two gates in Figure 3, namely the `NOR` and the `D-flip-flop`, the situation is different. Here, the balancing is not perfect, even without considering process variations and aging mechanisms. However, a reduced dependency between input pattern and leakage current is achieved. Consider the SDRL `NOR` gate. The two cases (`A1=0`, `A2=0`) and (`A1=1`, `A2=1`) are indistinguishable when assuming identical leakage characteristics for multiple instance of identical gates, and so are (`A1=0`, `A2=1`) and (`A1=1`, `A2=0`). However, the two cases (`A1=0`, `A2=0`) and (`A1=0`, `A2=1`) are not indistinguishable from each other. Taking the numbers provided in Table 1 as an example, two `NOR` gates with inputs (`A1=0`, `A2=0`) and (`A1=1`, `A2=1`) have a leakage current of $172.16\,\text{nA} + 38.42\,\text{nA} = 210.58\,\text{nA}$, while two `NOR` gates with inputs (`A1=0`, `A2=1`) and (`A1=1`, `A2=0`) have a leakage current of $173.44\,\text{nA} + 62.96\,\text{nA} = 236.40\,\text{nA}$. In summary, the variation in the leakage current caused by different input vectors is decreased but not eliminated. A similar observation can be made for the `D-flip-flop`, since the value of `Q` also affects the leakage current. In order to investigate the effectiveness of this balancing technique to counteract static power analysis attacks in practice we have synthesized the serialized PRESENT implementation from Figure 1 exclusively with `INV`, `NOR` and `D-flip-flop` gates and replaced each cell with its SDRL counterpart before implementing the circuit on the chip. The results of its security analysis are presented in Section 4.

2.4 Quadruple Algorithmic Symmetrizing (QuadSeal) [JIA$^+$15]

Quadruple Algorithmic Symmetrizing (QuadSeal) has been proposed as a countermeasure against both dynamic and static power analysis attacks in [JIA$^+$15]. The goal of this method is to balance all Hamming weights and distances occurring in a cipher implementation and rotating the inputs to the balanced structures to account for remaining dependencies due

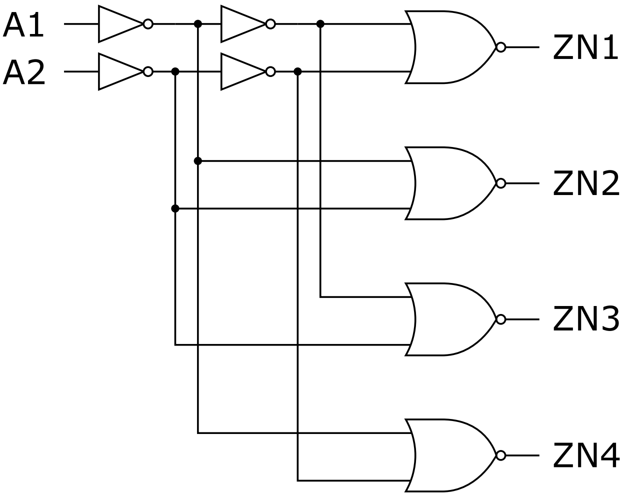


Figure 4: `NOR` gate with Exhaustive Logic Balancing (ELB).

to process variations, path imbalances and aging effects. In more detail, when applying this countermeasure to a cipher implementation, the unprotected circuit is quadrupled, while in three of the four circuits the S-box table is modified in the following way.

$$\mathrm{S}(\mathrm{state\_nibble} \oplus \mathrm{key\_nibble}) \tag{1}$$

$$\mathrm{S}^T(\mathrm{state\_nibble} \oplus \overline{\mathrm{key\_nibble}}) \tag{2}$$

$$\overline{\mathrm{S}}^T(\overline{\mathrm{state\_nibble}} \oplus \mathrm{key\_nibble}) \tag{3}$$

$$\overline{\mathrm{S}}(\overline{\mathrm{state\_nibble}} \oplus \overline{\mathrm{key\_nibble}}) \tag{4}$$

Then one of $4! = 24$ different permutations of inputs, keys and inverted inputs and keys is randomly selected (i.e., a 5-bit random number generator is sufficient). The full list of permutations is given in [JIA+15]. While the balancing of Hamming weights and Hamming distances is valuable to protect against attacks, the static power consumption of a combinatorial circuit like an S-box typically does not directly depend on the number of logical '1's in the inputs (see [KMM19] for an example of the input dependency of the leakage current exhibited by the PRESENT S-box in 65 nm CMOS). Thus, while this method is able to significantly reduce leakages from registers, it does not necessarily reduce the leakage from combinatorial S-boxes as well.

2.5 Exhaustive Logic Balancing (ELB) [this work]

In this paragraph we introduce Exhaustive Logic Balancing (ELB). ELB follows a similar concept as SDRL, but goes a step further. In particular, ELB makes sure that each gate is multiplied as often as the total number of different input vectors it may receive, and that in any stable state each of those input vectors is applied to one of the gates. An inverter or buffer gate with one input line can receive two different values, either logical '0' or logical '1'. Hence, the gate is duplicated and the gates can be connected as shown in Figure 3. A `NOR` gate with two input lines can receive four different input vectors and therefore needs to be quadrupled. In order to make sure that each input vector is received by one of the four `NOR` gates, the circuit has to be constructed as shown in Figure 4. Again, the output lines which are not required can be left unconnected, as long as it can be ensured that the IC design tools do not remove logic gates whose output is unconnected. Technically, any two-input logic gate can be quadrupled and implemented like this to

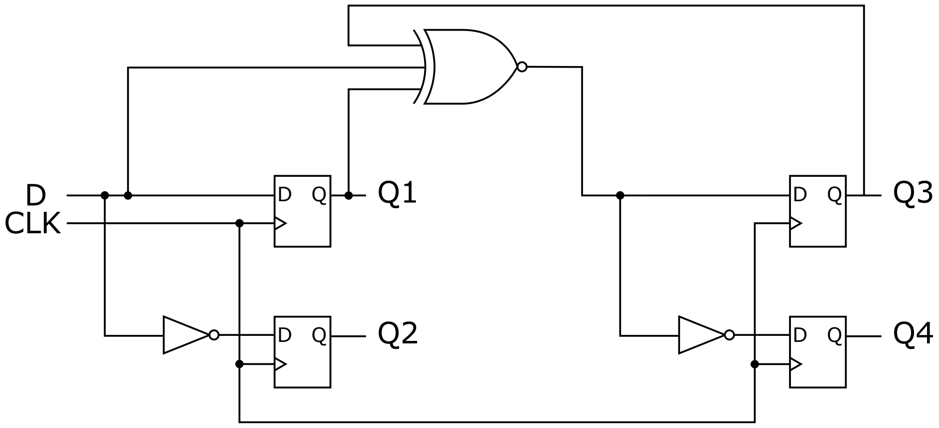


Figure 5: Logically balancing all D and Q values in the four `D-flip-flops` with a 3-input `XNOR`.

achieve a data-independent static power dissipation under the assumption that multiple instances of the same cell share exactly the same electrical characteristics. While this assumption is not precisely correct in reality, the characteristics of multiple instances of identical standard cells in close proximity on the same die should at least show a very similar electrical behavior. Unfortunately, the situation is more complex with respect to memory cells like flip-flops. At first sight it may appear that each flip-flop has only one data input and therefore simply needs to be duplicated, while giving the inverted input to the second flip-flop. However, as shown in Table 2, the leakage of a flip-flop also depends on the output value `Q`. Hence, the leakage depends on two data lines, which can have four different possible combinations. Therefore, each flip-flop needs to be quadrupled. Since `Q` is an output we can not apply the same technique as for the `NOR` gate above. Instead, we need to choose the input values for the four flip-flops as a function of their output values. One possible solution is shown in 5. Whenever applying a data value to input `D` and clocking once, each of the following four combinations is applied to one of the flip-flops: (`D=0`, `Q=0`), (`D=0`, `Q=1`), (`D=1`, `Q=0`), (`D=1`, `Q=1`). However, since the `XNOR` used for the logic function now causes its own data-dependent leakage current we need to replace it by a circuit with a balanced static power consumption. In this regard, we first express the `XNOR` function through only `NOR` and `INV` gates. The result is depicted in Figure 6. As a next step we replace those gates by their balanced version and apply some simple logic optimizations in order to reduce the number of balanced gates that have to be instantiated. The result can be seen in Figure 7. This final result achieves the optimal data independency that we are looking for. However, it is clear that the overhead to replace each flip-flop by this structure when trying to power balance a circuit is significant. The protection against static power attacks provided by this approach is analyzed experimentally in Section 4.

2.6 Threshold Implementation [NRR06]

Threshold Implementations (TIs) have been introduced in 2006 as the first hardware masking scheme that provides provable first-order security in the presence of glitches [NRR06]. Before the introduction of threshold implementations, masking schemes did not consider glitch resistance as a design objective and thus were commonly susceptible to a temporary recombination of the masks and masked values in combinatorial logic when the protected cryptographic primitive was realized as a hardware circuit. In consequence, early masking schemes could not easily guarantee practical first-order (or even higher-order) security in hardware. Since the introduction of TIs in 2006, the field of glitch-resistant masking has

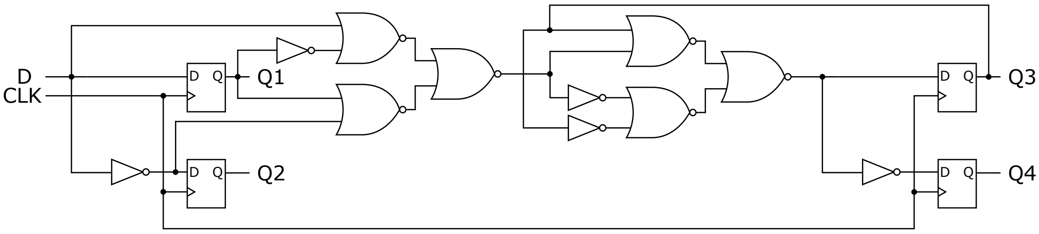


Figure 6: Logically balancing all D and Q values in the four `D-flip-flops` with only standard `NOR` and `INV` gates.

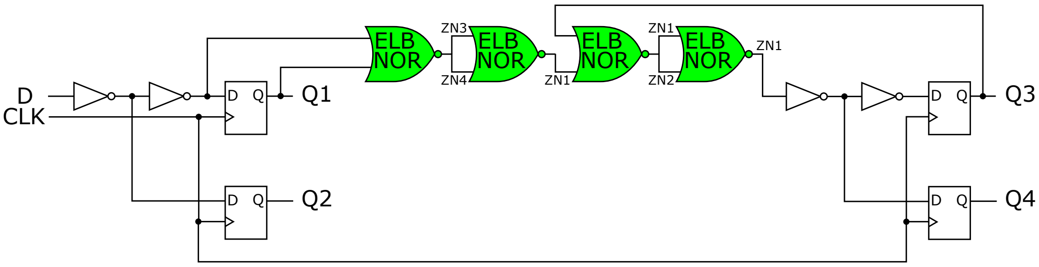


Figure 7: `D-flip-flop` with Exhaustive Logic Balancing (ELB). Each `ELB NOR` is an instantiation of Figure 4. The outputs to select (ZN1, ..., ZN4) are given.

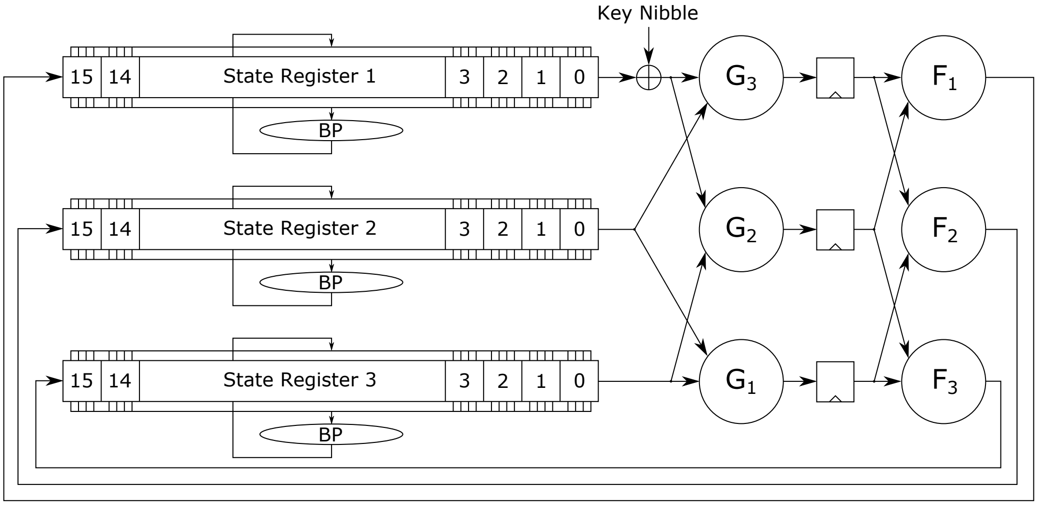


Figure 8: Architecture of the serialized PRESENT-80 threshold implementation. The key schedule is not shown.

grown significantly and many different schemes have been proposed and analyzed, including but not limited to [RBN+15, CRB+16, GMK16, GMK17, GM17, BDF+17, GM18, GIB18, FGP+18, MMSS19, CGLS20, CS20, SM21, CS21]. Yet, the plain and simple first-order threshold implementations are still one of the most popular SCA countermeasures today and arguably the easiest method to achieve provable first-order security in presence of glitches without requiring online randomness (although a recent work describes a $d+1$-masking scheme without the need for fresh randomness [SM21]).

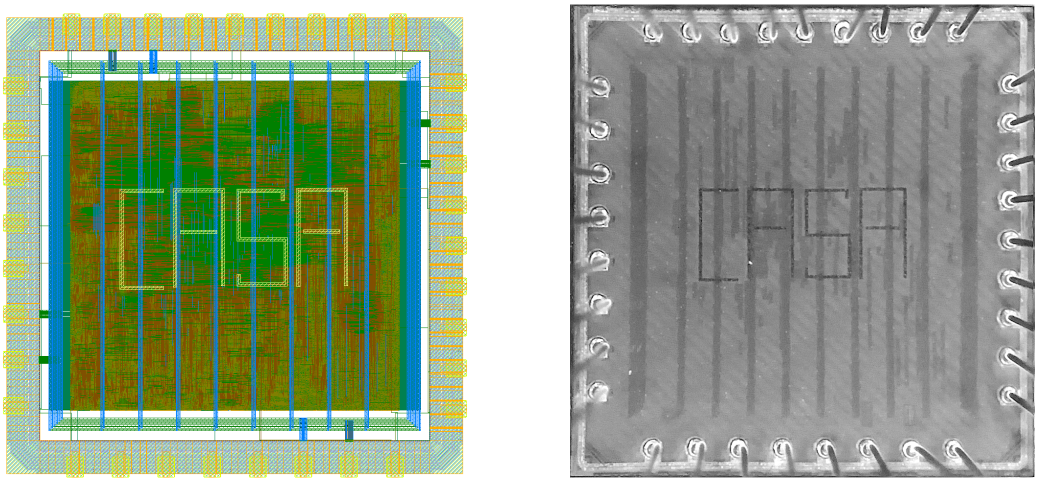
Previous works have indicated that static power side-channel adversaries may potentially be able to exploit the higher-order leakages of threshold implementations (and other masking schemes) with a lower data complexity than attackers who observe the dynamic power consumption or radiation [MMR17, Moo19]. Yet, such higher-order leakages can be hidden effectively when the masking schemes are combined with proper hiding countermeasures. In that case even static power adversaries able to acquire measurements with low environmental and electronic noise influences are expected to require prohibitively large amounts of observations in order to extract sensitive information. Thus, in this work, we not only analyze the effectiveness of threshold implementations alone, but also in combination with the hiding countermeasures introduced earlier in this section and draw conclusions about the resulting protection levels.

A first-order threshold implementation of the serialized PRESENT architecture has been proposed in [PMK+11]. Since the PRESENT S-box has an algebraic degree of 3, its TI would normally require at least 4 shares to provide first-order security ($td+1$). However, a cheaper alternative than a 4-share TI can be achieved when decomposing the S-box S into two functions F and G with algebraic degree 2. In that case, the TI can be implemented with only 3 shares as long as a register stage separates the component functions of F and G in order to prevent glitch propagation between the combinatorial circuits. This type of 3-share first-order TI with a decomposed S-box has been introduced in [PMK+11] and is shown in Figure 8 applied to our serialized architecture. We use this implementation for our test circuits.

While most of the hiding countermeasures could be applied in a straightforward manner to the threshold implementation of PRESENT, this was not the case for QuadSeal. When attempting to combine the two countermeasures we encountered conceptual problems. The idea of QuadSeal is based on balancing Hamming weights and Hamming distances at the register stages to thwart side-channel leakage (dynamic and static). To achieve this, QuadSeal requires the implementation of four different (although related) substitution boxes. Thus, all four S-boxes need to be implemented as separate TIs. Additionally, in order to stick with the 3-share TI, each of them needs to be decomposed into two quadratic functions. This is possible for the four S-boxes required for QuadSeal PRESENT, but we have found no way of implementing the shared evaluation of their component functions in such a way that their collective outputs at each stage have a balanced Hamming weight and Hamming distance. In all evaluated cases either the intermediate register between the component functions, or their output was not properly balanced considering the whole quadrupled circuit. Therefore, we refrained from implementing a hybrid circuit that only realizes one of the two concepts properly.

3 Target and Setup

In the following we introduce the target device analyzed in this work and the measurement setup and procedure used to acquire the experimental results presented in Section 4.



(a) Layout

(b) Photo

Figure 9: Layout and microscope photography of the 1380 µm × 1380 µm large 28 nm ASIC prototype.

3.1 Device Under Test (DUT)

The target for our practical analysis is a 28 nm CMOS ASIC prototype which we developed as a dedicated test chip for our investigation. The layout of the chip and a microscope photography of the manufactured and bonded die can be seen in Figure 9. The ASIC is 1380 µm × 1380 µm in size and has been designed to be operated at frequencies up to 100 MHz even under worst-case operating conditions. The chip requires an IO power supply of 1.8 V and a 0.9 V core power supply. The 981 µm × 981 µm large standard cell area in the center of the die contains 1 195 507 gate equivalents (GE) of logic. This includes 11 cryptographic co-processors based on the PRESENT block cipher which are practically analyzed for their static power side-channel security in Section 4. To be precise, each co-processor is based on the serialized PRESENT architecture described in Section 2 which is depicted in Figure 1 without masking applied and in Figure 8 as a threshold implementation. The 11 cipher cores differ from each other in the particular countermeasures that are employed to avoid key extraction via static power side-channel analysis. The levels of protection range from an unprotected circuit to a combination of exhaustive balancing and provably-secure masking. The full list of circuits evaluated in this work including their post-layout area consumption and an overhead comparison is given in Table 3.

The PRESENT core denoted by *High Performance (HP)* is a raw and unprotected implementation of the serialized cipher architecture shown in Figure 1, but optimized for maximum clock frequency. As already discussed in Section 2 such an optimization goal favors the use of low threshold voltage (LVT) cells in all timing critical paths. It is noteworthy that the HVT circuit, optimized for minimum leakage current, is smaller than the HP circuit, despite the fact that the slower high threshold voltage (HVT) cells are identically sized as the faster low threshold voltage (LVT) cells. The difference comes from the selection of standard cells with the lowest drive strength in the HVT circuit, which are generally smaller and consume less power than cells with a higher driving strength. Table 3 clearly shows that all other protected circuits come at an area overhead, which proves to be significant in some cases. Table 4 presents post-layout estimations of the critical path delay (or latency), maximum operating frequency and average power consumption (when operated at 100 MHz) for typical operating conditions (25 °C, 0.9 V) of all 11 circuits,

Table 3: Post-layout area consumption of the PRESENT co-processors.

PRESENT Core	Area [GE]	Overhead factor
High Performance (HP)	2 535.00	× 1.00
High Threshold Voltage (HVT)	2 406.67	× 0.95
Random Start Index Shuffling (RSIS)	2 613.00	× 1.03
Symmetric Dual-Rail Logic (SDRL)	10 789.33	× 4.26
Quadruple Algorithmic Symmetrizing (QuadSeal)	12 636.33	× 4.98
Exhaustive Logic Balancing (ELB)	20 207.00	× 7.97
Threshold Implementation + HP	7 233.33	× 2.85
Threshold Implementation + HVT	6 982.67	× 2.75
Threshold Implementation + RSIS	9 856.33	× 3.89
Threshold Implementation + SDRL	27 907.33	× 11.01
Threshold Implementation + ELB	58 442.33	× 23.05

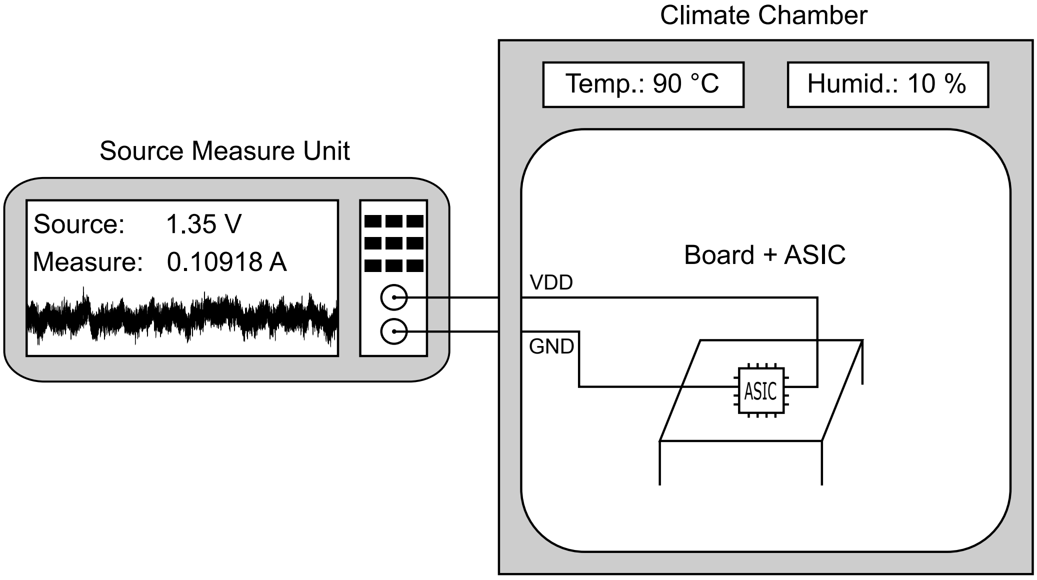
Table 4: Post-layout estimations of the critical path delay, maximum frequency and average power consumption at 100 MHz operation for all PRESENT co-processors.

PRESENT Core	Crit. Path [ps]	Freq. [MHz]	Dyn. Power [µW]	Stat. Power [µW]
HP	435.7851	2294.7	111.2826	35.9710
HVT	563.7664	1773.8	107.7309	1.7616
RSIS	597.3369	1674.1	103.4829	12.3302
SDRL	2064.9476	484.3	240.2608	4.1869
QuadSeal	785.0767	1273.8	463.3175	51.4306
ELB	1959.9415	510.2	673.4377	9.4815
TI + HP	358.3832	2790.3	277.4649	101.9850
TI + HVT	594.5498	1681.9	309.3094	3.7409
TI + RSIS	612.0424	1633.9	312.5164	54.4392
TI + SDRL	2510.5112	398.3	650.3030	7.7135
TI + ELB	2377.2272	420.7	1981.1074	17.3661

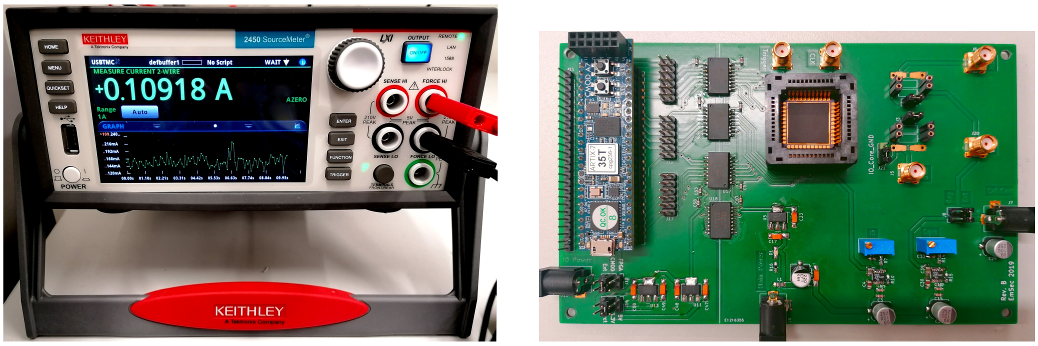
extracted using the Synopsys IC design flow. While all PRESENT cores require the same number of clock cycles (547) for one encryption (see [PMK+11]), the protected versions clearly show a reduced maximum frequency and average power consumption compared to the unprotected implementation. It is important to clarify, however, that the ASIC has been designed to operate at frequencies up to 100 MHz and even the slowest circuits in Table 4 achieve frequencies well above that threshold (at least for typical operating conditions). Thus, none of the circuits except the HP versions have been tightly constrained by their clock period and higher frequencies at the price of an increased area and energy consumption would definitely be possible. The comparably low static power consumption for SDRL and ELB circuits can be explained by the fact that they also consist of HVT cells with minimum drive strength exclusively. In part, this also causes their significantly higher latency compared to the other circuits.

3.2 Measurement Setup

The measurement setup utilized in our practical experiments is depicted in Figure 10. On top, a schematic of the full setup used to acquire the current measurements is given. The measurement board containing the mounted chip is placed inside a climate chamber to precisely control the environmental temperature. In contrast to all previous works we have used a source measure unit (SMU) a.k.a. sourcemeter for the static power measurements.



(a) Setup



(b) Source Measure Unit (SMU)

(c) Custom Measurement Board

Figure 10: Measurement Setup used in the practical experiments.

A photography of the Keithley 2450 SourceMeter [Kei] is shown in Figure 10(b). This instrument has specifically been designed for characterizing nano-scale semiconductors and other small-geometry and low-power devices. In our experiments we have used it to simultaneously supply the core voltage to the chip and measure the leakage currents through the device. A photography of the custom measurement board can be seen in Figure 10(c). We have designed this PCB for evaluating our 28 nm test chip which can be seen in the middle of the board plugged into a PLCC44 socket. A Digilent Cmod A7 FPGA board [Dig] can be plugged into the 48 pin DIP socket on the left of the measurement board in order to function as an interface between the ASIC and the PC.

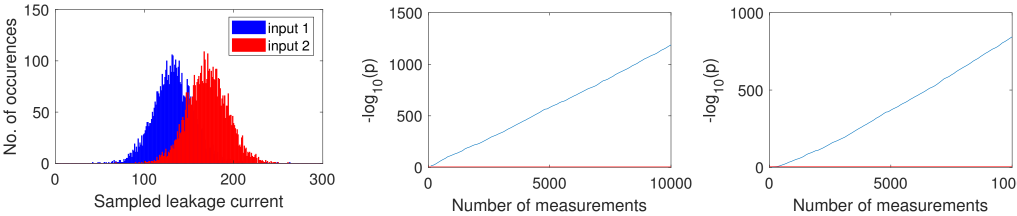
The procedure to acquire static power measurements using this setup works as follows. The FPGA board pauses the global clock signal of the ASIC during the first round of the PRESENT cipher operation and simultaneously generates a trigger signal to the SMU. The SMU waits for 20ms after receiving the positive trigger edge, takes a current measurement and saves it into the internal buffer. Afterwards, the SMU goes back to idle mode, waiting for the next trigger to arrive. The clock signal is continued and the PRESENT core completes its computation. Then a new encryption is initiated and the process repeats from the beginning. As soon as 100 measurements are collected, the internal buffer is read out by the measurement script and the data is saved on the hard drive. Using this method, the data acquisition takes about 108.24 ms per measurement (the time required for fetching the buffer from the instrument and saving the traces to the hard drive is already included), which means that the acquisition of 1 000 000 traces takes about 30 hours and 4 minutes. This is significantly faster than many previous works [MMR17, KMM19, Moo19, Moo20].

4 Experimental Results

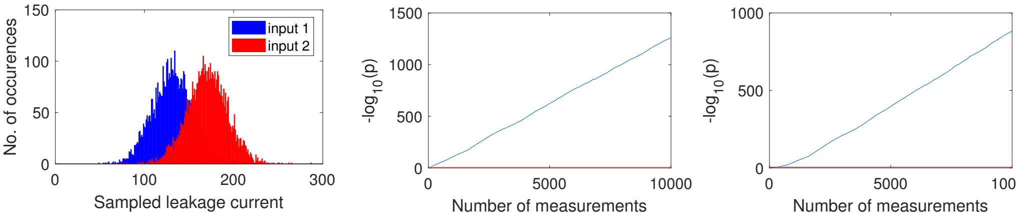
In this section we present our experimental analysis of the 11 different PRESENT co-processors realized on the 28 nm CMOS ASIC. As a first step we analyze the hiding countermeasures alone, under normal operating conditions. In this regard, we have placed the measurement board in the climate chamber set to a constant temperature of 20 °C and powered the ASIC by its nominal core voltage of 0.9 V. To compare the leakage exhibited by the PRESENT cores in this scenario we have collected measurements for two different fixed inputs (fixed-vs-fixed) in a randomly interleaved manner in order to perform a leakage assessment using the t-test [SM15] and the χ^2-test [MRSS18] respectively. The results are depicted in Figure 11. For a visual comparison between the different techniques we have also plotted the histograms for the two groups which have been used to extract the -$\log_{10}$(p) confidence values.

The first thing to notice is that the HVT circuit does not seem to hide the data dependency any better than the high performance (HP) implementation. The differences between the means of the leakage distributions and the test results look very similar in both cases. The other four protected implementations perform better and a gradual reduction of the test confidence and the number of traces to overcome the confidence threshold can be observed from top to bottom. The exhaustive logic balancing (ELB) achieves the best results in these experiments, since both statistical tests fail to reject the null hypothesis given a data set of 10 000 traces. Since multiple previous works have demonstrated that increasing the supply voltage and the temperature of the device under test significantly increases the leakage currents in relation to the measurement noise [Moo19, MMR20], we have repeated the same measurements as before with the temperature set to 90 °C and a core supply voltage of 1.35 V (50% over-voltage). Those results are shown in Figure 12. All results are improved by a significant margin in terms of confidence and number of traces to detect leakage. Therefore, we are able to confirm that the manipulation of operating conditions is a viable method to enhance the magnitude of the leakage currents and to improve the overall quality of the measurement results. Apart from the significantly increased distinguishability across the board, the most interesting observation is probably that the ELB circuit now also shows a significant amount of leakage. Hence, we can conclude that the variations of the physical and electrical characteristics between identical CMOS standard cells placed in close proximity to each other is certainly large enough to weaken the balancedness of the static power consumption sufficiently to detect a clear data dependency. In part this may be caused by the (uneven) aging-related degradation of the transistors which is immediately amplified when the power supply and temperature are as drastically increased as in our experiments. However, we have used a fresh sample of the ASIC for these experiments to avoid a prior manifestation of effects like described in [KMM19].

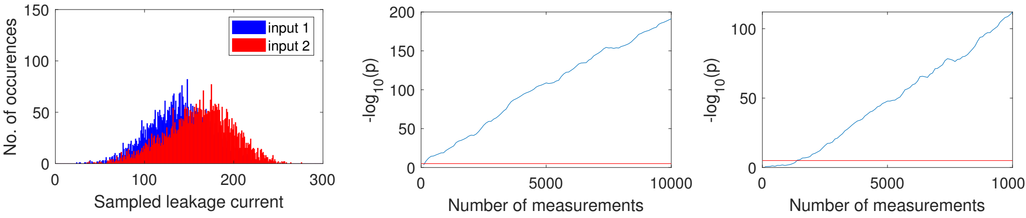
As a next step we now analyze the combined hiding and masking countermeasures, again under the leakage-enhancing operating conditions of 90 °C and 1.35 V. The results are depicted in Figure 13. Here, the t-test is performed at first, second and third order. It can be seen that no data dependency is reported with confidence for any of the first- or second-order tests. The χ^2-test is independent of statistical moments and, like the third-order t-test, reports leakage in four of the five experiments. Only for the combination of the threshold implementation with the exhaustive logic balancing the tests fail to reject



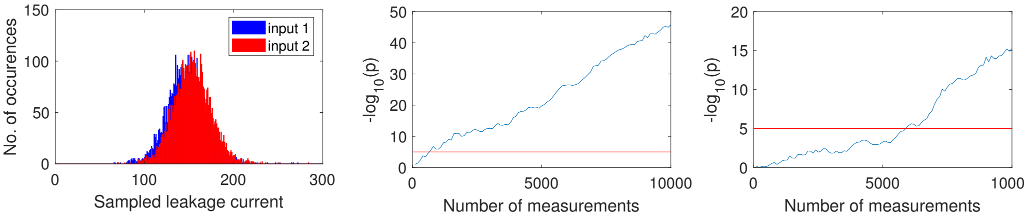
(a) histogram for HP (b) t-test for HP (c) χ^2-test for HP



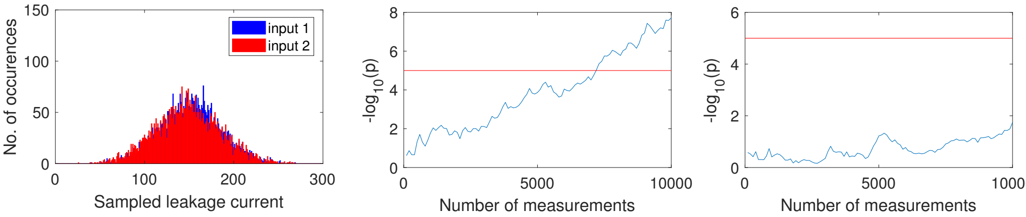
(d) histogram for HVT (e) t-test for HVT (f) χ^2-test for HVT



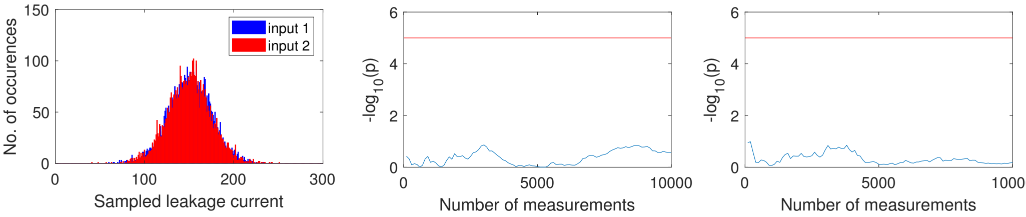
(g) histogram for RSIS (h) t-test for RSIS (i) χ^2-test for RSIS



(j) histogram for SDRL (k) t-test for SDRL (l) χ^2-test for SDRL

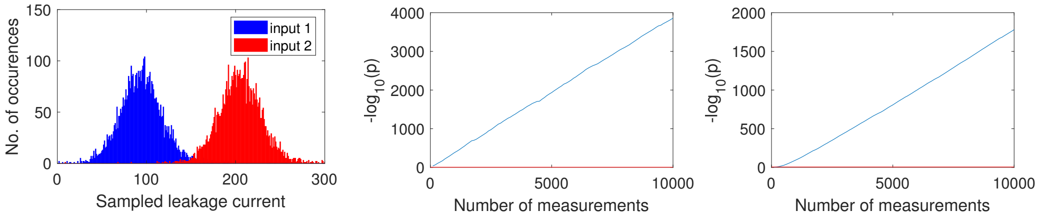


(m) histogram for QuadSeal (n) t-test for QuadSeal (o) χ^2-test for QuadSeal

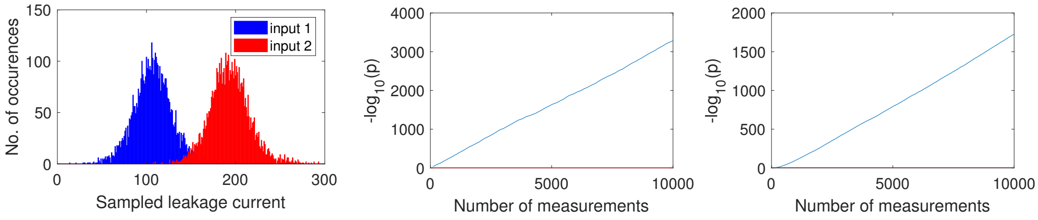


(p) histogram for ELB (q) t-test for ELB (r) χ^2-test for ELB

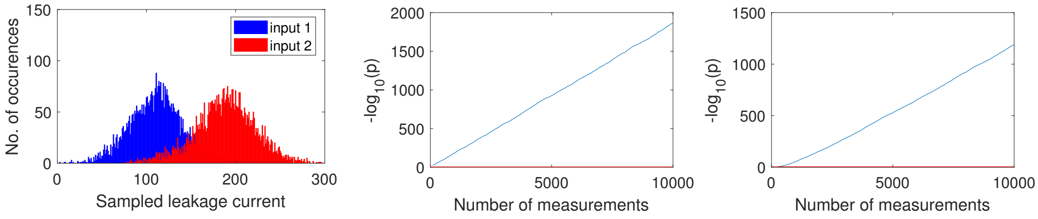
Figure 11: Leakage assessment of the (unmasked) hiding countermeasures at 20 °C, 0.9 V.



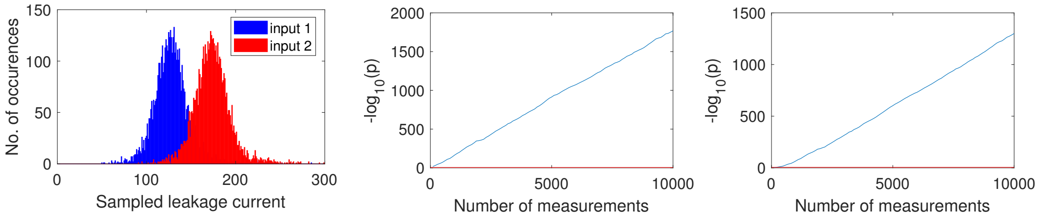
(a) histogram for HP (b) t-test for HP (c) χ^2-test for HP



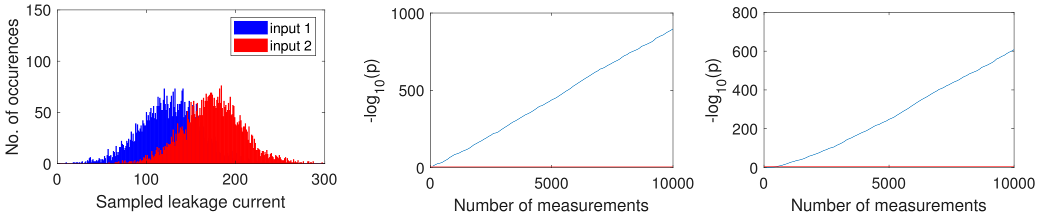
(d) histogram for HVT (e) t-test for HVT (f) χ^2-test for HVT



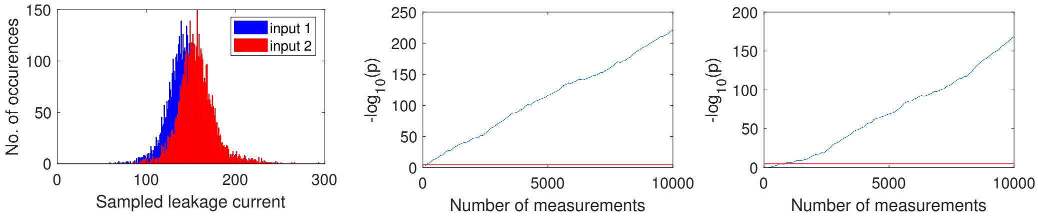
(g) histogram for RSIS (h) t-test for RSIS (i) χ^2-test for RSIS



(j) histogram for SDRL (k) t-test for SDRL (l) χ^2-test for SDRL

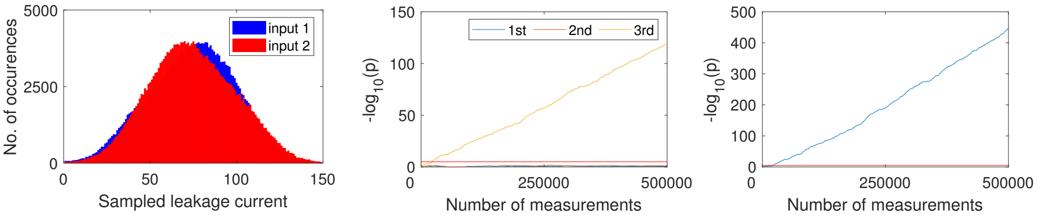


(m) histogram for QuadSeal (n) t-test for QuadSeal (o) χ^2-test for QuadSeal

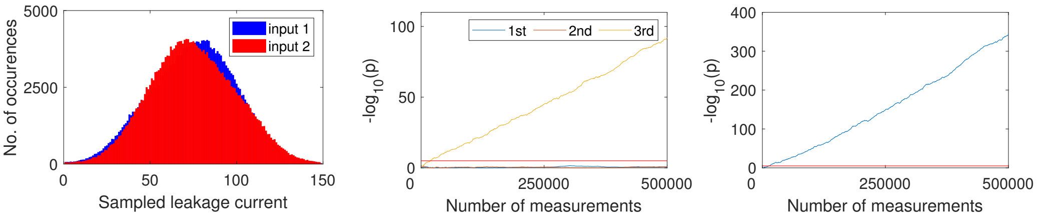


(p) histogram for ELB (q) t-test for ELB (r) χ^2-test for ELB

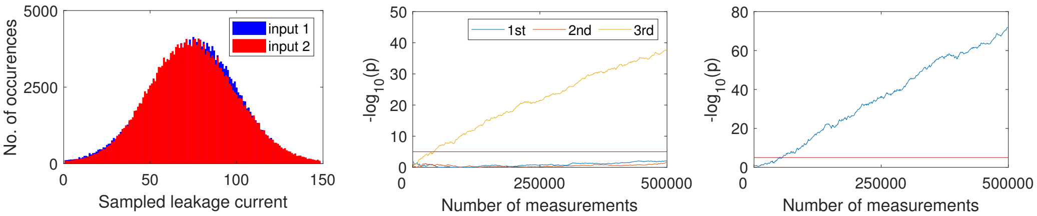
Figure 12: Leakage assessment of the (unmasked) hiding countermeasures at 90 °C, 1.35 V (50% over-voltage).



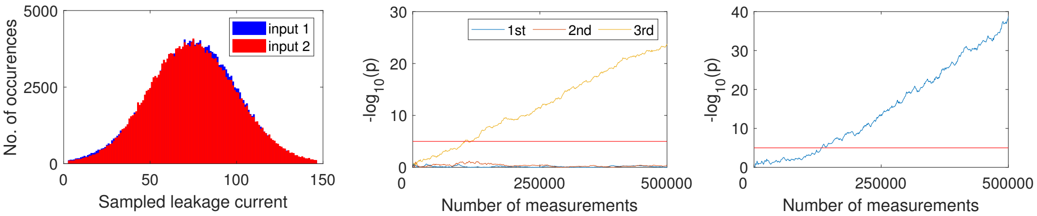
(a) histogram for TI HP (b) *t*-test for TI HP (c) χ^2-test for TI HP



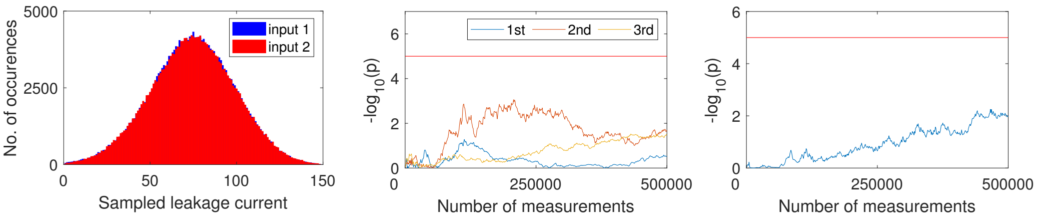
(d) histogram for TI HVT (e) *t*-test for TI HVT (f) χ^2-test for TI HVT



(g) histogram for TI RSIS (h) *t*-test for TI RSIS (i) χ^2-test for TI RSIS



(j) histogram for TI SDRL (k) *t*-test for TI SDRL (l) χ^2-test for TI SDRL



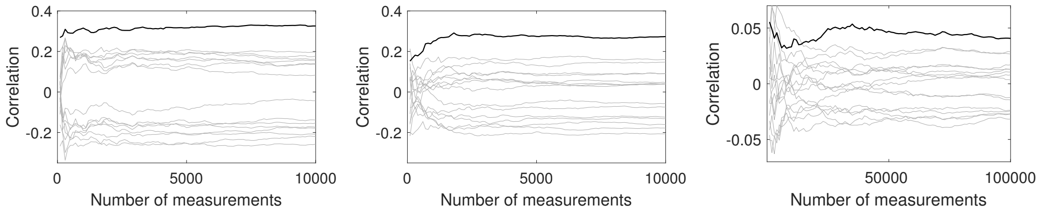
(m) histogram for TI ELB (n) *t*-test for TI ELB (o) χ^2-test for TI ELB

Figure 13: Leakage assessment of the combined masking and hiding countermeasures at 90 °C, 1.35 V (50% over-voltage).

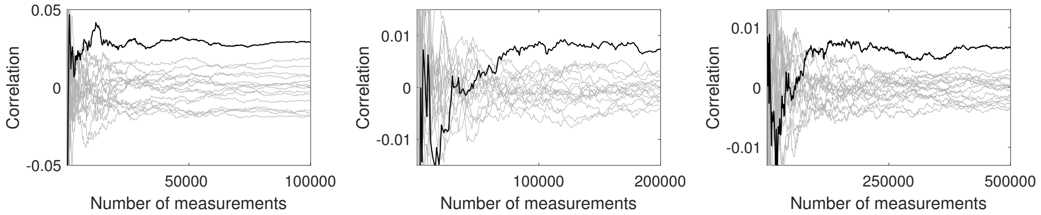
the null hypothesis given 500 000 traces. Our leakage assessment results already give us a decent idea about the level of protection each of the (combined) countermeasures is able to provide. However, the presence of detectable leakage alone does not necessarily prove the insecurity of a device. Leakages about the inputs and outputs, independent of the secret internal key, are flagged by leakage detection methods but do not necessarily undermine the security of a device. Hence, we have also attempted to perform key recovery attacks on all 11 different PRESENT co-processors. In order to provide a fair comparison we chose to perform Moments-Correlating DPA (MCDPA) attacks on the targets as this collision-based method does not depend on the choice of a suitable leakage model [MS16]. This type of attack has already been applied to static power measurements in [MMR17]. We have also attempted classical CPA attacks with explicit leakage models [BCO04], but learned that this approach leads to a distortion of the comparison. In fact, for some of the circuits the Hamming weight of the S-box output is the optimal model, for others its a single S-box output bit (LSB, MSB, ...) or a combination of multiple bits. None of the specific models we tested worked well on all circuits. Please note, that transitional models like the Hamming distance between consecutively processed values are not a promising candidate here since the static leakage does not naturally capture transitions. Only the small difference in the leakage of a flip-flop cell between (`D=1, Q=0`) and (`D=1, Q=1`) (or analogous combinations) could cause a correlation between the Hamming distance of values and the measured leakage current. The independence of a leakage model featured by Moments-Correlating DPA is indeed a crucial property for a fair comparison of the vulnerability of the circuits. The attack succeeds in all experiments independent of an explicit model and allows a comparison of the data complexity of the attacks. In fact, whenever the leakage of an intermediate value does not closely resemble one of the classical leakage models like the Hamming weight or distance, but rather a more complex leakage function (common for protected implementations) it is plausible that MCDPA is able to extract more information than classical CPA. This expectation is backed up by our observation that no individual CPA in our tests could outperform the MCDPA with respect to attacks on the TI variants. Our MCDPA results for all circuits are shown in Figure 14. Please note the differences in the number of traces utilized for each of the attacks. In order to enable an easier comparison between the different results we have assembled Table 5, which not only lists the number of traces required for a successful recovery of a sub-key difference and the resulting correlation coefficient, but also puts the data complexity for an attack in relation to the area of the circuit. The only discrepancy between the leakage detection results and the key recovery attacks is that the shuffled variants, in relation to the other implementations, show leakage early and strong in a detection scenario, but are still relatively hard to exploit. Due to the nature of shuffling, a stronger attack could probably be performed when recording the leakage after each clock cycle of a cipher round and thereby building a leakage trace over time (similar to a dynamic power measurement). In that case, integrated DPA attacks could reduce the data complexity for a key recovery (see Section 2). Technically, with unrestricted control over the clock signal (the strongest attacker model in this context), the adversary would be capable of single-stepping through the whole encryption operation and measuring the leaked current after each clock cycle. However, we do not consider such an analysis here in order to keep all attacks identical. Tailoring each attack to the countermeasure under analysis would greatly complicate the comparison.

5 Conclusions and Future Work

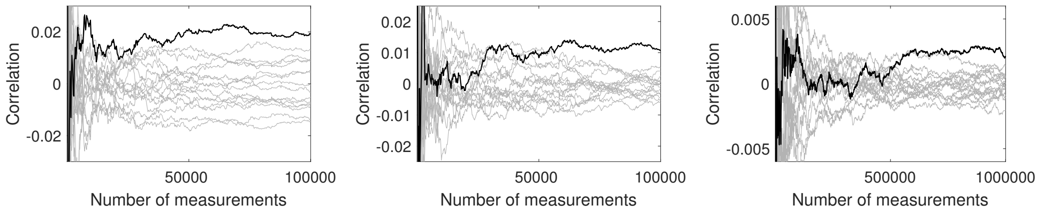
The standby power of CMOS chips silently leaks information to potential adversaries. Several practical case studies have demonstrated this concerning fact throughout the last couple of years. Common side-channel countermeasures used to thwart dynamic leakage



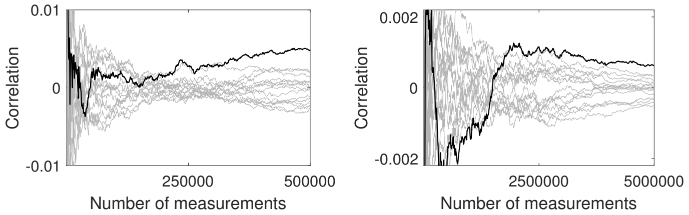
(a) MCDPA^{1st} on HP (b) MCDPA^{1st} on HVT (c) MCDPA^{1st} on RSIS



(d) MCDPA^{1st} on SDRL (e) MCDPA^{1st} on QuadSeal (f) MCDPA^{1st} on ELB



(g) MCDPA^{3rd} on TI HP (h) MCDPA^{3rd} on TI HVT (i) MCDPA^{3rd} on TI RSIS



(j) MCDPA^{3rd} on TI SDRL (k) MCDPA^{3rd} on TI ELB

Figure 14: MCDPA attacks on all countermeasures at 90 °C, 1.35 V (50% over-voltage). MCDPA^{1st} = first-order MCDPA; MCDPA^{3rd} = third-order MCDPA.

Table 5: Data complexities and correlation coefficients for all MCDPA attacks. Data complexities given as absolute values (DC MCDPA) and per gate equivalents (DC / GE).

PRESENT Core	Area [GE]	DC MCDPA	DC / GE	Correlation Coefficient
HP	2 535.00	< 100	< 0.039	0.3258
HVT	2 406.67	200	0.083	0.2734
RSIS	2 613.00	15 000	5.741	0.04069
SDRL	10 789.33	8 800	0.816	0.02907
QuadSeal	12 636.33	67 000	5.302	0.007471
ELB	20 207.00	120 000	5.939	0.006618
TI + HP	7 233.33	23 600	3.263	0.01913
TI + HVT	6 982.67	53 000	7.590	0.01070
TI + RSIS	9 856.33	596 000	**60.469**	0.002144
TI + SDRL	27 907.33	320 000	11.467	0.004860
TI + ELB	58 442.33	**2 930 000**	50.135	**0.0006170**

attacks have shown to be of limited effectiveness against this threat. Thus, specialized countermeasures based on the principles and characteristics of the static power consumption of CMOS devices need to be developed and tested. Practical experiments are especially vital in this process as simulation results often do not sufficiently model all mechanisms that play into the vulnerability of a device. In this work we tried to make a first step in that direction by implementing and evaluating a set of countermeasures consisting of both, previously proposed techniques from the literature and novel ideas, on a 28 nanometer CMOS chip. Our experiments have partially been performed under extreme environmental conditions (90 °C and 50% over-voltage) to figuratively squeeze the information out of our target device. The result of that analysis is that none of the tested countermeasures could withstand attacks with 3 000 000 traces and more than the half of the countermeasure-protected circuits allow extraction of sub-keys with less than 100 000 traces. The strongest protection was achieved by a combination of exhaustive balancing and provably secure hardware masking. However, this combined countermeasure increases the circuit size by a factor of 23, the critical path by a factor of 4, the energy consumption by a factor of 14 and was still susceptible to attacks. This result also speaks to the limits of balancing techniques in general, since even exhaustively balanced circuits are not sufficiently balanced to avoid key extraction. Purely algorithmic approaches, like a combination of masking and shuffling achieve a better cost efficiency, but exhibit a much higher leakage in a detection scenario which may become problematic for device certification. In summary, it seems that existing countermeasures, even rather expensive ones, can only increase the data complexity of static power attacks to a certain extent. The quest for better solutions has to continue.

Future Work. From our point of view, masking schemes which avoid univariate leakage altogether could potentially provide a high level of resistance against SPSCA adversaries. However, that is conceptually difficult to realize since univariate leakage with respect to static power adversaries is much more inclusive than univariate leakage with respect to dynamic power adversaries. A static power adversary can virtually see the cumulative leakage of any gate in a circuit in a single snapshot and not only the leakage of gates that switch simultaneously. Yet, thinking about approaches in this direction may be worthwhile.

Acknowledgments

The work described in this paper has been supported in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and through the project 271752544 "NaSCA: Nano-Scale Side-Channel Analysis".

References

[ABD+14] Massimo Alioto, Simone Bongiovanni, Milena Djukanovic, Giuseppe Scotti, and Alessandro Trifiletti. Effectiveness of leakage power analysis attacks on dpa-resistant logic styles under process variations. *IEEE Trans. Circuits Syst. I Regul. Pap.*, 61-I(2):429–442, 2014.

[AE03] Mohab Anis and Mohamed Elmasry. *Multi-Threshold CMOS Digital Circuits: Managing Leakage Power.* Springer, 2003.

[AO14] Zia Abbas and Mauro Olivieri. Impact of technology scaling on leakage power in nano-scale bulk CMOS digital standard cells. *Microelectronics Journal*, 45(2):179–195, 2014.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

[BCS+17] Davide Bellizia, Danilo Cellucci, Valerio Di Stefano, Giuseppe Scotti, and Alessandro Trifiletti. Novel measurements setup for attacks exploiting static power using DC pico-ammeter. In *2017 European Conference on Circuit Theory and Design, ECCTD 2017, Catania, Italy, September 4-6, 2017*, pages 1–4. IEEE, 2017.

[BDF+17] Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel implementations of masking schemes and the bounded moment leakage model. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 535–566, 2017.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

[CGLS20] Gaëtan Cassiers, Benjamin Grégoire, Itamar Levi, and François-Xavier Standaert. Hardware private circuits: From trivial composition to full verification. *IACR Cryptol. ePrint Arch.*, 2020:185, 2020.

[CRB+16] Thomas De Cnudde, Oscar Reparaz, Begül Bilgin, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Masking AES with d+1 shares in hardware. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 194–212. Springer, 2016.

[CS20] Gaëtan Cassiers and François-Xavier Standaert. Trivially and efficiently composing masked gadgets with probe isolating non-interference. *IEEE Trans. Inf. Forensics Secur.*, 15:2542–2555, 2020.

[CS21] Gaëtan Cassiers and François-Xavier Standaert. Provably secure hardware masking in the transition- and glitch-robust probing model: Better safe than sorry. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(2):136–158, 2021.

[DGS+11] Milena Djukanovic, Luca Giancane, Giuseppe Scotti, Alessandro Trifiletti, and Massimo Alioto. Leakage power analysis attacks: Effectiveness on DPA resistant logic styles under process variations. In *International Symposium on Circuits and Systems (ISCAS 2011), May 15-19 2011, Rio de Janeiro, Brazil*, pages 2043–2046. IEEE, 2011.

[Dig] Digilent. Cmod a7 reference manual. https://reference.digilentinc.com/reference/programmable-logic/cmod-a7/reference-manual. Accessed: 15.01.2021.

[DR98] Joan Daemen and Vincent Rijmen. The block cipher rijndael. In Jean-Jacques Quisquater and Bruce Schneier, editors, *Smart Card Research and Applications, This International Conference, CARDIS '98, Louvain-la-Neuve, Belgium, September 14-16, 1998, Proceedings*, volume 1820 of *Lecture Notes in Computer Science*, pages 277–284. Springer, 1998.

[FGP+18] Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and François-Xavier Standaert. Composable masking schemes in the presence of physical defaults & the robust probing model. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):89–120, 2018.

[FMM20] Bijan Fadaeinia, Thorben Moos, and Amir Moradi. BSPL: balanced static power logic. *IACR Cryptol. ePrint Arch.*, 2020:558, 2020.

[GIB18] Hannes Groß, Rinat Iusupov, and Roderick Bloem. Generic low-latency masking in hardware. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):1–21, 2018.

[GM17] Hannes Groß and Stefan Mangard. Reconciling d+1 masking in hardware and software. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 115–136. Springer, 2017.

[GM18] Hannes Groß and Stefan Mangard. A unified masking approach. *J. Cryptographic Engineering*, 8(2):109–124, 2018.

[GMK16] Hannes Groß, Stefan Mangard, and Thomas Korak. Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. In Begül Bilgin, Svetla Nikova, and Vincent Rijmen, editors, *Proceedings of the ACM Workshop on Theory of Implementation Security, TISCCS 2016 Vienna, Austria, October, 2016*, page 3. ACM, 2016.

[GMK17] Hannes Groß, Stefan Mangard, and Thomas Korak. An efficient side-channel protected AES implementation with arbitrary protection order. In Helena Handschuh, editor, *Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings*, volume 10159 of *Lecture Notes in Computer Science*, pages 95–112. Springer, 2017.

[GSST07] Jacopo Giorgetti, Giuseppe Scotti, Andrea Simonetti, and Alessandro Trifiletti. Analysis of data dependence of leakage current in cmos cryptographic hardware. In *Proceedings of the 17th ACM Great Lakes Symposium on VLSI*, GLSVLSI '07, page 78–83, New York, NY, USA, 2007. Association for Computing Machinery.

[HMY13] Basel Halak, Julian P. Murphy, and Alexandre Yakovlev. Power balanced circuits for leakage-power-attacks resilient design. *SAI*, pages 1178–1183, July 2013.

[HOM06] Christoph Herbst, Elisabeth Oswald, and Stefan Mangard. An AES smart card implementation resistant to power analysis attacks. In Jianying Zhou, Moti Yung, and Feng Bao, editors, *Applied Cryptography and Network Security, 4th International Conference, ACNS 2006, Singapore, June 6-9, 2006, Proceedings*, volume 3989 of *Lecture Notes in Computer Science*, pages 239–252, 2006.

[JIA$^+$15] Darshana Jayasinghe, Aleksandar Ignjatovic, Jude Angelo Ambrose, Roshan G. Ragel, and Sri Parameswaran. Quadseal: Quadruple algorithmic symmetrizing countermeasure against power based side-channel attacks. In *CASES*, pages 21–30, 2015.

[Kei] Tektronix Keithley. 2450 sourcemeter smu instrument datasheet. `https://de.tek.com/datasheet/smu-2400-graphical-sourcemeter/model-2450-touchscreen-source-measure-unit-smu-instrument-`. Accessed: 15.01.2021.

[KMM19] Naghmeh Karimi, Thorben Moos, and Amir Moradi. Exploring the effect of device aging on static power analysis attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):233–256, 2019.

[MMP11] Amir Moradi, Oliver Mischke, and Christof Paar. Practical evaluation of DPA countermeasures on reconfigurable hardware. In *HOST 2011, Proceedings of the 2011 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 5-6 June 2011, San Diego, California, USA*, pages 154–160. IEEE Computer Society, 2011.

[MMR17] Thorben Moos, Amir Moradi, and Bastian Richter. Static power side-channel analysis of a threshold implementation prototype chip. In David Atienza and Giorgio Di Natale, editors, *Design, Automation & Test in Europe Conference & Exhibition, DATE 2017, Lausanne, Switzerland, March 27-31, 2017*, pages 1324–1329. IEEE, 2017.

[MMR20] Thorben Moos, Amir Moradi, and Bastian Richter. Static power side-channel analysis - an investigation of measurement factors. *IEEE Trans. VLSI Syst.*, 28(2):376–389, 2020.

[MMSS19] Thorben Moos, Amir Moradi, Tobias Schneider, and François-Xavier Standaert. Glitch-resistant masking revisited or why proofs in the robust probing model are needed. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):256–292, 2019.

[Moo19] Thorben Moos. Static power SCA of sub-100 nm CMOS asics and the insecurity of masking schemes in low-noise environments. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):202–232, 2019.

[Moo20] Thorben Moos. Unrolled cryptography on silicon A physical security analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):416–442, 2020.

[Mor14] Amir Moradi. Side-channel leakage through static power - should we care about in practice? In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 562–579. Springer, 2014.

[MRSS18] Amir Moradi, Bastian Richter, Tobias Schneider, and François-Xavier Standaert. Leakage detection with the x2-test. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):209–237, 2018.

[MS16] Amir Moradi and François-Xavier Standaert. Moments-correlating DPA. In Begül Bilgin, Svetla Nikova, and Vincent Rijmen, editors, *Proceedings of the ACM Workshop on Theory of Implementation Security, TIS@CCS 2016 Vienna, Austria, October, 2016*, pages 5–15. ACM, 2016.

[NRR06] Svetla Nikova, Christian Rechberger, and Vincent Rijmen. Threshold implementations against side-channel attacks and glitches. In Peng Ning, Sihan Qing, and Ninghui Li, editors, *Information and Communications Security, 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006, Proceedings*, volume 4307 of *Lecture Notes in Computer Science*, pages 529–545. Springer, 2006.

[NYH13] Nianhao Zhu, Yujie Zhou, and Hongming Liu. Counteracting leakage power analysis attack using random ring oscillators. In *Conference on Sensor Network Security Technology and Privacy Communication System*, pages 74–77, 2013.

[PMK$^+$11] Axel Poschmann, Amir Moradi, Khoongming Khoo, Chu-Wee Lim, Huaxiong Wang, and San Ling. Side-channel resistant crypto for less than 2, 300 GE. *J. Cryptol.*, 24(2):322–345, 2011.

[PR16] C. Padmini and J. V. R. Ravindra. Calpan: Countermeasure against leakage power analysis attack by normalized ddpl. In *ICCPCT*, pages 1–7, 2016.

[PSKM15] Santos Merino Del Pozo, François-Xavier Standaert, Dina Kamel, and Amir Moradi. Side-channel attacks from static power: when should we care? In Wolfgang Nebel and David Atienza, editors, *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, DATE 2015, Grenoble, France, March 9-13, 2015*, pages 145–150. ACM, 2015.

[RBN$^+$15] Oscar Reparaz, Begül Bilgin, Svetla Nikova, Benedikt Gierlichs, and Ingrid Verbauwhede. Consolidating masking schemes. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 764–783. Springer, 2015.

[RPD09] Matthieu Rivain, Emmanuel Prouff, and Julien Doget. Higher-order masking and shuffling for software implementations of block ciphers. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September*

6-9, 2009, Proceedings, volume 5747 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2009.

[SM15] Tobias Schneider and Amir Moradi. Leakage assessment methodology - A clear roadmap for side-channel evaluations. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.

[SM21] Aein Rezaei Shahmirzadi and Amir Moradi. Re-consolidating first-order masking schemes nullifying fresh randomness. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(1):305–342, 2021.

[VMKS12] Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. Shuffling against side-channel attacks: A comprehensive study with cautionary note. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 740–757. Springer, 2012.

[YK17a] Weize Yu and Selçuk Köse. False key-controlled aggressive voltage scaling: A countermeasure against LPA attacks. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 36(12):2149–2153, 2017.

[YK17b] Weize Yu and Selçuk Köse. Security-adaptive voltage conversion as a lightweight countermeasure against LPA attacks. *IEEE Trans. on VLSI*, 25(7):2183–2187, 2017.

[YW18] Weize Yu and Yiming Wen. Leakage power analysis (LPA) attack in breakdown mode and countermeasure. In *SOCC*, pages 102–105, 2018.

[ZZL13] Nian-Hao Zhu, Yu-Jie Zhou, and Hong-Ming Liu. Employing symmetric dual-rail logic to thwart LPA attack. *IEEE Embed. Syst. Lett.*, 5(4):61–64, 2013.

[ZZL14] Nian-hao Zhu, Yu-jie Zhou, and Hong-ming Liu. A standard cell-based leakage power analysis attack countermeasure using symmetric dual-rail logic. *Journal of Shanghai Jiaotong University (Science)*, 19(2):169–172, 2014.

Chapter 4

Evaluation of Masked Implementations

In this chapter we introduce the peer-reviewed publications accumulated in this thesis with relation to the evaluation of masked implementations. In total, this chapter covers one paper published at the International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE) and two papers in the IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES).

Contents of this Chapter

4.1 On the Easiness of Turning Higher-Order Leakages into First-Order . . . 185
4.2 Glitch-Resistant Masking Revisited 205
4.3 Deep Learning Leakage Assessment 245

4.1 On the Easiness of Turning Higher-Order Leakages into First-Order

Publication Data

Thorben Moos and Amir Moradi. On the easiness of turning higher-order leakages into first-order. In Sylvain Guilley, editor, *Constructive Side-Channel Analysis and Secure Design - 8th International Workshop, COSADE 2017, Paris, France, April 13-14, 2017, Revised Selected Papers*, volume 10348 of *Lecture Notes in Computer Science*, pages 153–170. Springer, 2017

The acceptance rate at the International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE) 2017 was **53,3%** [Accc].

Content This work presents an alternative technique to analyze higher-order leakages of masked implementations of cryptographic algorithms. The idea is based on distinguishing first-order moments of carefully chosen slices of the acquired leakage distributions instead of distinguishing the full distributions based on the smallest informative statistical moment. It is demonstrated that this technique can outperform classical distinguishers in simulations as well as practical experiments.

Contribution The author of this thesis is a principal author of this publication. In particular, all practical experiments have been conducted and evaluated by the author of this thesis, who also contributed significantly to the writing and the presentation of the results. The author would like to thank the co-author for his substantial contributions to the simulation results presented in this work.

On the Easiness of Turning Higher-Order Leakages into First-Order

Thorben Moos and Amir Moradi

Horst Görtz Institute for IT Security, Ruhr-Universität Bochum, Bochum, Germany
`{firstname.lastname}@rub.de`

Abstract. Applying random and uniform masks to the processed intermediate values of cryptographic algorithms is arguably the most common countermeasure to thwart side-channel analysis attacks. So-called masking schemes exist in various shapes but are mostly used to prevent side-channel leakages up to a certain statistical order. Thus, to learn any information about the key-involving computations a side-channel adversary has to estimate the higher-order statistical moments of the leakage distributions. However, the complexity of this approach increases exponentially with the statistical order to be estimated and the precision of the estimation suffers from an enormous sensitivity to the noise level. In this work we present an alternative procedure to exploit higher-order leakages which captivates by its simplicity and effectiveness. Our approach, which focuses on (but is not limited to) univariate leakages of hardware masking schemes, is based on categorizing the power traces according to the distribution of leakage points. In particular, at each sample point an individual subset of traces is considered to mount ordinary first-order attacks. We present the theoretical concept of our approach based on simulation traces and examine its efficiency on noisy real-world measurements taken from a first-order secure threshold implementation of the block cipher PRESENT-80, implemented on a 150nm CMOS ASIC prototype chip. Our analyses verify that the proposed technique is indeed a worthy alternative to conventional higher-order attacks and suggest that it might be able to relax the sensitivity of higher-order evaluations to the noise level.

1 Introduction

It has become a general knowledge that implementations of cryptographic algorithms are in danger of being attacked by means of side-channel analysis (SCA) key-recovery attacks, if dedicated countermeasures have not (or incorrectly) been integrated. Amongst the known and common SCA countermeasures, *masking* is by far the most-widely studied scheme and has interested both academia and industry. Its underlying sound proofs and theoretical foundation should be named among the reasons for such a popularity. Except particular constructions (e.g., [7, 12]), the security of masking schemes is based on the uniformity of the masks. More precisely, in an $(s+1)$-sharing construction, which is called s-order masking, for a particular x each $(x_1, \ldots, x_{s+1})$ with $x = \bigoplus_{i=1}^{s+1} x_i$ should occur equally likely[1]. Otherwise, it can be pretended that the randomness source is biased, which potentially leads to exploitable leakage.

With respect to the adversary model, security of masking schemes is evaluated based on two different models: *i*) probing model [10], and *ii*) bounded moment model [2]. The former one is primarily used for security proofs and more conservative than the later one, which is usually applied in practical evaluations. Our focus is mainly on the *bounded moment* model, and we call a device *without first-order leakage* if the leakages associated to two different given sets of operands x and y (of the same operation[2]) are not distinguishable[3] from each other through average, i.e., first-order statistical moment. Similarly the leakages should **further** not be distinguishable through variance, i.e., second-order centered moment, for second-order security, and likewise for higher orders. Optionally, the described setting can be incorporated by a pre-processing step, which combines different leakage points. Compared to *univariate* settings, where the combination of leakage points is not required, in a *multi-variate* scenario two (or more) different leakage points are combined prior to evaluation/attack (see [14] for more details).

In short, in order to attack an s-order masked implementation, multi-variate $(s+1)$-order statistical moments should be observed if the operations are serially performed on the shares (i.e., a typical software implementation with sequential nature). On the other hand, in case of a hardware implementation usually univariate $(s+1)$-order statistical moments are observed due to the inherent parallel processing fashion. It is noteworthy that the complexity of higher-order evaluations increases exponentially with s. Further, estimation of higher-order statistical moments becomes extremely hard in practice when the leakages are sufficiently noisy [22].

Instead of a conventional higher-order attack, we present in this work a trick that converts higher-order leakages to the first order and exploits them for key recovery. The focus of our scheme is **univariate** higher-order leakages, i.e., mainly targeting masked hardware implementations. It is essentially based on the principle of pruning the traces according to the distribution of leakage points. Its detailed expression is given in Section 3. Indeed, a similar approach has initially been considered in [24], to exploit the leakage of a masked dual-rail logic style (MDPL) [20]. We review the relevant state of the art in Section 2. Compared to a classical higher-order attack (e.g., mean-free square as an optimal second-order univariate attack) our scheme can be more efficient in particular cases. More precisely, it can exploit the leakage and recover the key while the classical higher-order attacks fail. As a case study, given in Section 4, we present practical results based on an ASIC prototype chip of a provably first-order secure threshold implementation (TI) [17] of the block cipher PRESENT [4].

---

[1] In case of Boolean masking.

[2] For example, two different plaintexts of an AES encryption with a fixed key.

[3] t-test can be used to detect the distinguishability [25].

2 State of the Art

For the majority of masking schemes it is a mandatory requirement that the masks are drawn from a uniform distribution. If this distribution is not uniform, but rather stems from a biased randomness source, vital security claims are not met and exploitable first-order leakage can emerge. Thus, an adversary might be interested in compromising the security of masked implementations deliberately by forcing a bias into the masks that conceal key-dependent intermediate values. One way of achieving this goal is to attack the randomness source directly by means of fault attacks. Of course, the feasibility of this approach depends highly on the particular implementation that is investigated. Another, more generic strategy, which has mainly been applied to compromise software-based masking schemes on microcontrollers, is to categorize the traces that are recorded in a power analysis attack into groups that only contain a biased subset of all possible masks. Intuitively, such an attack can be performed on a software-based masking scheme by determining a point in the power traces where the mask value is processed and then discarding all traces with a measured power consumption above (or below) a certain threshold at that sample point. Assuming now that the investigated device leaks information about the processed intermediate values by means of the Hamming weight (HW) model (which is a reasonable assumption for microcontrollers, see [13]), one has selected a subset of traces with a probability different from $\frac{1}{2}$ for each mask bit to be 1 (or 0). This allows a better-than-random guess what the mask value would be, e.g. all-one (or all-zero), which enables successful first-order attacks on the reduced set of traces. Hence, without preprocessing the power values in the traces, but only by ignoring a subset of the acquired measurements, one has moved the higher-order leakages to a setting where they can be exploited in the first order. Technically, due to the prior selection of power traces, this is still a higher-order attack, but in fact does not require the estimation of higher-order statistical moments. This kind of attack, which we extend and generalize for a different setting in the following course of this work, is referred to as biased mask attack, e.g. in [13] and [26]. Regardless of the surprisingly simple attack procedure, biased mask attacks have not gained much popularity since multi-variate higher-order attacks, utilizing the higher-order statistical moments of the *full* set of traces, are considered more powerful in the general case. Indeed, the loss of information due to disregarding a subset of the measurements is undeniable. Additionally, some kind of initial profiling has to be performed to find a sample point in the power traces where the mask value is leaked.

The described procedure can not be mapped directly to hardware implementations, because in parallel designs the mask is not processed discretely but usually together with the masked data and a number of further intermediate values at the same time. Consequently, only the cumulative leakage of mask and masked data can be observed in a univariate fashion and is not only buried in electronic noise, like for software implementations, but also in the switching noise originating from the remaining parts of the circuit (see [13]). On the one hand, due to the univariate nature of the leakages, the necessity for a profiling phase is removed, but on the other hand the categorization of the traces based on the leakage of the mask value is much less precise. Nevertheless several attempts have been made to perform biased mask attacks on hardware implementations of gate- and algorithmic-level masking schemes. In [27], such an approach is considered for the first time. It is shown by toggle count simulations of a small test circuit (S-box + key XOR) that categorizing power traces with a simple threshold filter is sufficient to remove the one bit of entropy that is introduced by the use of the logic style Random Switching Logic (RSL). The affiliated work in [24] utilizes gate-level simulations of an AES chip design to show that routing imbalances in the DPA-resistant logic style MDPL [20] can be exploited to estimate the mask bit. Again, this can be used to remove the effect of the masking scheme by performing conventional first-order DPA attacks exclusively on the subset of traces that is obtained through a simple filtering operation. In [8] the authors extend their approach to an algorithmic-level hardware masking scheme for the first time. In accordance to the biased mask attacks on software-based implementations the authors are able to verify that a secure hardware masking scheme can equally be compromised by means of simple first-order distinguishers, when only a subset of the traces is considered. Unfortunately, the article fails to investigate how to select a suitable subset of traces that is most informative for an attack. Even more importantly it is not examined at all whether a first-order attack on their specific (or any other choice of) subset can outperform a univariate second-order attack using the mean-free square on the full set of traces. Finally, none of the listed works on hardware masking schemes verified the described attack procedures with practical measurements, taken from a physical hardware device. To the best of our knowledge, no subsequent work explores any of these data points either.

The last branch of research that can be considered related to our approach uses a subset of power traces to enhance the correlation in CPA [6] attacks in general, without concentrating on protected implementations or circumventing specific countermeasures in particular. These works, presented e.g., in [11] and [19], focus on selecting power traces with a high Signal-to-Noise ratio (SNR). They come to the conclusion that, considering the distribution of power values at the point of interest, especially those traces with a small probability density function value, have the highest SNR. In a simplified phrasing this means that concentrating on the power traces whose value at the point of interest is extraordinarily low or high (leftmost or rightmost slices of the leakage distribution) leads to the best correlation for the correct key candidate.

3 Underlying Approach

In this section we introduce and define our novel approach to exploit higher-order leakages. For the sake of simplicity, let us focus on a single sample point of side-channel leakages. The main idea is to observe the distribution of the **univariate** leakages, categorize them into e.g., two non-overlapping parts, and then perform the attack(s) on each part independently. This indeed is the same concept which has been applied in [11] on *unprotected* implementations with the goal of improving the attacks with respect to the required number of traces (see Section 2). However, we employ more-or-less the same technique to exploit higher-order leakages. Let us express the underlying concept with simulation results. Suppose that the leakage of a device under test (DUT) can be represented by a noisy Hamming weight (HW) model as

$$l(x) = HW(x) + \mathcal{N}(\mu, \delta^2),$$

with mean $\mu = 0$ and standard deviation δ. Further, suppose that the intermediate values of the DUT are masked following the concept of first-order Boolean masking. Hence, every value x is represented by (x_m, m) with $x_m = x \oplus m$ and m being a random mask with uniform distribution. In a univariate setting, the leakage of the DUT associated to x is represented by

$$l(x_m) + l(m) = HW(x_m) + HW(m) + \mathcal{N}(0, \delta^2).$$

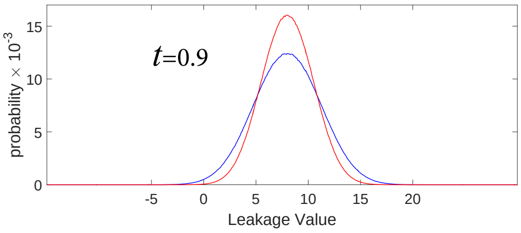
If we simulate $1,000,000$ times the leakage for two different $x \in \{0,1\}^8$ values and a particular $\delta = 2$, two different distributions are observed, that are depicted in Figure 1(a). These two distributions are not distinguishable from each other through their means, i.e., a first-order distinguisher would not be able to differentiate them. Along the same lines, t statistics of a Welch's t-test would give a low-confidence result as well, i.e., t being smaller than 4.5.

However, if we consider only those leakages which are less than a threshold, see Figure 1(b), the leakages are distinguishable from each other through their means. For example, in this case the t statistics yields the value 133, i.e., high confidence of a first-order distinguisher. The threshold in this example has been defined in such a way that 20% of the leakages are below the threshold and the remaining 80% above. As shown in Figure 1(c) to Figure 1(e), considering the upper 80%, lower 80% or upper 20% leakages would lead to distinguishability through means as well. However, in case of Figure 1(f) and Figure 1(g) when the middle part or the side parts of the distributions are considered, the mean does not reveal any distinguishability. This is indeed due to the symmetric form of the original distributions shown in Figure 1(a).

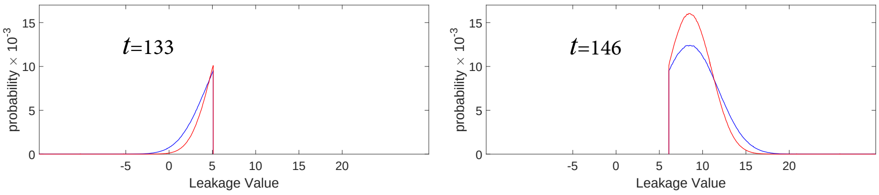
We should highlight that these observations are not limited to first-order masking. As an example, we repeated the same simulation under second-order Boolean masking with univariate leakage

$$l(x_m) + l(m_1) + l(m_2) = HW(x_m) + HW(m_1) + HW(m_2) + \mathcal{N}(0, \delta^2),$$

where $x_m = x \oplus m_1 \oplus m_2$ and the uniform distribution for m_1 and m_2. The distributions and the t statistics as distinguishability measure after classifying the leakages based on a particular threshold are shown in Figure 2. Following the concept of second-order masking, the distributions are distinguishable only through their skewness (see Figure 2(a)). However, by categorizing them based on a 20% threshold (either above or below the threshold) the means reveal the difference between the distributions. Interestingly, the symmetric forms, i.e.,

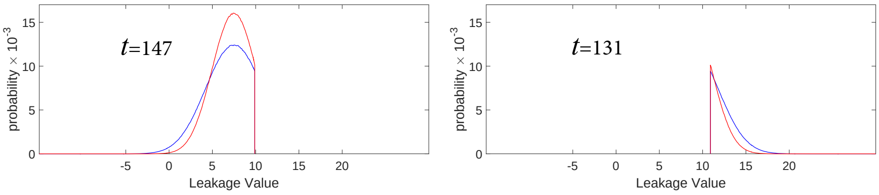


(a) original



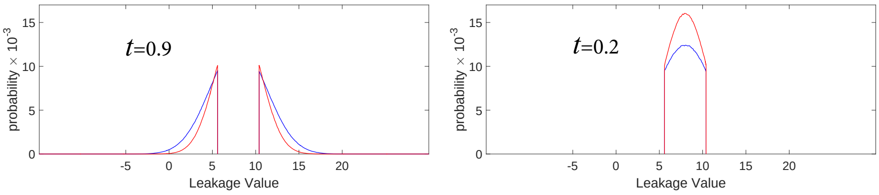
(b) lower 20%

(c) upper 80%



(d) lower 80%

(e) upper 20%



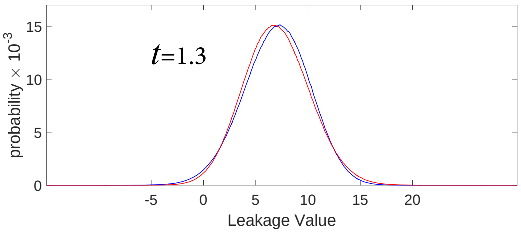
(f) 20% sides

(g) without 20% sides

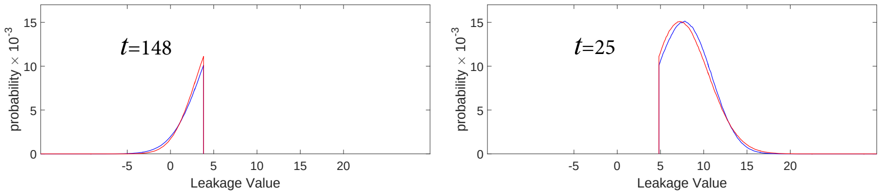
Fig. 1. Simulated leakage distributions of two different values represented by first-order masking, t represents the statistics of the t-test.

middle part or the sides (Figure 2(f) and Figure 2(g)), also lead to high-evidence first-order distinguishability.

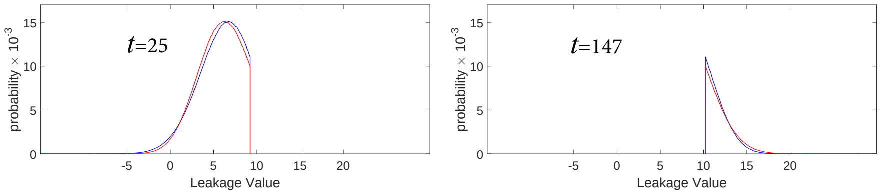
When evaluating the effectiveness of this approach it is important to know for which threshold value the attack performs best. To identify the optimal threshold, we conducted another simulation based on first-order masking. We have randomly selected a vector of n elements as $X : (x^1, \ldots, x^n)$, where $x^i \in \{0,1\}^8$. Then, by two separate uniformly-distributed $n-$element mask vectors M_1 and M_2 we formed $X_{M_1} = (x^1_{m_1}, \ldots, x^n_{m_1})$, where $x^i_{m_1} = x^i \oplus m^i_1$ (resp. for



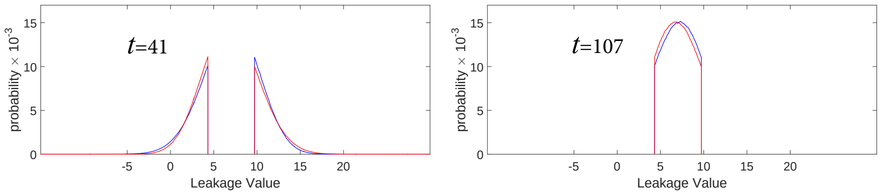
(a) original



(b) lower 20% (c) upper 80%



(d) lower 80% (e) upper 20%



(f) 20% sides (g) without 20% sides

Fig. 2. Simulated leakage distributions of two different values represented by second-order masking, t represents the statistics of the t-test.

X_{M_2}). Following the univariate noisy Hamming weight leakage model, we formed two leakage vectors $L_1 : (l_1^1, \ldots, l_1^n)$ and $L_2 : (l_2^1, \ldots, l_2^n)$ in such a way that for example

$$l_1^i = HW(x_{m_1}^i) + HW(m_1^i) + \mathcal{N}(0, \delta^2).$$

Following the concept of Moments-Correlating DPA (MC-DPA) [15], we first formed a model $\dot{L}_1 : (\dot{l}_1^1, \ldots, \dot{l}_1^n)$ as

$$\dot{l}_1^i = \mu\left(\{\forall l_1^j | x^j = x^i\}\right),$$

and finally estimated the correlation $\rho(\dot{L}_1, L_2)$ as the first-order correlation. For the second-order correlation, we first formed a model $\ddot{L}_1 : (\ddot{l}_1^1, \ldots, \ddot{l}_1^n)$ as

$$\ddot{l}_1^i = \delta^2 \left(\{\forall l_1^j | x^j = x^i\} \right),$$

and respectively made L'_2 as mean-free square of L_2 as

$$l_2'^i = \left(l_2^i - \mu \left(\{\forall l_2^j | x^j = x^i\} \right) \right)^2.$$

Hence, correlation $\rho(\ddot{L}_1, L'_2)$ can be estimated as the second-order correlation. On the other hand, we selected a part of L_1 and L_2 based on a threshold and following the above procedure estimated the first-order correlation. We conducted this simulation for $n = 1,000,000$ and several values for noise standard deviation δ. For each setting, we examined different thresholds to split the leakages. More precisely, from lower 5% up to lower 50% and from upper 50% to upper 95%, each with steps of 5%. The results are shown in Figure 3(a).

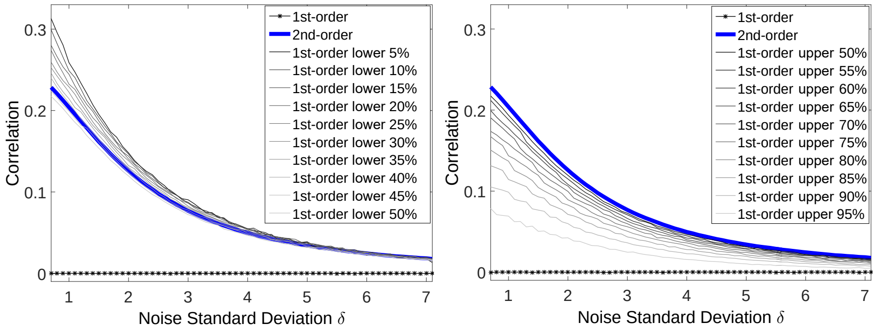
As shown by the graphics, none of the cases, where over 50% of the leakages are considered, can compete with the optimal second-order distinguisher. In contrast, when less than 50% of the leakages are considered, the underlying approach outperforms the second-order one. Further, by increasing the noise level they all become similar and close to the second-order distinguisher. It is noteworthy that due to the symmetry of the distributions in case of this simulation (i.e., first-order masking) the results of the other cases, i.e., upper $< 50\%$ and lower $> 50\%$, are not shown.

This simulation has been repeated following the above-explained univariate leakage of second-order Boolean masking. Figure 3(b) shows the corresponding results. As expected, the first- and second-order distinguishers would not reveal any dependency. Interestingly, the underlying approach extremely outperforms the optimal third-order distinguisher, and even by increasing the noise standard deviation it still performs better.

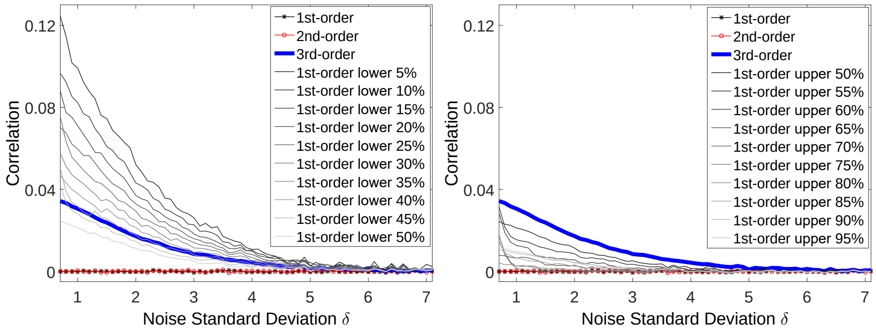
We should note that any other distinguisher, where instead of any particular statistical moment the distribution of the leakages are considered, would also differentiate the univariate higher-order leakages. But, these distinguishers (e.g., MIA [9]) would need to predict the probability distributions, e.g., by histogram where the number of bins and the size of each bin play an important role for the efficiency of the distinguisher, alternatively by Kernel where the important issues include the type of the Kernel function and the associated parameters. The diversity of their results based on the selected parameters can make such distinguishers more complicated or less efficient compared to higher-order attacks. However, in the approach presented here we just consider the distribution obtained based on pure histogram. More precisely, the histogram made by the nature of the SCA measurements (i.e., 256 bins as the result of the 8-bit ADC[4] of the acquisition equipment digital oscilloscope) would suffice to find the threshold for a given percentage, e.g., lower 20%.

---

[4] Analog to Digital Converter.



(a) first-order Boolean masking



(b) second-order Boolean masking

Fig. 3. Correlation (based on MC-DPA), simulated univariate (a) second-order and (b) third-order leakages, comparison between different distinguishers for different threshold values over noise standard deviation.

4 Practical Results

Now that we have presented the theoretical concept of our approach, it is time to evaluate the soundness of the technique based on real-world measurements taken from the physical implementation of a hardware masking scheme. After a description of the target device and the measurement setup we analyze the side-channel leakage of the test chip by means of conventional higher-order attacks, which are based on the estimation of higher-order statistical moments. As a second step we present the results of our novel approach for different threshold values. At the end, both types of attacks are compared in terms of the required number of measurements for a successful key recovery and the convenience of the procedure from an attacker's point of view.

Target. The target platform for our practical evaluations is a 150 nm CMOS ASIC prototype chip. A layered view of the fabricated chip can be seen in Figure 4. The prototype contains 6 different cores and was specifically developed to

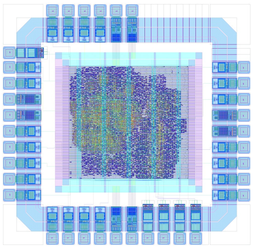


Fig. 4. ASIC prototype with 6 cores in 150 nm CMOS.

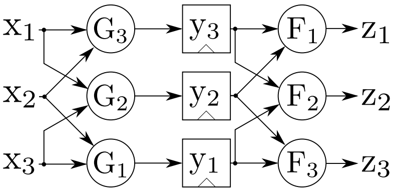


Fig. 5. Threshold implementation of the 4-bit PRESENT S-box with 3 shares.

evaluate the side-channel resistance of state-of-the-art block ciphers and DPA countermeasures in practice. The core of the ASIC that is targeted in the following experiments realizes the block cipher PRESENT-80 under 3-share first-order threshold implementation concept. PRESENT-80 is an ultra-lightweight block cipher (ISO/IEC 29192-2:2012 lightweight cryptography standard) that features a block size of 64 bit as well as a key length of 80 bit and consists of 31 computation rounds [4], whereas threshold implementations have been introduced as an efficient hardware masking scheme in [18].

Concerning hardware implementations of masking schemes, it has historically been a challenging task to ensure that glitches in the combinatorial parts of the circuit do not recombine the shares and thus lead to exploitable leakage. Threshold implementations prevent this issue by adding the so-called non-completeness property to the masked computations [2]. Non-completeness means here that each fully combinatorial circuit must be independent of at least one of the shares. This is achieved by splitting the non-linear parts of a circuit into several shared functions that do not operate on all shares at once, but rather perform only one part of the overall computation that refers to its respective inputs. Accordingly, glitches can never recombine all shares at once, meaning that an adversary is not able to learn any information about the secret from the side-channel leakage of only one of these circuits. Indeed, multiple leakages of multiple combinatorial (sub-) circuits need to be combined to perform a successful (higher-order) attack. Following this concept, which is based on Boolean secret sharing and multi-party computation, the threshold implementation technique can be used to implement non-linear functions of symmetric block ciphers in such a way that provable security against first-order power analysis attacks can be guaranteed, even in the presence of glitches. Higher-order threshold implementations can furthermore be used to conceal the leakages at higher-order statistical moments [3]. A second property that has to be fulfilled when sharing a non-linear function is the uniformity of the outputs. For each unshared input to the non-linear function, each shared output should occur equally likely. In this way the output of the shared functions is still uniformly distributed and a remasking is not required. More precisely during the full execution of a block cipher that is implemented in this masking scheme no fresh masks needs to be fed. The plaintext is split up into the required number of shares at the beginning of the algorithm (see [18]), which implies the generation of two or more plaintext-sized masks, and all further computations are performed on those shares. Compared to conventional masking, the drawback of this method is a higher number of required shares. In particular at least three shares (two masks) are required to realize each non-linear part of a circuit[5]. Additionally the number of shares increases with the degree of the function that needs to be implemented [18]. Hence, larger S-boxes, e.g. 8-bit, are difficult to implement efficiently in this scheme [5]. Nevertheless, for ciphers with small S-boxes, e.g., PRESENT-80, threshold implementation has become the de facto standard for hardware masking [2].

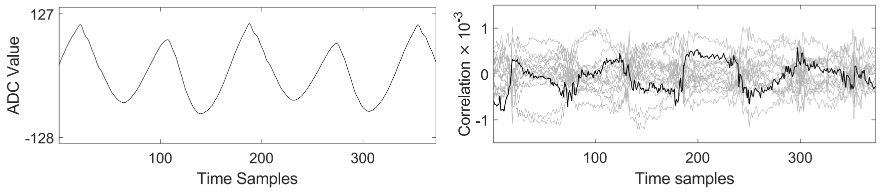
The realization of the PRESENT-80 block cipher as a threshold implementation was introduced in [21]. The authors proposed several implementation profiles with different levels of security. Our targeted ASIC core implements profile 2, which refers to a nibble-serial implementation of the block cipher with a shared data path (with 3 shares) but an unshared key schedule. Hence, one instance of the shared S-box is implemented and the 4-bit nibbles of the cipher state are processed in a pipelined manner. A schematic view of the shared S-box, based on a decomposition to quadratic functions F and G with $S(x) = F\left(G\left(x\right)\right)$, can be seen in Figure 5. Due to the register stage between the G- and the F functions one full cipher-round takes 18 clock cycles[6]. It is noteworthy that although first-order threshold implementation corresponds to Boolean masking with 3 shares it provides only first-order security due to its underlying quadratic functions (i.e., G and F in Figure 5). In other words, this implementation is supposed to exhibit second- and third-order leakages.

Measurement Setup. We performed our measurements on a Side-channel Attack Standard Evaluation Board (SASEBO-R) [1] that was specifically developed to evaluate the side-channel resistance of cryptographic hardware. For this purpose it provides a socket for an ASIC prototype, which is connected by a 16-bit bidirectional data bus as well as a 16-bit address signal to a Xilinx Virtex-II Pro control FPGA, clocked by a 24-MHz oscillator. For the side-channel measurements a Teledyne LeCroy HRO 66zi oscilloscope was used. We collected 5 million measurements for random plaintexts and a fixed key by measuring the voltage drop over a $1\,\Omega$ resistor in the Vdd path, while the ASIC was operated at a frequency of 3 MHz and a supply voltage of 1.8 V. Each of the power traces contains 100,000 sample points recorded at a sampling rate of 500 MS/s with a resolution of 8 bits. Due to a very low amplitude of the signal two × 10 AC amplifiers in series have been employed, resulting in a × 100 gain. Figure 6(a) depicts a sample trace over the two clock cycles that we are referring to in the following course of this analysis. The two random and uniform 64-bit masks that are needed for the initial sharing of the plaintext are generated and delivered by a PRNG (AES-128 in counter mode) on the control FPGA of the SASEBO-R, which in turn is seeded by the PC via UART.

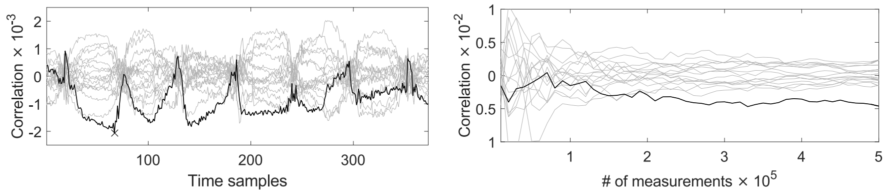
---

[5] Lower number of shares can be achieved at the price of additional fresh masks [23].

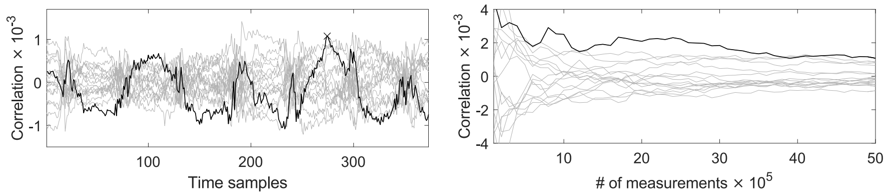
[6] The permutation layer in one separate clock cycle.



(a) sample power trace (b) first-order CPA



(c) second-order CPA (d) correlation trend (2nd-order)



(e) third-order CPA (f) correlation trend (3rd-order)

Fig. 6. Sample power trace and conventional first-, second- and third-order CPA with 5 million measurements using the HW of the G-box output.

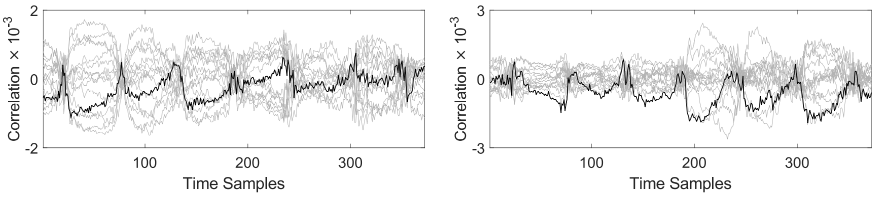
Results of Conventional Attacks. To evaluate the effectiveness of the presented approach on noisy real-world measurements it is necessary to assess the vulnerability of the underlying hardware masking scheme by means of conventional DPA attacks in a first step. To this end, we performed first-, second- and third-order Correlation Power Analysis (CPA) attacks [6] using the Hamming weight (HW) of the S-box output (which is the same as the output of the F function in Figure 5). This did not lead to a successful recovery of any key nibble. Hence, we performed the same attack using the HW of the output of the G function (i.e., the value of the intermediate register) and obtained the results which are depicted in Figure 6. All results are plotted over the two clock cycles that leak the targeted intermediate value. This is on the one hand the clock cycle in which the G-boxes are evaluated in parallel and on the other hand the succeeding clock cycle where the outputs of the F-boxes are computed based on the G-box outputs. As expected the first-order attack is not successful. The second-order CPA, on the other hand, reveals the correct key nibble, but only by a slight margin. The third-order attack does not succeed since the correct key candidate does not lead to the overall highest correlation during the targeted two clock cycles. In particular several ghost peaks with a higher correlation can be identified. For both, the second- and the third-order CPA, we have plotted the evolution of the correlation for the most leaking time sample (marked by a cross in Figure 6(c) and Figure 6(e)). In this way we obtain a quantitative measure to express how many traces are required to reveal the higher-order leakages. For the second-order attack at least 200,000 traces are required, whereas for the third-order attack even with the entire 5,000,000 measurements the correct candidate might not be detectable. We observed the same results targeting several other key nibbles. Indeed, it can be concluded that our measurements are sufficiently noisy to serve as a suitable data source for our further analysis.

The efficiency of CPA attacks relies on the linear dependency between the hypothetical power model (here HW of the G-box output) and the actual leakage of the device. Alternatively, Moments-Correlating DPA (MCDPA) [15] can relax such a necessity at the price of (usually) requiring more traces compared to a corresponding CPA with a suitable power model. To examine whether a collision setting can improve the number of required measurements here, which would indicate an imperfect choice of the leakage model in the CPA evaluations, we performed an MCDPA on the same traces. Hereby, the leakage of one S-box is used to build a model which is then used in an attack on another S-box, leading to a recovery of the linear difference between the corresponding key nibbles. In our case the same hardware instance of the S-box is used for both steps, which ensures a similar leakage model. Figure 7 shows the results indicating that only the third-order MCDPA is able to reveal the correct key difference with 5 million measurements [7]. And even this is only true when exclusively the second leaking clock cycle is considered. Otherwise, there are again ghost peaks with a higher correlation. Nevertheless, 1.5 million measurements are required to exploit the third-order leakage. This result enhances our confidence that the Hamming weight of the output of the G-box is a suitable leakage model for our target.

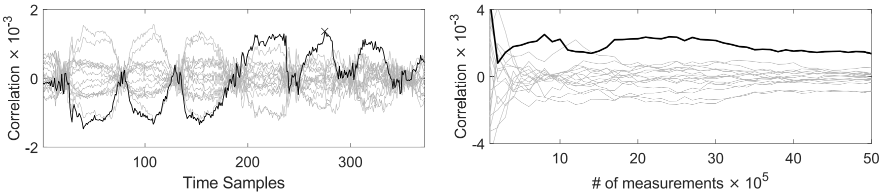
Results of Our Novel Approach. Hereafter, we concentrate on applying our novel approach (expressed in Section 3) on the same traces. In this regard we first obtained a histogram for each sample point using all 5,000,000 traces. The histograms – as given before – have been made by 256 bins, i.e., the full range of signed 8-bit integers -128 to 127 which reflect the sampled power consumption values unaltered (direct result of the oscilloscope ADC). Therefore, for each given $x\%$ threshold we obtain a threshold trace. This trace contains a threshold value for each sample point individually in such a way that $x\%$ of the traces have a value smaller than the threshold at that sample point and $(100 - x)\%$ have a higher value. As the next step, we conducted the attacks on a subset of traces either as "lower $x\%$" or "upper $(100 - x)\%$". It should be noted that such a separation of traces as well as the attack is performed on each sample point separately. In other words, for each sample point it is individually decided which traces to be considered in the attack.

---

[7] Only positive correlation values indicate a collision in an MCDPA attack.



(a) first-order MCDPA (b) second-order MCDPA

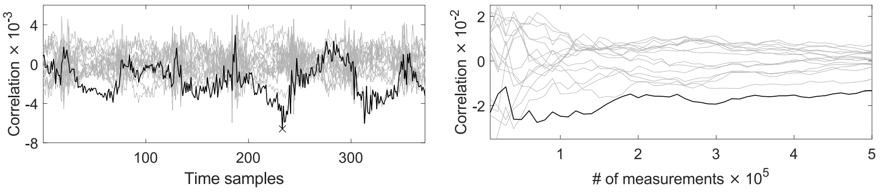


(c) third-order MCDPA (d) correlation trend (3rd-order)

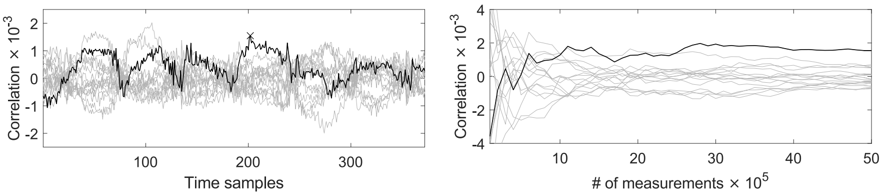
Fig. 7. Conventional first-, second- and third-order MCDPA with 5 million measurements.

We have examined the threshold values between 5% and 95% with intervals of 5%. In Figure 8 we represent the result of the attacks (CPA with HW of the G-box output) for the most successful settings, i.e., 20% and 30% thresholds. Interestingly it can be noted that attacks on subsets with a power consumption below the threshold, i.e., lower 80% and lower 70%, lead to a positive correlation for the correct key candidate, and vise versa for the corresponding upper 20% and upper 30%. This is in fact due to the different biases that are introduced into the three shares by selecting measurements with a power consumption either above or below a certain threshold.

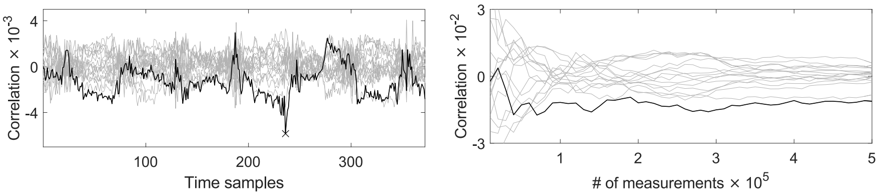
Comparison. When comparing our approach to the corresponding conventional second-order CPA, the value of the highest correlation for the correct key candidate is not very meaningful. Due to the fact that a much smaller number of measurements contributes to the results of our approach the correlation values are usually significantly higher compared to the conventional attacks. Hence we have to rely on the required number of measurements as well as a visual inspection of the results as the only available metrics for a comparison. Regarding the required number of measurements we can refer to Figure 8(b) and Figure 8(f) that only 50,000 and respectively 70,000 measurements are required to reveal the leakage with our approach. It should be noted that these numbers as well as Figure 8(b) and Figure 8(f) reflect the number of traces used to both, find the threshold and perform the attack on. In other words, when it is shown that 50,000 traces are required for a "upper 20%" attack, all 50,000 traces are used to find the threshold. Amongst them, around 50,000×20%=10,000 traces are used in the attack. Hence, compared to the conventional second-order attack, the at-



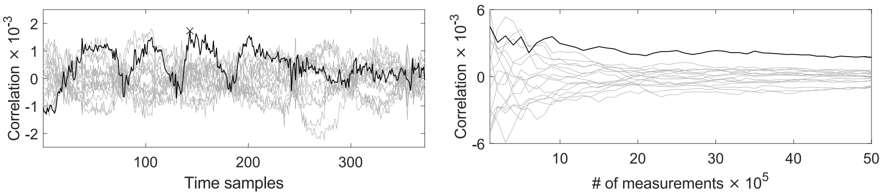
(a) upper 20% first-order CPA (b) correlation trend



(c) lower 80% first-order CPA (d) correlation trend



(e) upper 30% first-order CPA (f) correlation trend



(g) lower 70% first-order CPA (h) correlation trend

Fig. 8. First-order CPA on different slices of the 5 million measurements using the Hamming weight of the G-box output.

tack with "upper 20%" required 4 times less traces altogether and, due to the fact that only a subset is considered, includes 20 times less traces in the actual CPA computations. In accordance to the simulation results (in Section 3) we can see that the attacks on subsets of traces, that include more than 50% of the measurements, are not able to outperform the conventional attack. More precisely, the "lower 80%" and "lower 70%" attacks (Figure 8(d) and Figure 8(h)) need respectively around 2,500,000 and 700,000 traces while the conventional second-order attack requires 200,000 traces.

All of the presented attacks have been repeated for other key nibbles and therefore on other parts of the power traces as well. These experiments revealed that concentrating on the "upper 30%" part (for each sample point individually) was indeed most commonly the best choice, although the particular threshold values vary slightly between different key nibbles. Another tendency that could be observed is that the subsets which have been selected from above a threshold were generally significantly more informative than the subsets below a threshold (independent of being each others counterpart). However, for all targeted key nibbles our approach was able to outperform the conventional second-order attack in terms of the required number of measurements for at least one choice of subset.

5 Conclusions

In this work we have presented and examined an alternative approach to analyze the higher-order leakages of masked hardware implementations. The proposed technique is able to turn higher-order leakages with a simple selection procedure into a setting where they can be exploited by a first-order distinguisher. This does not only remove the necessity to estimate higher-order statistical moments when attacking masking schemes, which becomes exponentially more complex with an increasing statistical order, but it may also be able to relax the sensitivity of higher-order attacks to the noise level. We have presented the theoretical foundation of our approach by means of simulations and carried out several experiments on noisy real-world measurements to back up our claims. Our analyses lead to the conclusion that our approach indeed represents an alternative to conventional higher-order attacks, and even more importantly is able to outperform them in specific settings. In our setup for example a standard first-order CPA on the subset of traces, that contains only the 20% highest power consumption values (individuality at each sample point), is able to exploit the leakage with 4 times less traces than the conventional second-order CPA attack (i.e., by mean-free square). Hence, a significant improvement could be achieved by simply ignoring a specific part of the traces (at each sample point).

It has been given in literature that masking and hiding countermeasures should be combined to achieve a high level of security. In works like [16] hardware masking is implemented by power-equalization schemes to practically complicate higher-order attacks. As a future work, we will investigate the feasibility of the approach introduced here on such implementations. Another interesting approach to explore is whether it is worthwhile to combine the result of the attacks after splitting the traces. More precisely, we have shown the result of the attacks for "upper 20%" and "lower 80%". The question is whether combining these results would lead to a more effective attack.

Acknowledgements

The authors would like to acknowledge Axel Poschmann for the hardware designs and Stefan Heyse for his help on taping out the prototype chip. This work is partly supported by the German Research Foundation (DFG) through the project "NaSCA: Nano-Scale Side-Channel Analysis".

References

1. Side-channel Attack Standard Evaluation Board SASEBO-R Specification – Version 1.0. `http://www.risec.aist.go.jp/project/sasebo/download/SASEBO-R_Spec_Ver1.0_English.pdf`. Research Center for Information Security, National Institute of Advanced Industrial Science and Technology, Japan.
2. Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel Implementations of Masking Schemes and the Bounded Moment Leakage Model. In *EUROCRYPT 2017*, Lecture Notes in Computer Science. Springer, 2017. to appear.
3. Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Higher-Order Threshold Implementations. In *ASIACRYPT 2014*, volume 8874 of *Lecture Notes in Computer Science*, pages 326–343. Springer, 2014.
4. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.
5. Erik Boss, Vincent Grosso, Tim Güneysu, Gregor Leander, Amir Moradi, and Tobias Schneider. Strong 8-bit Sboxes with Efficient Masking in Hardware. In *CHES 2016*, volume 9813 of *Lecture Notes in Computer Science*, pages 171–193. Springer, 2016.
6. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In *CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.
7. Claude Carlet, Jean-Luc Danger, Sylvain Guilley, and Houssem Maghrebi. Leakage Squeezing of Order Two. In *INDOCRYPT 2012*, volume 7668 of *Lecture Notes in Computer Science*, pages 120–139. Springer, 2012.
8. Zhimin Chen and Patrick Schaumont. Slicing Up a Perfect Hardware Masking Scheme. In *HOST 2008*, pages 21–25. IEEE, 2008.
9. Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual Information Analysis. In *CHES 2008*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2008.
10. Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. In *CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.
11. Yongdae Kim, Takeshi Sugawara, Naofumi Homma, Takafumi Aoki, and Akashi Satoh. Biasing power traces to improve correlation in power analysis attacks. In *COSADE 2010*, pages 77–80, 2010.
12. Houssem Maghrebi, Sylvain Guilley, and Jean-Luc Danger. Leakage Squeezing Countermeasure against High-Order Attacks. In *WISTP 2011*, volume 6633 of *Lecture Notes in Computer Science*, pages 208–223. Springer, 2011.
13. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, 2007.
14. Amir Moradi and Oliver Mischke. On the Simplicity of Converting Leakages from Multivariate to Univariate - (Case Study of a Glitch-Resistant Masking Scheme). In *CHES 2013*, volume 8086 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2013.
15. Amir Moradi and François-Xavier Standaert. Moments-Correlating DPA. In *Workshop on Theory of Implementation Security*, TIS '16, pages 5–15. ACM, 2016.

16. Amir Moradi and Alexander Wild. Assessment of Hiding the Higher-Order Leakages in Hardware. In *CHES 2015*, volume 9293 of *Lecture Notes in Computer Science*, pages 453–474. Springer, 2015.
17. Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. *J. Cryptology*, 24(2):292–321, 2011.
18. Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. *J. Cryptology*, 24(2):292–321, 2011.
19. Changhai Ou, Zhu Wang, Degang Sun, Xinping Zhou, Juan Ai, and Na Pang. Enhanced Correlation Power Analysis by Biasing Power Traces. In *ISC 2016*, volume 9866 of *Lecture Notes in Computer Science*, pages 59–72. Springer, 2016.
20. Thomas Popp and Stefan Mangard. Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints. In *CHES 2005*, volume 3659 of *Lecture Notes in Computer Science*, pages 172–186. Springer, 2005.
21. Axel Poschmann, Amir Moradi, Khoongming Khoo, Chu-Wee Lim, Huaxiong Wang, and San Ling. Side-Channel Resistant Crypto for Less than 2,300 GE. *J. Cryptology*, 24(2):322–345, 2011.
22. Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.
23. Oscar Reparaz, Begül Bilgin, Svetla Nikova, Benedikt Gierlichs, and Ingrid Verbauwhede. Consolidating Masking Schemes. In *CRYPTO 2015*, volume 9215 of *Lecture Notes in Computer Science*, pages 764–783. Springer, 2015.
24. Patrick Schaumont and Kris Tiri. Masking and Dual-Rail Logic Don't Add Up. In *CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 95–106. Springer, 2007.
25. Tobias Schneider and Amir Moradi. Leakage Assessment Methodology - A Clear Roadmap for Side-Channel Evaluations. In *CHES 2015*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.
26. Stefan Tillich, Christoph Herbst, and Stefan Mangard. Protecting AES Software Implementations on 32-Bit Processors Against Power Analysis. In *ACNS 2007*, volume 4521 of *Lecture Notes in Computer Science*, pages 141–157. Springer, 2007.
27. Kris Tiri and Patrick Schaumont. Changing the Odds Against Masked Logic. In *SAC 2006*, volume 4356 of *Lecture Notes in Computer Science*, pages 134–146. Springer, 2006.

4.2 Glitch-Resistant Masking Revisited

Publication Data

> Thorben Moos, Amir Moradi, Tobias Schneider, and François-Xavier Standaert. Glitch-resistant masking revisited or why proofs in the robust probing model are needed. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):256–292, 2019

The acceptance rate for Volume 2019 of the IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES) was **19.6%** [Acca].

Content This work analyzes the robust probing security and composability of multiple hardware-based masking schemes proposed in side-channel literature. It is revealed that none of the analyzed masked multiplication gadgets is able to provide local and compositional security in the presence of glitches for arbitrary protection orders. At the very least there is no recipe given for their secure instantiation at higher orders. In fact, a number of local and compositional flaws are exhibited which prevent the secure instantiation of these schemes at higher orders and lead to detectable security order reductions in real-world power measurements in such cases.

Contribution The author of this thesis contributed a significant part of the practical experiments and their corresponding evaluation presented in this publication. He also contributed substantially to the writing of Sections 8 and 9 and Appendix A. The author would like to thank all co-authors for their significant contributions to the success of this work, culminating in the election of this publication as the best paper of TCHES 2019.

Glitch-Resistant Masking Revisited

or Why Proofs in the Robust Probing Model are Needed

Thorben Moos[1], Amir Moradi[1],
Tobias Schneider[2] and François-Xavier Standaert[2]

[1] Horst Görtz Institute for IT Security, Ruhr-Universität Bochum, Germany
[2] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium
[1]firstname.lastname@rub.de, [2]firstname.lastname@uclouvain.be

Abstract. Implementing the masking countermeasure in hardware is a delicate task. Various solutions have been proposed for this purpose over the last years: we focus on Threshold Implementations (TIs), Domain-Oriented Masking (DOM), the Unified Masking Approach (UMA) and Generic Low Latency Masking (GLM). The latter generally come with innovative ideas to cope with physical defaults such as glitches. Yet, and in contrast to the situation in software-oriented masking, these schemes have not been formally proven at arbitrary security orders and their composability properties were left unclear. So far, only a 2-cycle implementation of the seminal masking scheme by Ishai, Sahai and Wagner has been shown secure and composable in the robust probing model – a variation of the probing model aimed to capture physical defaults such as glitches – for any number of shares.

In this paper, we argue that this lack of proofs for TIs, DOM, UMA and GLM makes the interpretation of their security guarantees difficult as the number of shares increases. For this purpose, we first put forward that the higher-order variants of all these schemes are affected by (local or composability) security flaws in the (robust) probing model, due to insufficient refreshing. We then show that composability and robustness against glitches cannot be analyzed independently. We finally detail how these abstract flaws translate into concrete (experimental) attacks, and discuss the additional constraints robust probing security implies on the need of registers. Despite not systematically leading to improved complexities at low security orders, e.g., with respect to the required number of measurements for a successful attack, we argue that these weaknesses provide a case for the need of security proofs in the robust probing model (or a similar abstraction) at higher security orders.

Keywords: Hardware Masking · Glitches Composability · Robust Probing Model · Threshold Implementations · Consolidated Masking Scheme · Domain-Oriented Masking · Unified Masking Approach · Generic Low-Latency Masking

1 Introduction

Masking (aka secret sharing) is one of the most popular countermeasures against side-channel attacks [CJRR99]. Evaluating its security guarantees is known to be non-trivial, especially as the number of shares and claimed security order increase. The latter is confirmed by various security flaws that have been exhibited in early proposals of higher-order masking schemes, which we organize in two categories. First, *local flaws* correspond to cases where a masked gadget (e.g., a multiplication algorithm, a masked S-box, . . .) does not deliver its security guarantees. A typical example of a local flaw is the attack against the higher-order masking scheme of Schramm and Paar [SP06], put forward by Coron et al. [CPR07]. Second, *composability flaws* correspond to cases where the combination of locally secure gadgets leads to additional weaknesses. A typical example of a composability flaw is the attack against the higher-order masking scheme of Rivain and Prouff [RP10] (which describes locally secure gadgets), put forward by Coron et al. [CPRR14].

In order to avoid such security flaws, two main theoretical advances have been introduced in the literature. First, security proofs in the *probing model* of Ishai et al. [ISW03] can be used to analyze the local security of a masked gadget. Second, the notions of *Non-Interference* (NI) and *Strong Non-Interference* (SNI) can be used to capture the compositional security of masked gadgets [BBD+16].

Those theoretical advances are complemented by practical ones exploiting program verification techniques. For example, the work by Barthe et al. describes a tool able to verify the security of a masked implementation up to a certain order [BBD+15]. Other works propose similar but more specialized ideas [EWS14, Rep16].

Furthermore, under some assumptions of sufficiently noisy and independent leakages, security in the (abstract) probing model implies security in the (more concrete) noisy leakage model [PR13], as shown by Duc et al. [DDF14]. Since (under the independence condition only), probing security also implies security in the bounded moment leakage model [BDF+17], which is frequently used to assess the concrete security order of actual implementations [SM16], these results suggest probing security as a useful first step to verify for any masked implementation.

Concretely, this first (abstract) evaluation step of masked implementations can typically rely on two approaches. Either security is claimed for arbitrary orders. In this case, a hand-made proof is required for the masked gadgets considered (and this proof has to guarantee composability in case the target implementation is a full cipher mixing many gadgets). Or security is claimed up to a given order that can be exhaustively analyzed thanks to program verification techniques. To a large extent, all recent results in (what we denote as) software-oriented masking (to be understood as the masking schemes primarily designed for software implementations) follow one of these approaches, leading to easy-to-interpret guarantees. We cite [Cor14, BBP+16] and [BDF+17] as recent examples.

Hardware-oriented masking. In parallel to software-oriented masking, significant efforts have also been devoted to the design of masking gadgets for hardware implementations. In this context, one important additional issue is that physical defaults such as glitches can easily contradict the independence assumption required for secure masking [MPG05]. Since this break of the independence assumption directly leads to devastating attacks [MPO05], the literature then focused on the design of gadgets with better resistance to glitches. A popular illustration of such progresses is the introduction of *Threshold Implementations* (TIs), which showed that a simple algorithmic property (namely, the *non-completeness* property) is sufficient to mitigate the glitch issue [NRS11]. The latter was then successfully applied to many first-order threshold implementations (e.g., [PMK+11, MPL+11, BGN+14b]).

Yet, as in the software case, the generalization from first-order TIs to higher-order TIs proved to be challenging. For example, the first attempt to build a higher-order TI in [BGN+14a] was not successful because of a lack of refreshing leading to a composability flaw [Rep15, RBN+15]. Since then, various papers proposed innovative ways to implement higher-order masking in hardware, mixing engineering intuitions and elements borrowed from the software-oriented masking literature. We mention for example the Consolidated Masking Scheme (CMS) in [RBN+15, CRB+16], the Domain-Oriented Masking (DOM) in [GMK16, GMK16, GMK17], the Unified Masking Approach (UMA) in [GM17, GM18] and the Generic Low Latency Masking (GLM) in [GIB18].

An interpretation issue. Reading these papers, it is tempting to conclude that they provide solutions for higher-order secure (glitch-resistant) masking gadgets, with a certain degree of composability. Indeed, most of them use the number of shares as a parameter of their designs and provide performance evaluations for full ciphers (which suggests composability is part of the authors' concerns). Yet, contrary to the usual situation in software-oriented masking, none of these proposals comes with a probing security proof at arbitrary order. For example, the CMS implementation in [CRB+16] investigates the concrete security of a second-order masked AES design (using the tools of [SM16]), the DOM implementations in [GMK17] investigate the concrete security of first- and second-order masked AES designs, the UMA implementations in [GM17] investigate the concrete security of d-th order masked `Ascon` designs for $d = 1, 2, 3$, and, analogous to the GLM scheme, analyze the side-channel resistance of the `Ascon` S-box for $d = 1, 2, 3$ using the formal verification tool introduced in [BGI+18]. Hence, these examples raise the question whether the CMS, DOM, UMA and GLM algorithms (or their generalization) directly lead to higher-order secure implementations, or whether the lack of proofs for these designs leaves room for weaknesses in the higher-order cases, that require attention/tweaks? We show the second statement is correct by:

- exhibiting a local flaw in the (generalized) CMS multiplication of [CRB+16],
- exhibiting a local flaw in the DOM-*dep* multiplication of [GMK16],
- exhibiting a composability flaw in the UMA of [GM17],
- showing that these flaws are reproduced in the GLM of [GIB18].

We note that these flaws do not invalidate the innovative ideas in these schemes: they only show that when moving to higher security orders, the engineering intuition that led to the successful design of gadgets secure at low orders benefits from a more formal analysis. In this regard, our main claim is that this collection of examples illustrates the difficulty to interpret the (lack of) higher-order security guarantees provided by CMS, DOM, UMA and GLM, and that, without the appropriate tweaks, these schemes cannot be extended beyond the contexts in which they were exhaustively analyzed. The latter leads to an error-prone situation for engineers willing to implement higher-order (glitch-resistant) masking in hardware. We use it to argue that as in the software case, *hardware-oriented masking schemes should either restrict claims to the specific orders that have been exhaustively investigated, or provide a hand-made proof for arbitrary orders.*

The need of robust probing security. The previous issues can be solved by integrating the additional information provided by physical defaults such as glitches in the probing model, as recently proposed by Faust et al. [FGP+18]. This reference describes a variant of the multiplication algorithm of Ishai et al. in [ISW03] and proved its security in the *robust probing model* for this purpose. To the best of our knowledge, this is the first (and so far only) multiplication algorithm proven secure and composable at arbitrary orders in the presence of glitches. In this respect, one more important question is whether dealing with and analyzing physical defaults and composability issues jointly is strictly needed? For example, is it enough to combine a glitch-resistant (probing) secure TI gadget with a strong (e.g., SNI) refresh and well-placed registers to obtain a gadget that is composable in the presence of glitches? We answer the question negatively by providing a counterexample to this approach, hence proving that analyzing the glitch-resistance and composability of masked gadgets independently is not enough, which provides a strong case for the need of the robust SNI (or a similar) abstraction – and justifies our subtitle.

Experimental confirmation. Since the masking schemes we investigate were not systematically analyzed in the (robust) probing model so far, the final problem we tackle is whether flaws in this model translate into concrete (in)security issues. We answer this question by investigating the concrete exploitability of the (local and composability) flaws exhibited in higher-order TIs, DOM, UMA and GLM based on an FPGA case study. Our experiments exhibit no big gap between theory and practice. The flaws put forward theoretically can be observed experimentally, sometimes leading to lower attack complexities than the generic attack at order $(d+1)$, sometimes not for the – low – security orders we consider experimentally (which, as per the analysis in [DFS15], Section 4.2, already implies concrete impact for some noise higher levels). In all cases, we argue that the presence of these flaws is problematic, since it prevents the extrapolation of the security guarantees of these schemes to higher orders. We also use our experiments to discuss the additional constraints that the robust probing security abstraction implies on the placement of registers within masked hardware implementations.

2 Background

We first recall security definitions that are relevant to our discussions.

The t-probing model was introduced by Ishai et al. in [ISW03] in order to prove the security of masked implementations. It assumes an adversary who can probe a limited number t of wires inside the target implementation. Probing security requires that the observation of these wires does not allow the adversary to learn sensitive information. Formally, this implies to define the target implementation as a circuit C (e.g., modeled as a graph) or as a sequence of leaking operations. Due to its simplicity, probing security was popular to analyze the first proposals of higher-order masking schemes. We next use the following definition:

Definition 1 (t-probing security [ISW03, RP10])**.** A circuit C is t-probing secure iff every t-tuple of its intermediate variables is independent of any sensitive variable.

In the case of block ciphers, sensitive variables typically correspond to partial computation results depending on the plaintext and key [CPR07]. Concretely, probing security can be achieved by splitting every sensitive variable k in at least $t+1$ values (usually called *shares*) so that their sum gives k, performing all computations on these shares, and re-combining the final result only.

One limitation of this definition of probing security is that it does not provide any guarantee of composability. Thus, while it is sufficient for the direct analysis of a complete circuit C, it does not allow the separate analysis of smaller circuit gadgets G. The latter typically comes in handy as the size of the circuits and the number of shares grows, making the direct analysis unpractical. More precisely, when gadgets are composed to produce a more complex circuit, it is needed to take into account that using an output of a gadget as input of another one can give additional information to the adversary. The following definitions of NI and SNI have been introduced by Barthe et al. for this purpose:

Definition 2 ($t-$Non-Interference [BBD$^+$16])**.** A circuit gadget G is $t-$*Non-Interfering* ($t-$NI) iff for any set of t_1 probes on its intermediate values and every set of t_2 probes on its output shares with $t_1 + t_2 \leq t$, the totality of the probes can be simulated with only $t_1 + t_2$ shares of each input.

Definition 3 ($t-$Strong Non-Interference [BBD$^+$16])**.** A circuit gadget G is $t-$ *Strong Non-Interfering* (t-SNI) iff for any set of t_1 probes on its intermediate values and every set of t_2 probes on its output shares with $t_1 + t_2 \leq t$, the totality of the probes can be simulated with t_1 shares of each input.

As illustrated in [BBP$^+$16] for the case of the AES S-box, combining NI and SNI gadgets enables compositional reasoning for arbitrary circuits. In order to satisfy these definitions, one has to build a simulator which can mimic the adversary's view using only black-box access to G (i.e., without the knowledge of any internal wire but only $t_1 + t_2$ shares (in the NI case) or t_1 shares (in the SNI case) of each secret input). The simulation is successful if no distinguisher can tell apart the simulation from the adversary's view. In this respect, one important technical clarification is that in the definitions of Barthe et al., the distinguisher can access the joint distribution of the (simulated) probes and input shares (which is strictly necessary for the compositional proofs). As a result, SNI is a stronger notion than NI, which is itself a stronger notion than probing security.

We finally introduce the robust probing model with the following example of a TI gadget implementing a Toffoli gate (i.e., $c = (x \odot y) \oplus z$, where $\odot$ denotes the logical `AND` and $\oplus$ denotes the logical `XOR` operation):

$$\begin{aligned} c_1 &= (x_2 \odot y_2) \oplus (x_2 \odot y_3) \oplus (x_3 \odot y_2) \oplus z_2, \\ c_2 &= (x_3 \odot y_3) \oplus (x_3 \odot y_1) \oplus (x_1 \odot y_3) \oplus z_3, \\ c_3 &= (x_1 \odot y_1) \oplus (x_1 \odot y_2) \oplus (x_2 \odot y_1) \oplus z_1, \end{aligned} \tag{1}$$

with the subscripts of the x, y, z, c variables indicating the shares' indices.

Based on this example, first assume that the gadget is implemented in a single cycle and in a glitch-free manner. In this case, the adversary can only probe the input shares x_i, y_i, z_i and output shares c_i, but not the intermediate values. That is, thanks to the glitch-free hardware, the output shares are produced from the input shares without any transient state that would leak additional information. It is easy to see that such an (ideal) gadget is 2-probing secure.

In practice though, most hardware implementations are not glitch-free and transient values leak additional information about the internal values [MPG05, MPO05]. The latter can be captured by the robust probing model which assumes that probes are "*extended*" so that when applied to any wire of a combinatorial circuit, the adversary can observe all the inputs this wire depends on [FGP+18].[1] In this case, the adversary can choose between probing output values stored in registers (which cannot be extended) and internal values before they are stored in registers (which can be extended). For example, in the gadget of Equation 1 implemented in a single cycle, an extended probe on the internal value c_1 would give access to x_2, x_3, y_2, y_3 and z_2 to the adversary. Interestingly, thanks to the non-completeness property (which requires that every combinatorial gadget excludes at least one share of any sensitive variable), this TI gadget remains 1-probing secure. We will refer to such implementations as *glitch-resistant*, reflecting the fact that they can cope with glitches by design (in contrast to *glitch-free* hardware which requires the problem to be solved at the micro-electronic level).

Additional remarks. As discussed in [FGP+18], the gadget of Equation 1 is neither NI nor SNI, even if it is implemented in glitch-free hardware. This is because it does not use any fresh internal randomness that can help the simulation. Remember that the distinguisher has access to the joint distribution of the (simulated) probes and input shares, so the simulator cannot leverage the shares of z for refreshing the shares of the $x \odot y$ product, as TIs typically exploit. Note also that the possibility for the adversary to choose between an output probe and an internal probe for the output values stored in a register (e.g., the c_i's in Equation 1) is essential to capture composability with glitches. Indeed, only the (stable, non-extended) output probes are included in the t_2 probes that are excluded from the input shares' count in the SNI definition.

3 Consolidated Masking Scheme (CMS)

At CRYPTO 2015, Reparaz et al. presented links between the established ISW multiplication and the concept of TIs [RBN+15]. They introduced an approach to realize masking in hardware with only $n = d + 1$ shares where d denotes the order of protection (i.e., number

---

[1] We only describe the glitch-extended probes that will be relevant to our discussions. Extensions corresponding to other physical defaults are discussed in [FGP+18].

of probes of the adversary), which we denote in the following as CMS.[2] This scheme was later applied in [CRB+16] to implement a masked AES with only $n = d + 1$ shares. In this section, we first recall the CMS multiplication as introduced in [RBN+15] and substantiated in [CRB+16]. Then we present a third-order flaw based on the particular (ring) refreshing strategy of the scheme.

3.1 Multiplication with Independent Inputs

While CMS can be applied to many different operations, we restrict our analysis to the common multiplication of two inputs. To this end, we rely on the description given in [CRB+16] for a two-input AND gate. Their approach is based on the consecutive application of multiple layers to the input shares: non-linear layer $\mathcal{N}$, linear layer $\mathcal{L}$, refresh layer $\mathcal{R}$, synchronization layer $\mathcal{S}$ (i.e., register stage), compression layer $\mathcal{C}$. In the case of $n = d+1$ shares masking, the linear layer $\mathcal{L}$ is skipped for the multiplication to ensure that each term given to the refresh layer $\mathcal{R}$ contains only one share of each input variable. This refresh is done in a circular manner (cf. Figure 1) requiring $(d + 1)^2$ random elements. In the compression phase, the refreshed values are summed up in order to achieve $n = d + 1$ shares for the output. The authors of [CRB+16] provide concrete instantiations only up to protection order $d = 2$. Based on these descriptions, we generalize their approach for an arbitrary number of shares with the algorithmic description in Algorithm 1.

Algorithm 1 CMS multiplication algorithm with $n \geq 2$ shares.

```
Input: shares $\mathbf{a} = (a_i)_{1\leq i\leq n}$ and $\mathbf{b} = (b_i)_{1\leq i\leq n}$, such that $\bigoplus_i a_i = a$ and $\bigoplus_i b_i = b$.
Output: shares $\mathbf{c} = (c_i)_{1\leq i\leq n}$, such that $\bigoplus_i c_i = a \cdot b$.
  for $i = 1$ to $n$ do
    $c_i = 0$
    for $j = 1$ to $n$ do
      $c_i = c_i + (a_i \cdot b_j + r_{(i-1)\cdot d+j \mod n^2} + r_{(i-1)\cdot d+j+1 \mod n^2})$;
    end for
  end for
```

Note that this algorithm is only a functional representation of the scheme and lacks the concrete distinction into layers which is the basis of the CMS concept. Therefore, Figure 1 depicts the layer-wise architecture for the $d = 3$ case based on the notations of Algorithm 1. Note also that the implementation of Figure 1 does not satisfy the non-completeness property of standard TIs (which is similar to Figure 1 in [CRB+16] that we extend in the natural manner). We will discuss the impact of tweaking the design to make it non-complete later in the section.

3.2 A Third-Order Flaw

In the following, we demonstrate that the CMS multiplication as given in Algorithm 1 and Figure 1 does not provide the claimed security guarantees for arbitrary d. Our flaw stems from the combination of the circular refresh strategy $\mathcal{R}$ with the specific compression layer $\mathcal{C}$. In particular, summing up all terms $a_i \cdot b_j + r_k$ for a specific value of i cancels out many of the random terms from the refresh layer. While this is not problematic for the orders $d = 1, 2$ considered in [CRB+16], it leads to a trivial attack with only three probes for orders $d \geq 3$. Concretely, after compression each output share c_i can be written as:

$$c_i = a_i \cdot b + r_{(i-1)\cdot d+1} + r_{i\cdot d+1}. \tag{2}$$

---

[2] Earlier proposals of higher-order TIs usually needed more shares [BGN+14a].

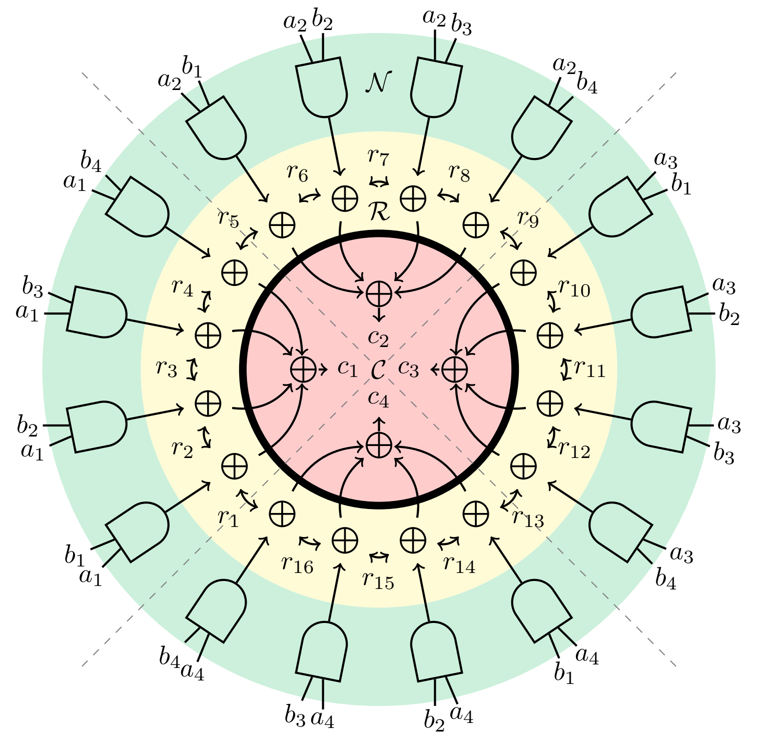


Figure 1: Architecture of CMS multiplication extending the proposal in [CRB+16] to $d = 3$, consisting of the (green) non-linear layer $\mathcal{N}$, the (yellow) (ring) refresh layer $\mathcal{R}$, the (black) synchronization (registers) layer $\mathcal{S}$, and the (red) compression layer $\mathcal{C}$.

Therefore, by probing:

$$P_1 = c_i, \tag{3}$$

$$P_2 = r_{(i-1)\cdot d+1}, \tag{4}$$

$$P_3 = r_{i\cdot d+1}, \tag{5}$$

the adversary can observe a joint distribution (P_1, P_2, P_3) which depends on $a_i \cdot b$. While a_i is still a random value independent of a, it does not suffice as a mask for b given the "zero bias" of multiplicative masking schemes [GT03] (e.g., for the binary case, $a_i \cdot b = 1$ implies $b = 1$). Thus, the joint distribution leaks about the sensitive value b invalidating the security of the multiplication scheme.

Example 1 ($d = 3$)**.** For better understanding, we demonstrate an attack on the simplest case with $d = 3$ (i.e., $n = 3 + 1 = 4$ shares) which is shown in Figure 1. The probes are placed according to the aforementioned guidelines as follows:

$$P_1 = c_1 = a_1 \cdot b + r_1 + r_5, \tag{6}$$

$$P_2 = r_1, \tag{7}$$

$$P_3 = r_5. \tag{8}$$

The histograms of the joint distribution of (P_1, P_2, P_3) for fixed $b \in \mathbb{F}_2$ are given in Table 1. It is noticeable that they differ based on the value of b. Therefore, an adversary could distinguish the value of b with only three regular probes for any order $d \geq 3$ which is less than the claimed order of security.

Table 1: Histogram of the joint distribution of (P_1, P_2, P_3) for $b = 0$ and $b = 1$.

(P_1, P_2, P_3)	0	1	2	3	4	5	6	7
$b = 0$	2	0	0	2	0	2	2	0
$b = 1$	1	1	1	1	1	1	1	1

3.3 Discussion

We first insist that the previous attack does not contradict the claims in [CRB+16] since (in the core of the paper) the authors make clear that their analysis is limited to the case $d = 2$, i.e., with $n = 3$ shares. Thus, our only claim is again that the title of the paper can be misleading, since the natural extension of the proposed algorithms does not lead to higher-order secure gadgets as one could expect, and the paper does not explicitly mention a low-order limitation. It leaves as an open problem to find efficient solutions to fix this flaw (some proposals can be found in Thomas De Cnudde's PhD dissertation [De 18]).

We also observe that considering a non-complete compression layer, despite not necessary from the glitches viewpoint (since a register stage prevents the propagation of the glitches before the compression in Figure 1), would actually make the attack slightly more difficult. For example, imagine that no c_i value in Figure 1 depends on the 4 shares of a_i or b_i: then an attack would only succeed by probing multiple c_i's together with the r_i's at their "borders" (postponing the appearance of the security flaw to higher orders). Denoting the number of c_i's to probe with m, the generalized attack will work with at most $3m$ probes, and possibly less if the probed c_i's share a border. It is however interesting that the non-completeness property turns out to be useful for composability purposes. We leave the exploitation of this observation (e.g., to design secure and efficient implementations at low orders) as an interesting scope for further research.

We finally note that, as the ring refreshing in [CRB+16] is not SNI, the generalization of the S-box design in this reference to higher-orders also suffers from composability flaws similar to the ones of the UMA and GLM schemes.

4 Domain-Oriented Masking (DOM)

Domain-Oriented Masking (DOM) was proposed in 2016 by Groß et al. with the goal to enable d-th order secure masking in hardware with only $n = d + 1$ shares [GMK16, GMK17]. The main contribution is a masked multiplier initially denoted as DOM-*indep*. Its randomness distribution is closely related to the ISW multiplication and it is therefore probing secure given independently shared inputs. However, for the multiplication of dependently shared inputs, Groß et al. include another alternative multiplication scheme called DOM-*dep* in their eprint version [GMK16]. It is used in some of their proposed designs to improve efficiency. In this section, we first recall the specification of DOM-*dep* and then demonstrate a $(\lceil \frac{d}{2} \rceil + 1)$th-order flaw for orders $d \geq 2$, contradicting the DOM security claims.

4.1 Multiplication with Dependent Inputs

A straightforward way to extend DOM-*indep* for allowing dependently shared inputs is to SNI refresh one of the inputs [GR17]. This provides security but comes with significant costs in randomness, area, and latency. DOM-*dep* was proposed as a more efficient alternative which does not require re-sharing. Instead, a blinding value z is introduced to multiply the inputs a and b as:

$$c = a \cdot b = a \cdot (b + z) + (a \cdot z). \tag{9}$$

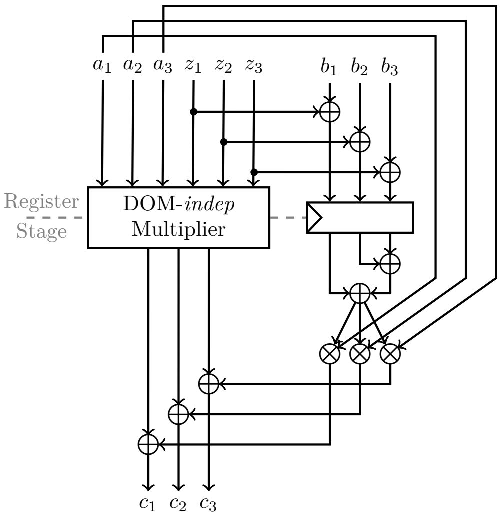


Figure 2: Architecture of DOM-*dep* for $d = 2$.

Since z is a random value, the authors proposed an efficient way to compute $a \cdot (b + z)$ by first decoding $(b + z)$, i.e., summing all shares, and then multiplying the result with each share of a. Therefore, DOM-*dep* requires only one full DOM-*indep* multiplication compared to two for the previously outlined (straightforward) approach. The generic scheme for any order is given in Algorithm 2 based on the descriptions provided in [GMK16], where the **x** notation is used to represent vectors of shares.

Algorithm 2 DOM-*dep* multiplication algorithm with $n \geq 2$ shares.

```
Input: shares $\mathbf{a} = (a_i)_{1 \leq i \leq n}$ and $\mathbf{b} = (b_i)_{1 \leq i \leq n}$, such that $\bigoplus_i a_i = a$ and $\bigoplus_i b_i = b$.
Output: shares $\mathbf{c} = (c_i)_{1 \leq i \leq n}$, such that $\bigoplus_i c_i = a \cdot b$.
  for $i = 1$ to $n$ do
    $z_i \xleftarrow{\$} \mathbb{F}_q$
    $x_i \leftarrow (b_i + z_i)$
  end for
  $x = \mathsf{Decode}(\mathbf{x})$
  $\mathbf{c} =$ DOM-indep $(\mathbf{a}, \mathbf{z})$
  for $i = 1$ to $n$ do
    $c_i \leftarrow c_i + (a_i \cdot x)$
  end for
```

We note that (as for the CMS multiplication), Algorithm 2 is only a functional representation of DOM-*dep* and does not show the concurrent operations and register stages required for a hardware design. Instead, these are depicted in Figure 2 (based on Figure 4 of [GMK16]) for the special case of order $d = 2$.

4.2 A $(\lceil \frac{d}{2} \rceil + 1)$th-Order Flaw

In the following, we demonstrate that DOM-*dep* as given in Algorithm 2 and Figure 2 does not provide the claimed security guarantees for arbitrary orders d. For simplicity, we assume that the input encodings of a and b are identical (i.e., $a_i = b_i,\ 1 \leq i \leq n$). The main problem of DOM-*dep* stems from the $\mathsf{Decode}(\mathbf{x})$ operation. In an idealized world (corresponding to unrealistic glitch-free hardware discussed at the end of Section 2), this operation would be performed without leaking information on intermediate values, and an adversary would not be able to probe any intermediate sum of the decoding. Therefore, with one probe on $\mathsf{Decode}(\mathbf{x})$ the adversary either receives (a) one of the input shares $b_i + z_i$, or (b) the output value $b + z$. Both cases cannot be used to construct an attack, since for (a) it is similar as probing an intermediate value in the secure DOM-*indep* multiplier (assuming $a_i = b_i$), and for (b) z is a true random value which cannot be probed directly (only its shares z_i). Hence, DOM-*dep* might be secure in this idealized model.

However, as a hardware-oriented masking scheme, DOM is aimed to be glitch-resistant and therefore to maintain security even in the more practical robust probing model. In this case, the adversary has access to more powerful probes which enable her to extract sensitive information from DOM-*dep*. For the operation $\mathsf{Decode}(\mathbf{x})$, probing the output value $b + z$ provides information about all input sums $b_i + z_i$, since there are no registers to prevent glitches. This alone does not suffice for an attack, because the shares b_i are still masked by the z_i's. Nevertheless, by also probing in the DOM-*indep* multiplication of a and z, it is possible to break the scheme. In particular, the adversary first accesses:

$$\{b_1 + z_1, b_2 + z_2, \dots, b_{n/2} + z_{n/2}\}, \tag{10}$$

with only one probe on the output of $\mathsf{Decode}(\mathbf{x})$. Then $\frac{n}{2}$ probes are placed in the cross-product terms of DOM-*indep* which consist of some of the already probed random terms z_i and the remaining unprobed input shares a_i:

$$\{a_{d/2+1} \cdot z_1, a_{n/2+2} \cdot z_2, \dots, a_n \cdot z_{n/2}\}. \tag{11}$$

The distribution of these $(\lceil \frac{d}{2} \rceil + 1)$ variables depends on the value of a. For odd values of n, another probe might be necessary to probe a_n when considering $\lfloor \frac{n}{2} \rfloor$ cross-product terms. However, since there is no register between $\mathsf{Decode}(\mathbf{x})$ and the subsequent share-wise multiplication, the adversary can simply place the extended probe on the computation $x \cdot a_n$. This provides the same input sums as before with the added benefit of leaking a_n. Therefore, DOM-*dep* does not provide the desired robust probing security for orders $d \geq 2$.

Example 2 ($d = 2$)**.** We demonstrate an attack on the simplest case with order $d = 2$ (i.e., $n = 3$) as shown in Figure 2. The probes are placed on the output of the computation of $x \cdot a_3$ and on one term of the cross-product according to the aforementioned guidelines. We choose to target the intermediate variable $(a_1 + z_1) \cdot a_3$ accessed by the extended probe. This leads to following probed variables:

$$P_1 = (a_1 + z_1) \cdot a_3, \tag{12}$$

$$P_2 = a_2 \cdot z_1, \tag{13}$$

The histograms of the joint distribution of (P_1, P_2) for fixed $a \in \mathbb{F}_2$ are given in Table 2. It is noticeable that they differ based on the value of a. Therefore, an adversary could distinguish the value of a with only one extended and one regular probe, which is less than the claimed order of security.

Table 2: Histogram of the joint distribution of (P_1, P_2) for $a = 0$ and $a = 1$.

(P_1, P_2)	0	1	2	3
$a = 0$	5	1	1	1
$a = 1$	4	2	2	0

4.3 Discussion

As previously mentioned, there is an easy (but costly) fix to this attack by using an SNI refreshing gadget before each multiplication of dependently shared values. By contrast, the introduction of register stages in $\mathsf{Decode}(\mathbf{x})$ does not solve the problem since any intermediate sum containing more than one share of b could be used for an attack with less than $d+1$ probes for protection orders $d \geq 3$. This for example implies that even a software implementation of DOM-*dep* without glitches would be vulnerable to the presented flaw. It recalls the gradation between the (minimum) amount of information leaked by an ideal (glitch-free) hardware implementation, the (intermediate) amount of information leaked by a standard software implementation (where some intermediate variables are leaked) and the worst-case amount of information leaked by a glitchy hardware one.

5 Unified Masking Approach (UMA)

Following the concept of DOM, Groß and Mangard proposed a more randomness-efficient hardware multiplication scheme denoted as Unified Masking Approach (UMA) in [GM17, GM18]. It essentially combines the software-oriented parallel masking algorithm of Barthe et al. [BDF⁺17] with the randomness optimizations of Belaid et al. [BBP⁺16] in order to achieve (so far the most) randomness-efficient masked multiplication in hardware. For certain orders d, UMA even outperforms known software solutions. In contrast to [BDF⁺17, BBP⁺16], the authors of UMA do not state any limitation regarding the composability of their multiplication scheme. In the following, we first shortly recall the UMA concept and then highlight composability issues.

5.1 A (not so) Universal Multiplication

The basis of UMA is the multiplication algorithm from Barthe et al. [BDF⁺17]. It is extended with optimizations from Belaïd et al. [BBP⁺16] and DOM [GMK17] for certain protection orders d to reduce the randomness complexity even further. Therefore, the generic solution given in Algorithm 3 ($\mathbf{a}_{+i}$ denotes a rotation of the share vector $\mathbf{a}$ by i positions) includes a distinction of different cases for d to account for these optimizations. The multiplication is split into five blocks: *Inner-Domain*, *Complete*, *Pseudo-Complete*, *Half-Complete*, and *Incomplete*.

- *Inner-Domain*: In this block, the inputs are multiplied share-wise. Since this operation is implemented without mixing the input domains (assuming independent inputs), it does not require the inclusion of register stages.

- *Complete*: With the *Pseudo-Complete* block, the *Complete* block implements the masked multiplication according to Barthe et al.'s algorithm. Each loop iteration is performed in parallel to each other, but a register stage is required after every addition to ensure security, resulting in a delay of five cycles.

- *Pseudo-Complete*: This block processes the remaining terms of Barthe et al.'s algorithm. It requires register stages after every addition, but the delay is four cycles since it contains one less addition than the *Complete* block.

- *Half-Complete*: This block contains a further case distinction for order $d = 2$. In this scenario, the multiplication is implemented according to Belaïd et al.'s optimal algorithm and requires three register stages. For the other cases, the authors rely on DOM which only adds a delay of one cycle, because the terms $\mathbf{r}^l + \mathbf{a} \cdot \mathbf{b}_{+2l+1}$ and $\mathbf{r}^l_{+2l+2} + \mathbf{a} \cdot \mathbf{b}_{+2l+2}$ are computed in parallel.

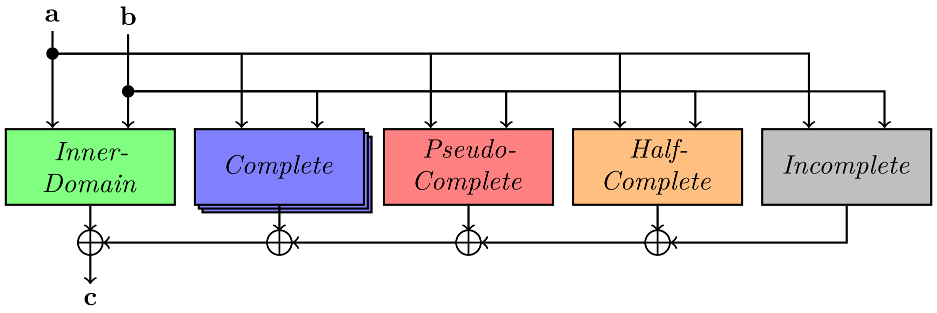


Figure 3: Connection of the UMA blocks [GM17].

- *Incomplete*: Similar to the previous block, the *Incomplete* terms are computed according to DOM and require the inclusion of one register stage.

Depending on the order d, these blocks are instantiated and their outputs are combined as depicted in Figure 5 from [GM17] (cf. Figure 3). *Inner-Domain* is always implemented and connected to $\lfloor \frac{d}{4} \rfloor$ *Complete* blocks, and optionally to one *Pseudo-Complete*, *Half-Complete*, or *Incomplete* block. Additional registers or control logic might be necessary to ensure synchronization between the different blocks given the difference in delay (which we will discuss in Section 8.2).

Algorithm 3 UMA multiplication algorithm with $n \geq 1$ shares.

Input: shares $\mathbf{a} = (a_i)_{1 \leq i \leq n}$ and $\mathbf{b} = (b_i)_{1 \leq i \leq n}$, such that $\bigoplus_i a_i = a$ and $\bigoplus_i b_i = b$.
Output: shares $\mathbf{c} = (c_i)_{1 \leq i \leq n}$, such that $\bigoplus_i c_i = a \cdot b$.

```
  $l = \lfloor \frac{d}{4} \rfloor$
  $\mathbf{c} = \mathbf{a} \cdot \mathbf{b}$                                  Inner-Domain
  for $i = 0 < \lfloor \frac{d}{4} \rfloor$ do
      $\mathbf{c} \leftarrow \mathbf{c} + \mathbf{r}^i + \mathbf{a} \cdot \mathbf{b}_{+2i+1} + \mathbf{a}_{+2i+1} \cdot \mathbf{b} + \mathbf{r}^i_{+1} + \mathbf{a} \cdot \mathbf{b}_{+2i+2} + \mathbf{a}_{+2i+2} \cdot \mathbf{b}$      Complete
  end for
  if $d \equiv 3 \mod 4$ then
      $\mathbf{c} \leftarrow \mathbf{c} + \mathbf{r}^l + \mathbf{a} \cdot \mathbf{b}_{+2l+1} + \mathbf{a}_{+2l+1} \cdot \mathbf{b} + \mathbf{r}^l_{+1} + \mathbf{a} \cdot \mathbf{b}_{+2l+2}$      Pseudo-Complete
  end if
  if $d \equiv 2 \mod 4$ then
      if $d = 2$ then
          $\mathbf{z} = \{r^l_1, r^l_2, r^l_1 + r^l_2\}$
          $\mathbf{c} \leftarrow \mathbf{c} + \mathbf{z} + \mathbf{a} \cdot \mathbf{b}_{+2l+1} + \mathbf{a}_{+2l+1} \cdot \mathbf{b}$
      else                                                       Half-Complete
          $\mathbf{c} \leftarrow \mathbf{c} + \mathbf{r}^l + \mathbf{a} \cdot \mathbf{b}_{+2l+1} + \mathbf{r}^l_{+2l+2} + \mathbf{a} \cdot \mathbf{b}_{+2l+2}$
      end if
  end if
  if $d \equiv 1 \mod 4$ then
      $\mathbf{z} = \{r^l, r^l\}$                                     Incomplete
      $\mathbf{c} \leftarrow \mathbf{c} + \mathbf{z} + \mathbf{a} \cdot \mathbf{b}_{+2l+1}$
  end if
```

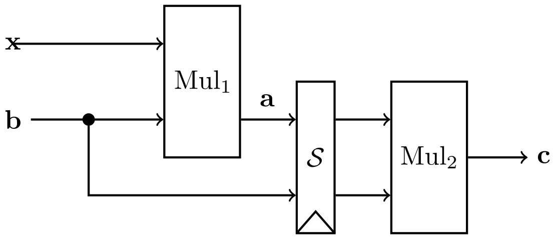


Figure 4: Composition of two UMA multiplications.

5.2 A Systematic Composability Flaw

Belaïd et al. and Barthe et al. analyzed the security of their multiplication algorithms with formal proofs and verification in regard to both probing security and SNI. Therefore, they were able to provide concrete assertions regarding the composability of their schemes. In particular, it was found that the randomness-optimized multiplications in [BBP$^+$16] are not SNI and that the parallel multiplications in [BDF$^+$17] are only d-SNI until order $d = 2$ (its composition with simple refreshing gadgets is d-SNI for larger d's). Therefore, a designer has to take great care where to utilize them without violating the security of the whole design. By contrast, for UMA the authors do not examine their multiplication regarding this criterion, and their case study uses the UMA multiplication without discussing composability explicitly. Therefore, a non-expert reader might be compelled to believe that the unified masking approach is indeed universal and can be used at any point of any masked design. In the following, we show that an exemplary composition of two UMA multiplications does not compose well.

Following the typical pattern of composability flaws put forward by Coron et al. in [CPRR14], our example is depicted in Figure 4. The input encoding **b** is initially refreshed by multiplying it with a random encoding **x**. Then the refreshed output **a** is multiplied with the original **b** resulting in **c**. This structure is commonly used when an input is multiplied with a linear transformation of itself, e.g., for the inversion in $GF(2^8)$ [BBP$^+$16]. For simplicity, we omitted the linear transformation from our construction. In addition to the register stage between the multiplications, there are multiple registers inside Mul$_1$ and Mul$_2$ depending on the order. For now, we assume that the registers are enabled in a sequential fashion, e.g., the second stage is enabled only after the first one. Given a freely-composable multiplication, e.g., ISW [ISW03] or DOM [GMK17], this structure should provide d-probing security. However, for UMA this is not true for orders $d > 1$ as we demonstrate by attacking the composition with d probes. Since the UMA multiplication differs in structure depending on the order, we look at multiple cases separately.

Example 3 ($d = 2$)**.** Firstly, we consider Belaïd et al.'s optimized multiplication for $d = 2$ (i.e., $n = 3$). In our structure, the second multiplication Mul$_2$ can be written as:

$$c_1 = a_1 \cdot b_1 + r_1^2 + a_1 \cdot b_2 + a_2 \cdot b_1, \tag{14}$$

$$c_2 = a_2 \cdot b_2 + r_2^2 + a_2 \cdot b_3 + a_3 \cdot b_2, \tag{15}$$

$$c_3 = a_3 \cdot b_3 + r_1^2 + r_2^2 + a_3 \cdot b_1 + a_1 \cdot b_3, \tag{16}$$

where $\{r_1^2, r_2^2\}$ denotes the randomness used for this multiplication (resp., $\{r_1^1, r_2^1\}$ for Mul$_1$). One possibility to attack b consists in probing a random element in Mul$_1$ and a cross-product term in Mul_2 as:

$$P_1 = r_1^1, \tag{17}$$

$$P_2 = a_1 \cdot b_3 = (x_1 \cdot b_1 + r_1^1 + x_1 \cdot b_2 + x_2 \cdot b_1) \cdot b_3. \tag{18}$$

Since the joint distribution of (P_1, P_2) (reproduced in Table 3) depends on the value of b, it can be used to distinguish the sensitive variable with only two probes which contradicts the security claims of UMA.

Table 3: Histogram of the joint distribution of (P_1, P_2) for $b = 0$ and $b = 1$.

(P_1, P_2)	0	1	2	3
$b = 0$	12	12	4	4
$b = 1$	14	10	2	6

This attack generalizes to higher orders. For simplicity, we first discuss the flaw for $d \equiv 0 \mod 4$, i.e., when the multiplication consists of only the *Inner-Domain* and *Complete* blocks. The first output share of Mul_1 is of the form:

$$\begin{aligned} a_1 = x_1 \cdot b_1 + r_1^1 + x_1 \cdot b_2 + x_2 \cdot b_1 + r_2^1 + x_1 \cdot b_3 + x_3 \cdot b_1, & \quad (19)\\ + r_3^1 + x_1 \cdot b_4 + x_4 \cdot b_1 + r_4^1 + x_1 \cdot b_5 + x_5 \cdot b_1, & \quad (20)\\ + \cdots & \quad (21)\\ + r_{\frac{d}{2}-1}^1 + x_1 \cdot b_{\frac{d}{2}} + x_{\frac{d}{2}} \cdot b_1 + r_{\frac{d}{2}}^1 + x_1 \cdot b_{\frac{d}{2}+1} + x_{\frac{d}{2}+1} \cdot b_1. & \quad (22) \end{aligned}$$

It contains $\frac{d}{2} + 1$ shares of each input encoding which are masked by $\frac{d}{2}$ random elements. Given that this output share is one of the inputs of Mul_2, it is multiplied with every share of $\mathbf{b}$. In particular, with $b_{\frac{d}{2}+2}$, i.e., a share that is not contained in a_1. By putting $\frac{d}{2}$ probes in Mul_1 and one probe in Mul_2 as:

$$P_1 = r_1^1, \tag{23}$$

$$P_2 = r_2^1, \tag{24}$$

$$\cdots \tag{25}$$

$$P_{\frac{d}{2}} = r_{\frac{d}{2}}^1, \tag{26}$$

$$P_{\frac{d}{2}+1} = a_1 \cdot b_{\frac{d}{2}+2}, \tag{27}$$

the adversary can access a joint distribution $(P_1, \ldots, P_{\frac{d}{2}+1})$ which depends on $\frac{d}{2} + 2$ shares of $\mathbf{b}$. Eventually, with the remaining $\frac{d}{2} - 1$ probes, the adversary can now access the still unknown shares by observing:

$$P_{\frac{d}{2}+2} = b_{\frac{d}{2}+3}, \tag{28}$$

$$\cdots \tag{29}$$

$$P_d = b_{d+1}, \tag{30}$$

which results in a joint distribution depending on $n = d + 1$ shares of $\mathbf{b}$ (i.e., all n shares) with only d probes which is against the universal security claim of UMA. This attack can be trivially applied to any order $d \equiv 0 \mod 4$.

For $d \equiv 3 \mod 4$ (resp., the DOM optimization for $d \equiv 2 \mod 4$), the *Pseudo-Complete* block (resp., *Half-Complete* block) adds two further shares of $\mathbf{b}$ and two random elements to the output share a_1 of Mul_1. Therefore, a similar attack can be repeated with $\frac{d}{2} + 2$ probes in Mul_1. In the incomplete case (i.e, $d \equiv 1 \mod 4$), one more share of $\mathbf{b}$ and one random element is added to a_1 and a similar attack requires $\frac{d}{2} + 1$ probes on Mul_1.

5.3 Discussion

Since the algorithms [BDF+17, BBP+16] which serve as a basis for UMA are not composable at every order, the fact that composability flaws pop up in UMA is not surprising. Interestingly, such composability flaws do not as directly appear in the specific application to the `Ascon` cipher chosen by Groß and Mangard. The main reason is that the `Ascon` S-box does not directly lead to simple dependent multiplications as in Figure 4 and composability flaws may only appear for larger orders d and require to combine the shares of several rounds. So as mentioned in the Introduction, the main problem of [GM17] is its interpretation. On the one hand, the gadgets used in UMA are clearly not universally composable. On the other hand, exhaustive analysis for full circuits at high security orders is rapidly computationally hard [EWS14, BBD+15, Rep16, GIB18]. Admittedly, it may very well be that using SNI gadgets in this context is an overkill and that the biases caused by the lack of composability remain hard to exploit given the noise levels considered in concrete implementations until quite large security orders (as per an argument in the lines of [DFS15], Section 4.2), or even that additional refreshings are not needed for this particular circuit. The tools introduced in [Dae16] could be one option for the evaluation of this issue, which we leave as an interesting scope for further research.

6 Generic Low-Latency Masking (GLM)

Low latency is an optimization goal which has only been recently examined in the context of masking and side-channel analysis. Some specific investigations have been targeting the block ciphers `Prince` and `Midori` [MS16], KECCAK [ABP+18] and the AES S-box [GC17]. However, the latter investigations do not provide generic solutions for arbitrary functions at arbitrary orders. In this respect, an important observation is that all the algorithms discussed in the previous sections require a fixed delay of one or multiple register stages per multiplication. Therefore, Groß et al. proposed a Generic Low-Latency Masking (GLM) scheme in [GIB18]. They essentially trade randomness and area for a lower latency by skipping the compression of the shares as much as possible in their designs. In the following, we first recall the concept of GLM and then briefly show the problems arising from the proposed refreshing and compression strategies.

6.1 Low-Latency Masking and Compression

The main idea of GLM is to skip the compression function inherent to the other masked multiplications. That is, instead of summing the cross-product terms in order to obtain $n = d + 1$ output shares, Groß et al. propose to continue the computations with the $(d + 1)^2$ uncompressed shares:

$$(a_1 \cdot b_1) \quad (a_1 \cdot b_2) \quad \dots \quad (a_n \cdot b_n). \tag{31}$$

To avoid collisions between shares (e.g., for the computation of $(a \cdot b) \cdot b$), certain inputs – and even parts of the circuit – are duplicated and independently encoded, ensuring that the inputs to every non-linear function are independent.

While this methodology can be applied to arbitrary functions, every non-linear operation increases the number of shares. When this number becomes prohibitive, the authors of GLM propose to use a refresh operation followed by a register stage and a compression function in order to reduce the number of shares again to $n = d + 1$. They recommend using the CMS refresh from [RBN+15] for this purpose.

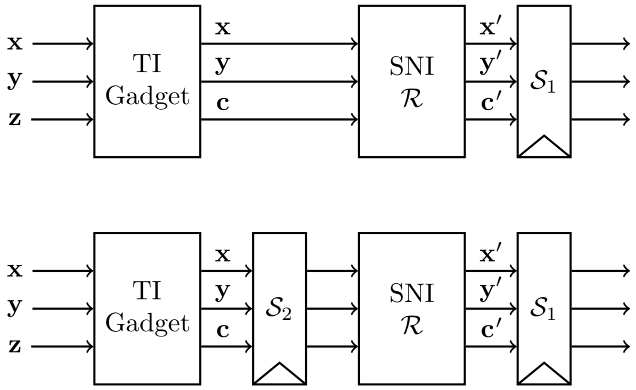


Figure 5: Examples of non-complete and SNI gadgets that do not compose in the robust probing model (independent of the synchronization stages / registers).

6.2 Combining Previous Attacks

As noted in Section 3.2, the CMS refresh from [RBN⁺15] does not generalize to arbitrary orders. Furthermore, even with a different distribution of the cross-product terms (i.e., with non-completeness after the compression), the CMS refreshing is not SNI for $d > 2$ and it opens the door to composability flaws as discussed in Section 5. Therefore, any GLM architecture which relies on this refresh might be vulnerable to these previous attacks. Fixing this issue is not trivial as the compression layer $\mathcal{C}$ does not include a dedicated register stage which leads to further composability problems, as discussed in Section 8.

6.3 Discussion

While the results in this section do not bring new technical elements, they illustrate that the interpretation issues that we mention in the introduction can easily lead to propagation of errors from one design to another, which can be avoided by formulating the algorithms and their security claims accurately. As in the previous section, we re-insist that the exploitation of a flaw may not be obvious for all designs (e.g., in the case of the `Ascon` cipher). So our only statement is that these limitations are not clearly stated in the original GLM paper and limit its claims for generality. Finding updated refresh $\mathcal{R}$ and compression $\mathcal{C}$ algorithms, which take these issues into account and enable true generality, is an interesting topic for future work, as also noted by the authors of [GIB18].

7 On the Need of the Robust Probing Model

The previous (and next) sections show that probing security and composability are the result of a delicate trade-off between combinatorial computations, refreshing layers and register stages. In this respect, one natural question is whether solving these problems separately is (formally) sufficient to solve them jointly. In this section, we show that combining a glitch-resistant (non-complete and probing-secure) gadget with SNI refreshes and registers is in fact not sufficient, providing a case for the need of the robust SNI abstraction in [FGP$^+$18] (or a similar abstraction allowing to capture this issue).

For this purpose, we use the simple examples of Figure 5 where the TI gadget is the one given in Equation 1 (Section 2) and the SNI refresh is a 3-bit ISW refresh. First consider the top design with only one synchronization (register) stage $\mathcal{S}_1$. In this case, it is easy to see that a "glitch-extended probe" on one share of $\mathbf{c}'$ reveals all the intermediate randomness (coming from the SNI refresh) needed to compute this share of $\mathbf{c}'$ from the input shares of $\mathbf{x}, \mathbf{y}, \mathbf{z}$. Hence, this randomness cannot be used to simulate this single (extended) adversarial probe. Furthermore, adding a second register layer does not solve the problem. In this case, the adversary can directly probe $\mathbf{c}$, which cannot be simulated (since the first TI gadget only leverages the input shares to ensure probing security).

As for the previous sections, the latter examples do not imply that there are no combinations of TI gadgets, SNI refreshes and registers that are robust against glitches and composable (e.g., by using more than $n = d + 1$ shares). They just show that formally, the definitions of the non-completeness property and of SNI (without glitches) do not compose. As suggested in [FGP+18, Lemma 5], some form of simulatability (e.g., captured by the robust-NI property) is needed for the first (combinatorial) gadget of Figure 5. We believe such a composability is increasingly needed as the claimed security orders in hardware masking increase, making exhaustive analysis impossible for full implementations.

8 Experimental validation

In Sections 3 to 6 we have analyzed the local and compositional security of multiplication gadgets which have been proposed for glitch-resistant hardware masking and revealed that the higher-order versions of all these schemes are affected by flaws in the (robust) probing model. In this section we answer the question whether these flaws actually lead to exploitable leakage in real-world power measurements from hardware implementations of the corresponding schemes. After concluding positively in this regard, we discuss the severity of these leakages with respect to the practical security level of the investigated circuits. Whether or not the detected weaknesses invalidate the claims of the respective authors is open to interpretation (it in part depends on whether claims are stated in terms of security order or number of measurements to disclose the key). Yet, they effectively limit the generality of those proposed gadgets, which is an important cautionary note to designers willing to implement them. This result confirms the necessity for proofs in the robust probing model when claiming security for arbitrary orders and when aiming to protect larger non-linear functions (like substitution boxes of block ciphers) or full cryptographic primitives. Besides, while all of the exhibited flaws up to this part of the paper originate from a lack of fresh randomness, compositional security in hardware also highly depends on the correct instantiation of register stages. Additional concerns regarding DOM, GLM and UMA in this respect, and their connection to the robust probing model, are discussed in the second part of this section and in Appendix A.

Setup. In order to examine the detectability of the aforementioned flaws in practice, we conducted common fixed-versus-random t-test evaluations [GJJR11, CDG+13] using power traces measured from an FPGA. We have used a SAKURA-G board [sak] and implemented the designs explained below on its Spartan-6 FPGA operated at a clock frequency of 6 MHz. The power traces have been measured by means of a digital sampling oscilloscope at the sampling rate of 500 MS/s by monitoring the output of the embedded AC amplifier of the SAKURA-G, which amplifies the voltage drop over the resistor placed in the Vdd path of the target FPGA.

We have followed the procedure explained in [SM15] to collect the corresponding traces suitable for fixed-versus-random t-test analysis. In this scenario the shared input and the required fresh randomness are generated by the control FPGA. Hence, the target FPGA, whose leakage is measured, just operates on the given input and does not generate any true- or pseudo-randomness. It is noteworthy that no masking or unmasking is performed in either the control or the target FPGA. The whole communication between the PC and the measurement board as well as between both FPGAs on the board is performed in a shared manner. Using the resulting traces we conducted first- and higher-order univariate and multivariate analyses, by using the incremental formulas introduced in [SM15].

8.1 Exploiting the Flaws

In order to keep the following results comparable we used ordinary $GF(2^4)$ multipliers as a basis to construct each of the masked multiplication gadgets. Thus, in all designs which we analyzed the unshared operands are of 4-bit size. We present results for CMS, DOM and UMA and omit the GLM scheme to avoid redundancy, since it simply adopts the flaws from CMS. We demonstrate that in all cases the leakage that is predicted by the exhibited flaws can be observed as multivariate leakage in the corresponding statistical moments.

8.1.1 CMS

As detailed in Section 3, CMS is neither probing secure nor SNI in the presence of glitches for orders $d > 2$, since its randomness distribution inherited from the ring structure is insufficient to deliver security for arbitrary protection orders. We analyze the construction for order $d = 3$ (i.e., $n = 4$), as it is the simplest case that suffers from the third-order flaw. To be more precise we have implemented the design shown in Figure 1 and replaced all AND gates by $GF(2^4)$ multipliers. Figure 6 shows a sample trace and the results of a non-specific t-test up to the fourth statistical moment with 300 million traces. It is obvious that the design only exhibits univariate leakage in the fourth order, as it would be expected from a securely $(d + 1)$-masked multiplication gadget with four shares. When moving to the multivariate analysis, however, third-order leakage can be observed with less than 100 million traces, as illustrated by Figure 7. The t-statistics curve in Figure 7(a) is obtained by calculating the second-order centralized moment of the joint distribution of each time sample with the corresponding time sample from the consecutive clock cycle (i.e., shifted by an offset of 1 clock cycle - or 83 time samples), starting from time sample 500. The t-statistics curve in Figure 7(c) is calculated with the third-order centralized moment of the joint distribution of each time sample with itself and the corresponding time sample in the consecutive clock cycle (starting from time sample 500 with 83 time samples per clock cycle). For instance, time sample 250 in Figure 7(c) corresponds to the third-order centralized statistical moment of the joint distribution of time samples 750, 750 and 833 in Figure 6(a).

It is noteworthy that in our first attempt of measuring this implementation we did not observe any (univariate or multivariate) leakage up to the third-order with up to 500 million traces, since the signal-to-noise ratio (SNR) was too small to detect the bias in the measurements associated with the joint distribution of the three probes given in Equations (6) to (8). Thus, for the experiments that led to the t-test results in Figures 6 and 7, we had to make sure that the manipulation of the probed values consumes enough power to overcome the small SNR. In this regard, we instantiated three extra modules connected to the 4-bit values c_1, r_1 and r_5 to amplify their corresponding leakage. Each of such extra modules (so-called leakage amplifiers) is formed by 6 times cascading a MIX module, which is a linear operation multiplying its 4-bit input to the following binary matrix (i.e., Midori's MixColumns matrix [BBI+15]):

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

Note that such leakage amplifier modules are separated and never mixed with each other, which could potentially violate the independence assumption of the masking scheme. They

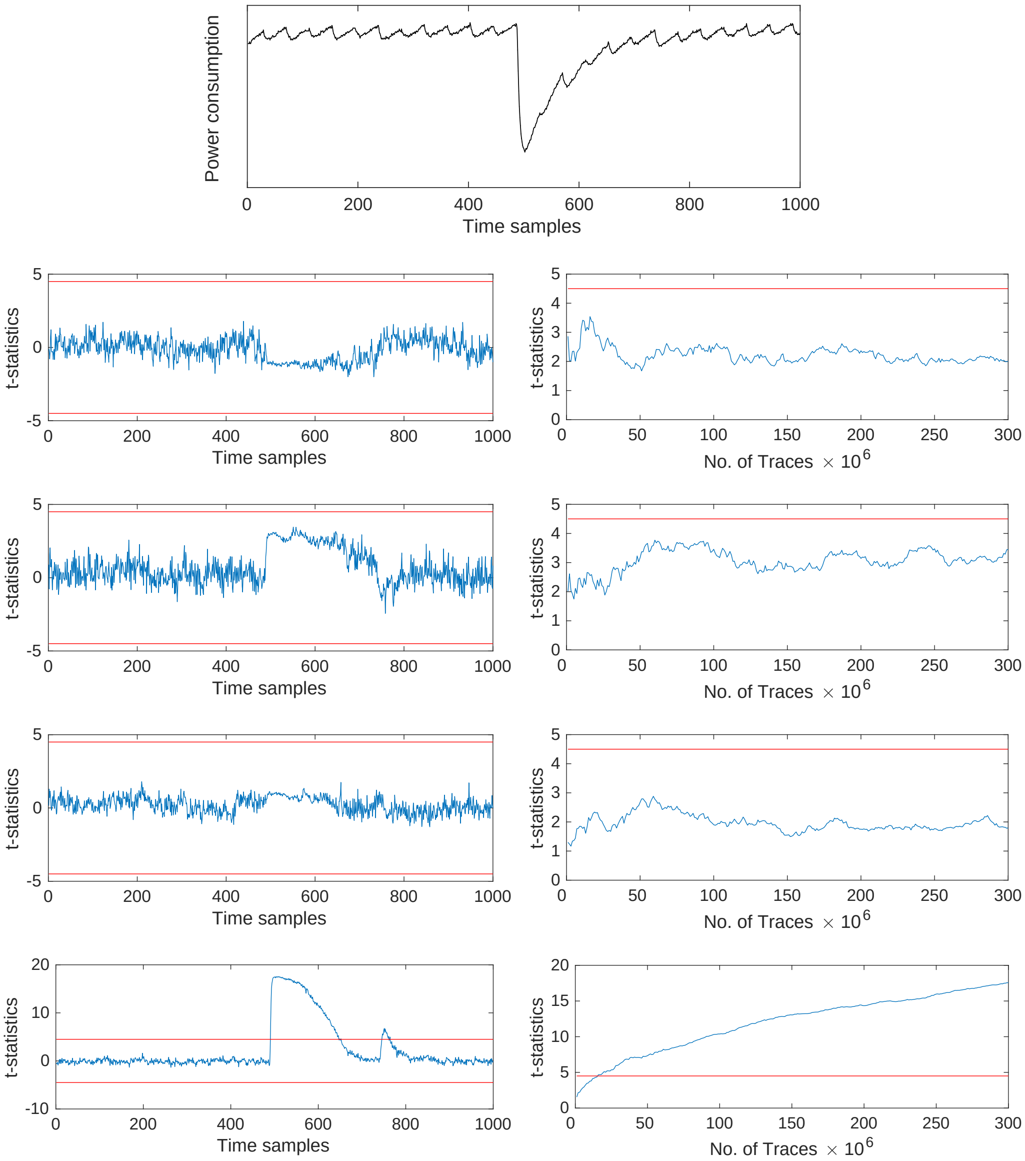
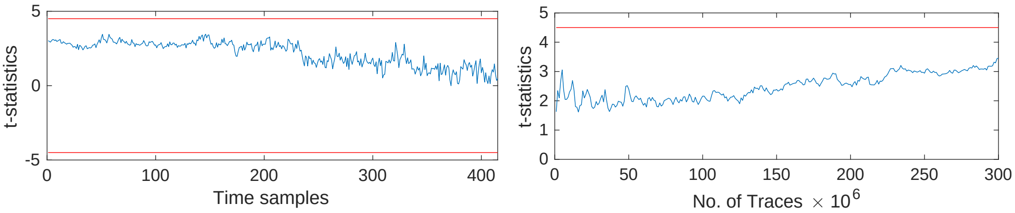


Figure 6: Sample power trace and univariate non-specific t-test results with 300 million measurements for a single $GF(2^4)$ multiplier masked by means of CMS with $d = 3$. The second to fifth rows show the t-statistics for the statistical moments 1 to 4, respectively. The left column depicts the t-values over time, the right column illustrates the evolution of the absolute maximum t-value over the number of traces.

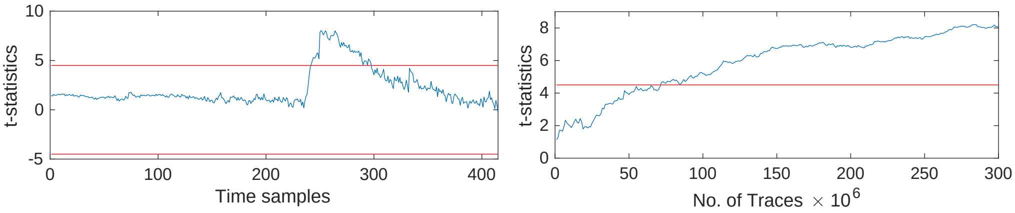
simply lead to a higher energy consumption depending on their corresponding input, which helps to achieve a higher SNR when the signal is much smaller than the noise level. As a result, the leakage corresponding to the third-order flaw becomes detectable.

8.1.2 DOM

Similar to the CMS experiments, we implemented the DOM-*dep* multiplier (shown in Figure 2 for the $d = 2$ case) by instantiating all multiplications as $GF(2^4)$ multipliers. We chose to perform the experimental verification for the $d = 3$ (i.e., $n = 4$) case, since the



(a) 2nd order, clocks $(t, t+1)$



(c) 3rd order, clocks $(t, t, t+1)$

Figure 7: Multivariate non-specific t-test results with 300 million measurements for a single $GF(2^4)$ multiplier masked by means of CMS with $d = 3$. The left column depicts the t-values over time, the right column illustrates the evolution of the absolute maximum t-value over the number of traces.

exploitation of this flaw imposes less restrictive constraints on the timing of the signals at the output of the register in the construction. Like before we could only detect the leakage by amplifying the power consumption of the probed values that are detailed in Section 4. Accordingly, one leakage amplifier is connected to $a_3 \cdot z_1$, another one to $a_4 \cdot z_2$. For the last one we implemented the XOR of the Decode operation in such a way that the XOR between the first two elements, i.e., $(a_1 + z_1) + (a_2 + z_2)$, is calculated before the third term is added. The output of this earlier evaluated XOR then supplies the third leakage amplifier module. Note that such a particular order of the aforementioned XORs does not violate the claims of the DOM-*dep* multiplier [GMK16] and could indeed occur in reality when synthesizing the construction.[3] As for the CMS case, we report univariate non-specific t-test results up to the fourth order in Figure 8, and multivariate analyses up to the third-order in Figure 9. It can be seen that the univariate fourth-order t-test and the multivariate third-order t-test indicate leakage with high-confidence ($t > 4.5$) when considering 500 million traces. Note that the corresponding offsets for the multivariate tests are identical to the ones previously outlined in the CMS case. Thus, the smallest data-dependent statistical moment is indeed the third one, which confirms the existence of the theoretically exhibited flaw.

Related work. Recently a first practical side-channel evaluation of a full block cipher (triple-DES) protected by domain-oriented masking has been published at COSADE 2018 [SH18]. This work makes extensive use of the DOM-*dep* multiplier to construct the DES substitution box and provides univariate t-test results with 50 million power traces taken from an FPGA implementation of the full cipher in the $d = 1$ case and 2 billion power traces in the $d = 2$ case. They come to the conclusion that their masked S-box indeed delivers the corresponding protection order promised by DOM. However, in view of our new results we assume that a multivariate analysis could have revealed a second-order

---

[3] Overall, we believe it is desirable that the security of a glitch-resistant gadget does not rely on the assumption that specific signal timings in the combinational paths are unlikely to occur, since this leads to security guarantees which can be falsified by physical defaults. The same is true for the example with $d = 2$ and two probes.

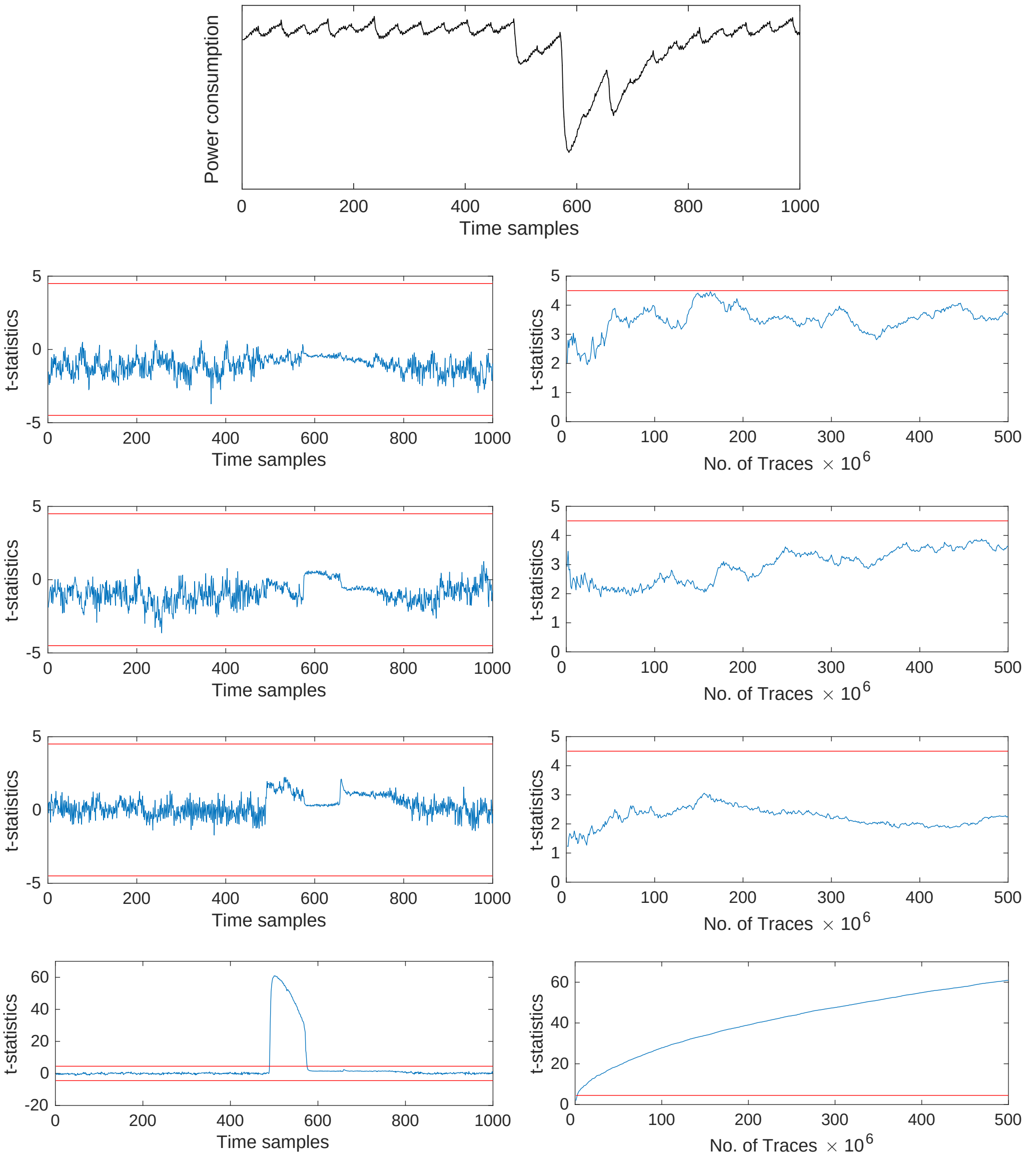
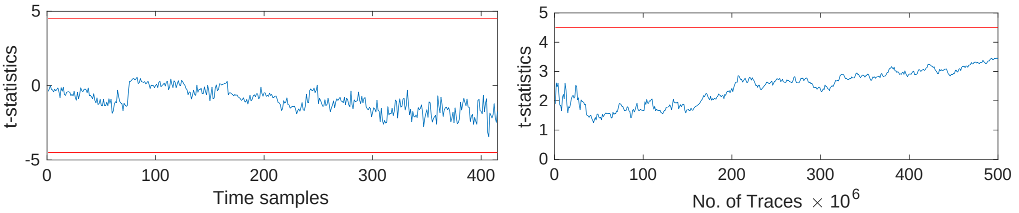


Figure 8: Sample power trace and univariate non-specific t-test results with 500 million measurements for a single DOM-*dep* multiplier with $d = 3$, based on $GF(2^4)$ multiplications. The second to fifth rows show the t-statistics for statistical moments 1 to 4.

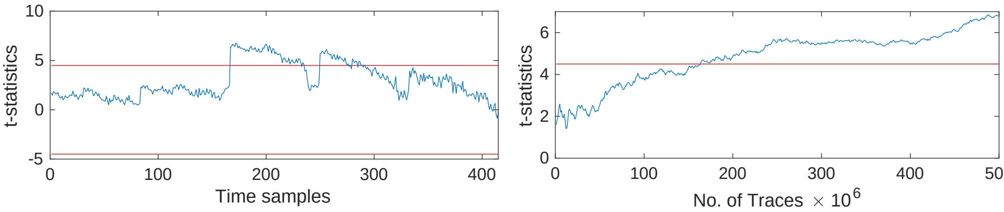
leakage in the $d = 2$ case, which is an interesting scope for further investigations.

8.1.3 UMA

In the case of UMA we do not evaluate a single instance to reveal the existence of a local flaw, but compose two multiplications (with $d = 2, n = 3$) as depicted in Figure 10 to show that UMA suffers from a lack of composability. Each of the multiplications (upper and lower half of the figure) consists of one half-complete block and the corresponding inner-domain terms (as detailed in [GM17]). Furthermore, the randomness optimizations by Belaïd et al. are in place. The registers which are depicted in black solid lines are mandatory by design. According to the authors of UMA, the green dashed registers are



(a) 2nd order, clocks $(t, t+1)$



(c) 3rd order, clocks $(t, t, t+1)$

Figure 9: Multivariate non-specific t-test results with 500 million measurements for a single DOM-*dep* multiplier with $d = 3$, based on $GF(2^4)$ multiplications.

optional pipeline registers; the red dashed registers are output synchronization stages to separate the multiplications from each other; the blue registers are optional pipelining registers due to the composition; and the purple registers are optional pipeline registers for the inner-domain terms (which are not considered by the authors of UMA). Each multiplication in Figure 10 depicts the instantiation of one $GF(2^4)$ multiplier. To show that UMA does not satisfy composability as ensured by the definition of strong non-interference (SNI) we consider the following scenario. If an adversary places one output probe on the leftmost output register of the first multiplication and one internal probe on the fresh randomness r_1^1 he can observe a joint distribution that depends on two shares of b and two shares of x and thus can not be simulated with only one share of each input. The latter leads to the attack with only two probes on the composed multiplications detailed in Section 5.

Probably caused by the lower number of shares compared to the previous experiments (or a stronger bias that is imposed by the flaw), we were not forced to use any leakage amplifiers or other particular considerations to detect the corresponding leakage. When applying all (mandatory and optional) registers that are included in Figure 10 the leakage corresponding to the composability flaw can directly be observed as multivariate second-order leakage. A sample trace and the univariate t-test results up to the third order are depicted in Figure 11. The multivariate second-order result can be seen in Figure 12. In this last experiment the multivariate leakage could not be observed in two consecutive clock cycles, but in sample points with an offset of 5 cycles. Figure 12(a) shows the resulting t-statistics curve when shifting this offset of 5 cycles over the whole 1000 time samples.

8.1.4 Discussion

In this section we have demonstrated that all of the exhibited flaws from Sections 3 to 6 are practically detectable in real-world power measurements, which effectively reduces the protection order of the corresponding schemes. However, our results do not imply that these flaws necessarily reduce the practical security level of full implementations instantiating these schemes. Admittedly, the biases caused by the flaws have a low amplitude and therefore may be hard to exploit in some cases. For example, for the concrete SNR and number of shares in our experiments, an exploitation of the univariate leakage in the

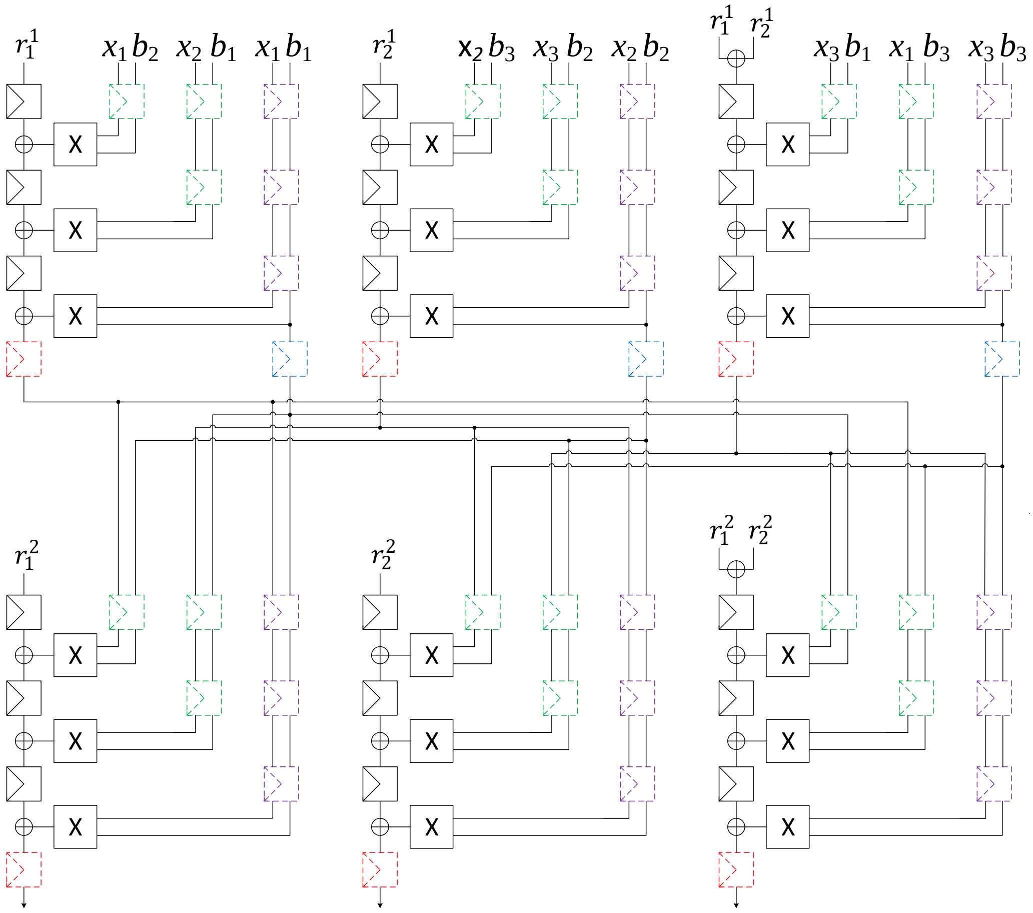


Figure 10: Composition of two UMA multiplications ($d = 2$) with several kinds of (mandatory and optional) pipelining and synchronization register stages.

$(d+1)$-th order will generally succeed with less traces than considering the multivariate d-th order leakage for an analysis. Yet, the reduction of the protection order raises doubts about higher noise levels and a larger number of shares (especially in the case of CMS, where the exploitation effort due to the flaw does not scale with the number of shares for orders $d > 2$). We note that our findings do not imply that it is impossible to construct d-probing secure circuits with the investigated gadgets. In case of UMA for example, the authors build a substitution box using the locally secure gadgets and verify the probing security of the composition by exhaustively analyzing the resulting circuit for small protection orders [GM18]. In this regard they make use of the recently introduced tool in [BGI$^+$18]. Such an approach is generally valid and can potentially lead to more efficient constructions than composing only SNI gadgets. However, the exhaustive analysis it performs still does not scale well for full implementations protected with a large number of shares.

8.2 Composability in Hardware - A Matter of Registers

As already mentioned, compositional security does not only depend on the amount of fresh randomness that is applied, but also on the correct instantiation of register stages in the composed circuits. While this is usually not an issue for software implementations, where all operations are inherently processed in a sequential manner, hardware implementations

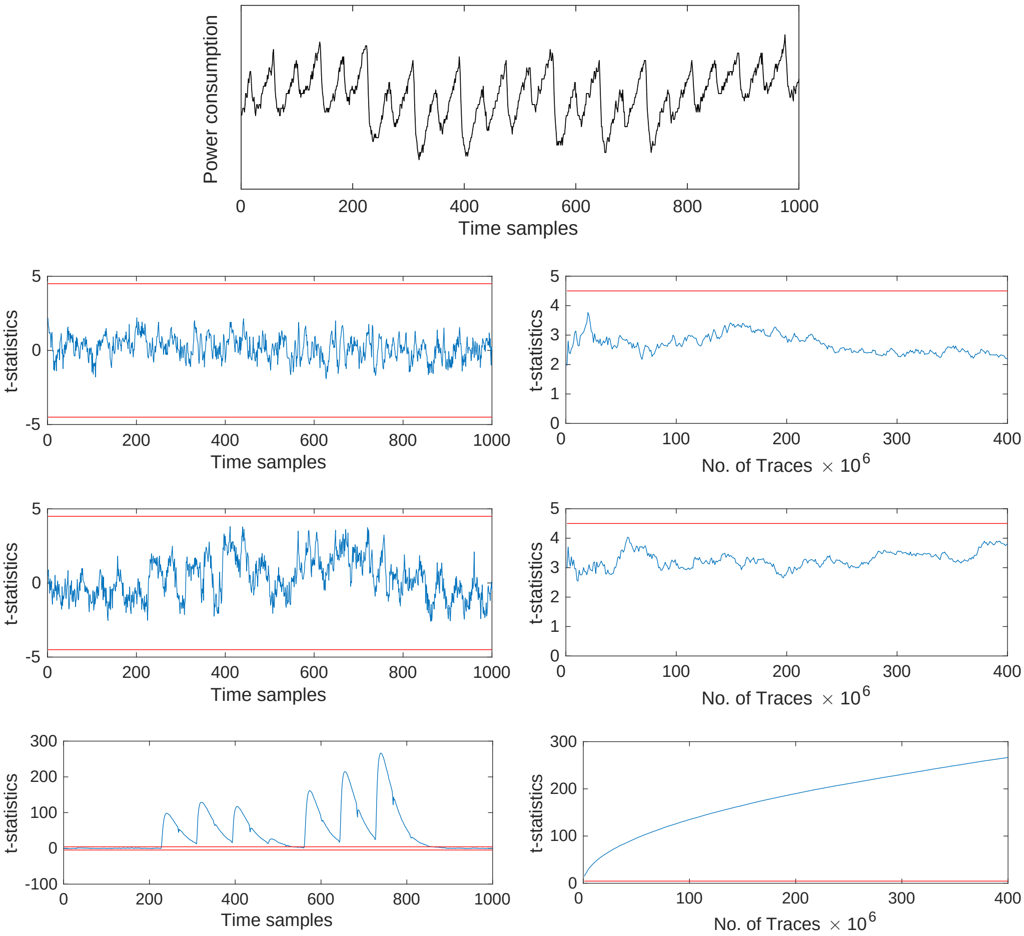


Figure 11: Sample power trace and univariate non-specific t-test results with 400 million measurements for two composed UMA multiplications with $d = 2$, based on $GF(2^4)$ multipliers. The second to fourth rows show the t-statistics for the statistical moments 1 to 3, respectively, arranged like before.

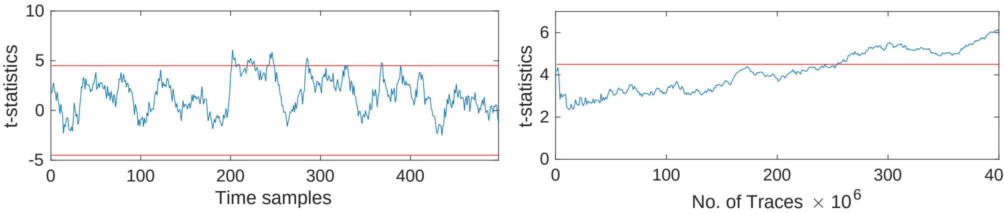


Figure 12: Multivariate second-order non-specific t-test results, clocks $(t, t+5)$, with 400 million measurements for two composed UMA multiplications with $d = 2$, based on $GF(2^4)$ multipliers, arranged like before.

offer a lot more freedom in terms of parallelization and order of operations. Thus, special care needs to be taken in order to not degrade the security of the whole implementation by an incorrect placement of memory elements. In the robust probing model, this is formalized by the fact that an adversary can always choose to probe an internal (glitchy) computation or a stable output, and only the latter ones are excluded from the probe count in the SNI definition. Combined with the fact that the “share fan-in” of a glitch-robust

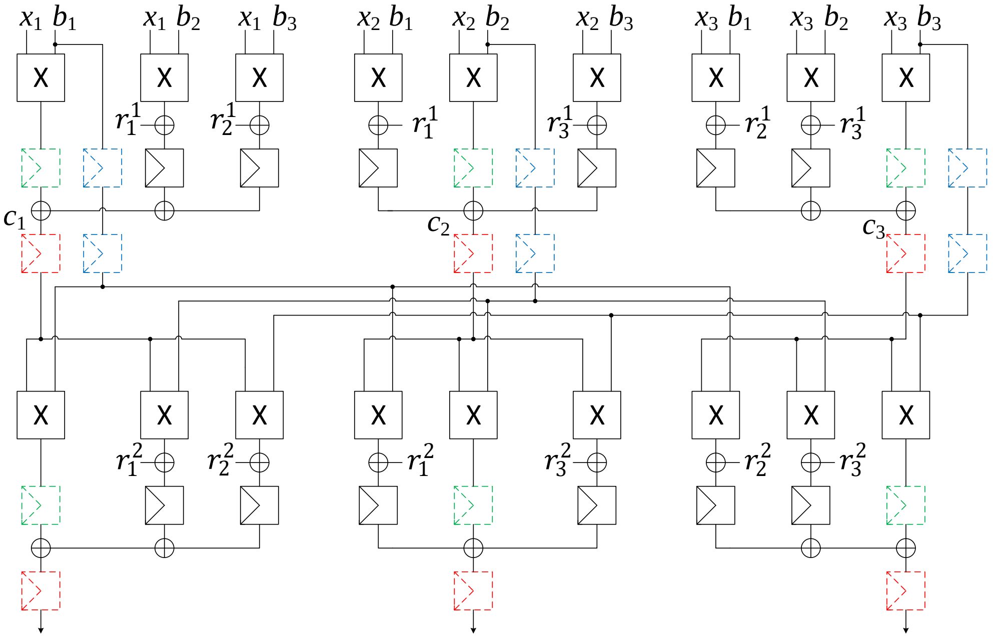


Figure 13: Composition of two DOM-*indep* multiplications (d = 2) with several kinds of (mandatory and optional) pipelining and synchronization register stages.

and composable multiplication should be minimum, it guided the design of the robust and composable multiplication algorithm and implementation in [FGP+18], which requires $(d+1)^2 + (d+1)$ registers to store the (refreshed) partial products and the final output.

8.2.1 DOM

As a case study, we take a look at the DOM-*indep* multiplier of the domain-oriented masking scheme, initially proposed in [GMK16]. The refresh layer (called resharing step in [GMK16]) of the DOM-*indep* multiplier is d-SNI. Furthermore, the full multiplier is d-probing secure in the presence of glitches. However, this is not sufficient to guarantee that any composition of DOM-*indep* multipliers leads to a d-probing secure (or d-SNI) circuit. In Figure 13, we have depicted such a composition of two DOM-*indep* multipliers for the $d = 2$ case (i.e., $n = 3$), where different possibilities for the inclusion of register stages are illustrated. Only the black solid registers are mandatory by design. In particular the green and red dashed registers are claimed to be optional (and not relevant for the security of the gadget) [GMK16]. We show in the following that especially the red dashed output registers which separate both multipliers from each other are in fact crucial for the compositional security. For this purpose, we have implemented the design in Figure 13, but left out the red output registers as well as the neighboring blue ones to ensure correct pipelining. With respect to the robust probing model, such an implementation violates the requirement that any composition of two gadgets with a limited share fan-in should be separated by memory elements [FGP+18]. Like before, the construction has been implemented based on $GF(2^4)$ multipliers. We acquired 500 million power traces suitable for a non-specific t-test evaluation. The results for the univariate case are shown in Figure 14. As illustrated in the figure, a significant univariate second-order leakage can be observed. To explain the source of this leakage, we consider one extended probe on the computation of the cross-product $c_1 \cdot b_2$, where c is the output of the upper DOM-*indep* multiplier. This probe

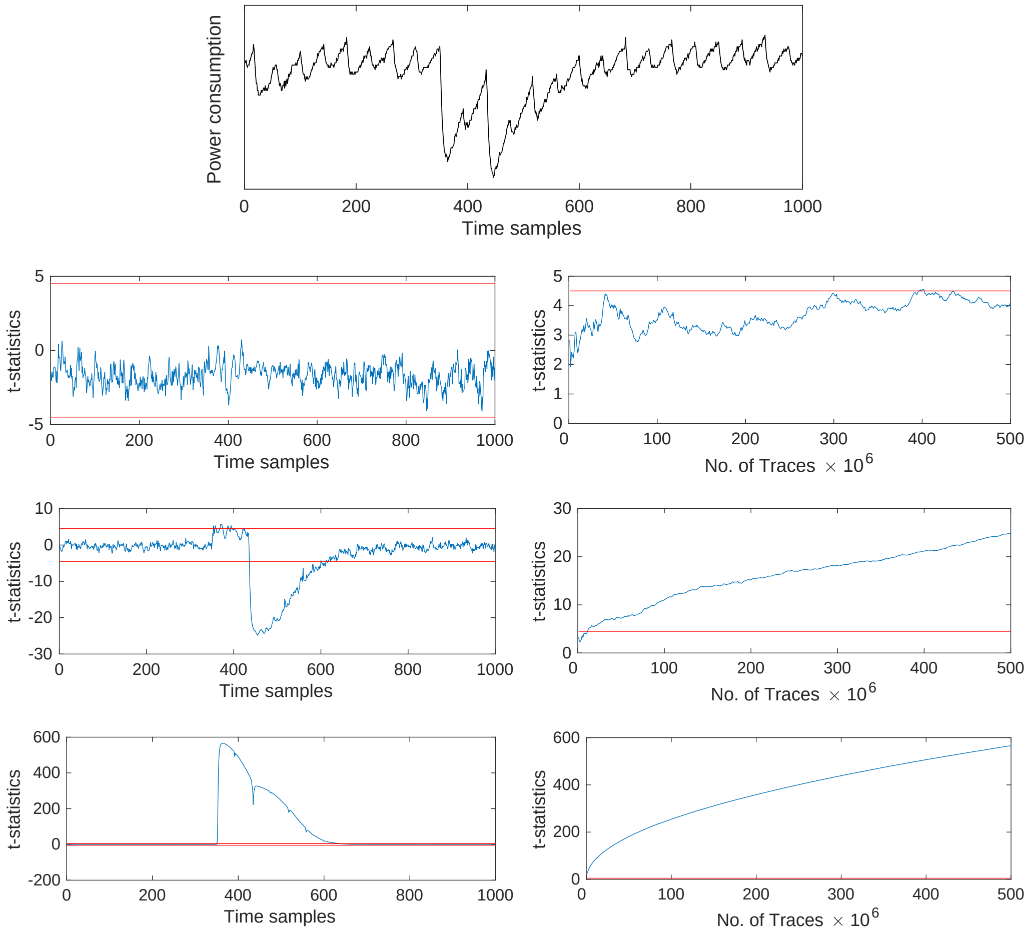


Figure 14: Sample power trace and univariate non-specific t-test results with 500 million measurements for two composed DOM-*indep* multiplications ($d = 2$) based on $GF(2^4)$ multipliers without output registers but with pipeline registers applied. The second to fourth row show the t-statistics for the statistical moments 1 to 3, respectively.

gives access to the following input variables:

$$P_{1,1} = b_2, \tag{32}$$
$$P_{1,2} = x_1 \cdot b_1, \tag{33}$$

$$P_{1,3} = x_1 \cdot b_2 + r_1^1, \tag{34}$$
$$P_{1,4} = x_1 \cdot b_3 + r_2^1. \tag{35}$$

Combining $P_{1,1}, P_{1,2}$ as:

$$P_1' = P_{1,1} + P_{1,2} = b_2 + x_1 \cdot b_1 \tag{36}$$

results in a distribution that depends on two shares of b already. Placing a second (regular) probe on b_3 (or an extended probe on some cross-product involving b_3) leads to a distribution depending on b, which results in (univariate) second-order leakage. An analogous attack with one probe exists for the $d = 1$ case.

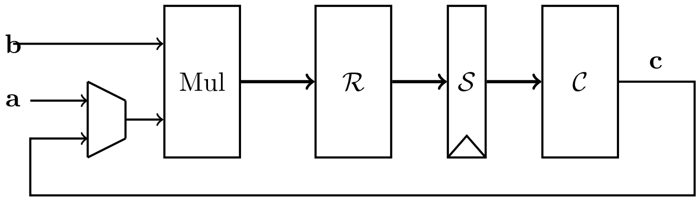


Figure 15: Iterative multiplication following the concept of GLM with refresh layer $\mathcal{R}$, synchronization layer $\mathcal{S}$, and compression layer $\mathcal{C}$.

Related work. The same article that already instantiated DOM-*dep* multipliers to protect a full triple-DES circuit also proposes to use DOM-*indep* multipliers in their construction [SH18]. Unfortunately, the authors make the mistake of composing DOM-*indep* multipliers without any output register in between (as apparent for example in Figure 3 of [SH18]). No practical side-channel analysis is presented for the implementations that make use of the DOM-*indep* multipliers. It is an interesting open question whether a univariate leakage in the d-th moment shows up in such a case, as in our experiments.

8.2.2 GLM

After having seen that the GLM scheme, explained in Section 6, is insecure for higher orders due to its instantiation of the CMS refresh layer, it might be tempting to simply replace the insufficient refreshing step by an SNI one, for example the DOM-*indep* refresh layer (especially since the authors of GLM specifically leave the search for a more suitable alternative open to future work [GIB18]). A simplified schematic of the GLM hardware design is shown in Figure 15. One can see that no register stage is placed after the compression layer. Due to the absence of this register stage, the just presented composability issues of the DOM-*indep* multiplication would arise, rendering the whole construction insecure. Adding such a register stage would on the one hand fix the security issue, but on the other hand also add an additional delay of one clock cycle per cross-product, which is not ideal for a low-latency construction. To confirm that including the output register indeed fixes the security problems, we have implemented the design in Figure 13 with all of the registers being present and measured another 500 million traces. A sample trace and the corresponding results can be seen in Figure 16 for the univariate case and in Figure 17 for the multivariate case. As expected, no leakage in the first two statistical moments can be observed, although admittedly the t-values in the multivariate second-order analysis come close to the 4.5 threshold. We observed those large t-values for an offset of 4 clock cycles and assume them to be a random occurrence.

8.2.3 Pipelining Registers

We further detail the relevance of pipelining registers for the security of multiplication gadgets in Appendix A, and show that they are not optional with case studies based on DOM and UMA. In this context as well, the main message is that in order to preserve robustness against glitches and composability jointly, it is needed to implement registers to separate all the refreshed partial product computations and the compressed output. As detailed in [FGP+18], the latter requires $(d+1)^2 + (d+1)$ registers for a 2-cycle multiplication, which is quite expensive. Interestingly, our conclusion for DOM and UMA is in fact identical. Finding solutions (or showing impossibility) with less registers (or randomness), is one more direction for future investigations.

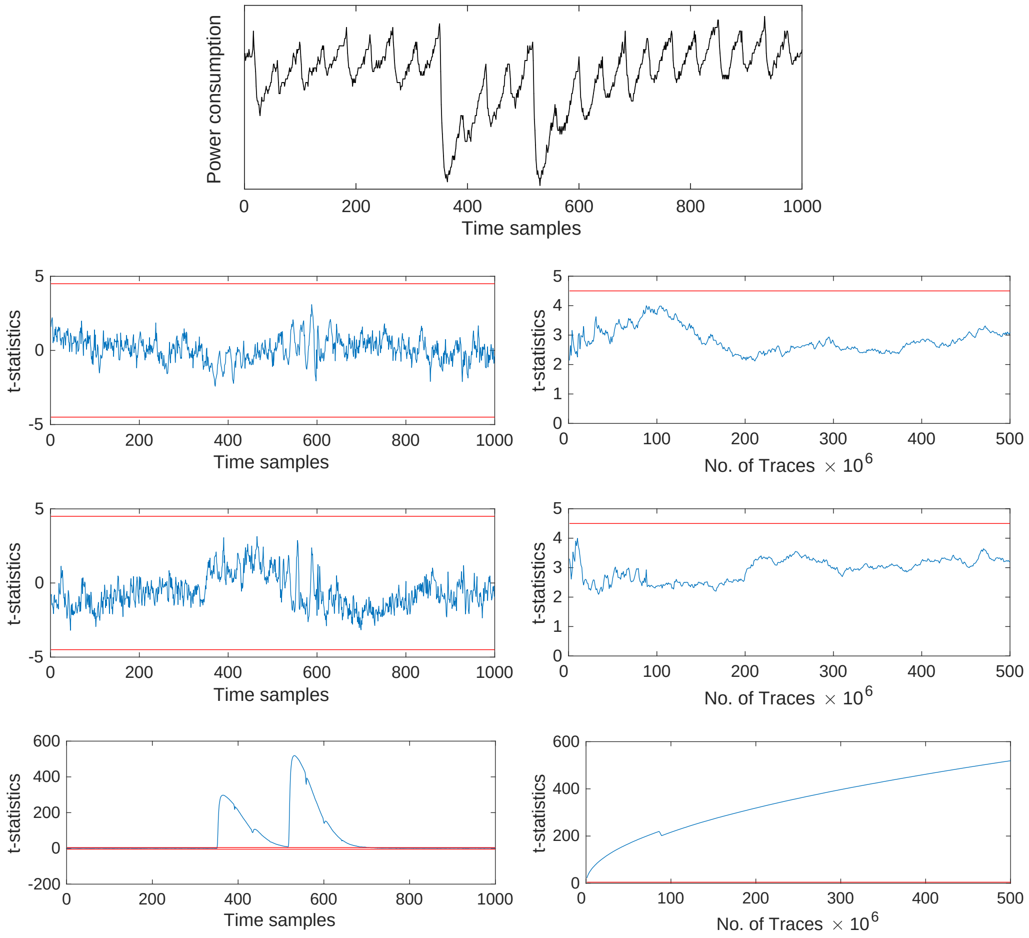


Figure 16: Sample power trace and univariate non-specific t-test results with 500 million measurements for two composed DOM-indep multiplications ($d = 2$) based on $GF(2^4)$ multipliers with all registers applied. The second to fourth row show the t-statistics for the statistical moments 1 to 3, respectively, arranged like before.

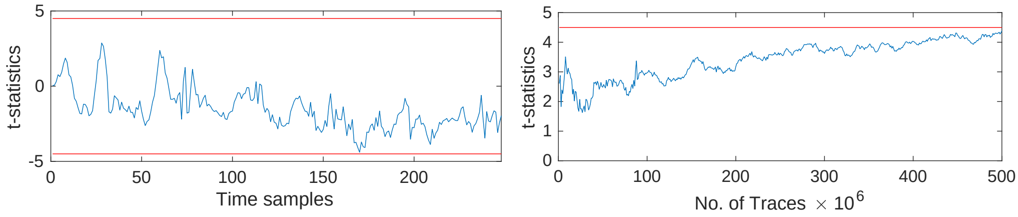


Figure 17: Multivariate second-order non-specific t-test results, clocks $(t, t+4)$, with 500 million measurements for two composed DOM-indep multiplications ($d = 2$) based on $GF(2^4)$ multipliers with all registers applied, arranged like before.

9 Further remarks and conclusions

In contrast to software-oriented masking, security proofs are not yet an established tool in hardware-oriented masking. One reason for this situation was the lack of an appropriate model that formally covers the local security and composability of masked gadgets in presence of physical defaults. As a result, engineering intuition and informal considerations of probing security with respect to glitches were often the only considered arguments supporting security claims of proposed masked circuits. The robust probing model in [FGP+18] now makes it possible to analyze and subsequently prove security guarantees of masked hardware gadgets. Our broad analysis of (scalable) hardware-oriented masking schemes revealed that not a single multiplication gadget which comes without a proof in the robust probing model actually delivers local *and* compositional security for arbitrary protection orders (at least when instantiated like proposed by the respective authors). This is confirmed by our empirical investigations, which showed that flaws with respect to the robust probing model can directly translate to exploitable leakage in real-world power measurements. Although the fact that these flaws lead to the most informative leakages depends on the implementations, it at least reveals an undesirable source of risk, especially as the claimed security order increases. In fact, only when tweaking the ISW multiplier [ISW03] in a way that makes it similar to the DOM-*indep* multiplier in [GMK16], but employing its pipeline registers and additionally storing its outputs in a further register stage, one ends up with a gadget that is SNI in the presence of glitches. This gadget was proposed and proven secure for arbitrary orders in [FGP+18] and additionally, it is the only $(d+1)$-masked multiplication circuit that did not exhibit detectable leakage up to the d-th order in our experiments.

Acknowledgments. The authors want to thank Hannes Groß and Stefan Mangard for their valuable feedback, especially for their help improving the flaw identified for DOM-*dep* by their tool from EUROCRYPT 2018 [BGI+18].
François-Xavier Standaert is a senior research associate of the Belgian Fund for Scientific Research (F.R.S.-FNRS). The work described in this paper has been supported in part by the European Commission through the ERC project 724725 (acronym SWORD), the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and through the DFG project 271752544 "NaSCA: Nano-Scale Side-Channel Analysis".

References

[ABP+18] Victor Arribas, Begül Bilgin, George Petrides, Svetla Nikova, and Vincent Rijmen. Rhythmic keccak: SCA security and low latency in HW. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):269–290, 2018.

[BBD+15] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, and Pierre-Yves Strub. Verified proofs of higher-order masking. In *EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 457–485. Springer, 2015.

[BBD+16] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. Strong non-interference and type-directed higher-order masking. In *CCS 2016*, pages 116–129. ACM, 2016.

[BBI+15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In *ASIACRYPT 2015*, volume 9453 of *LNCS*, pages 411–436. Springer, 2015.

[BBP+16] Sonia Belaïd, Fabrice Benhamouda, Alain Passelègue, Emmanuel Prouff, Adrian Thillard, and Damien Vergnaud. Randomness complexity of private circuits for multiplication. In *EUROCRYPT 2016*, volume 9666 of *LNCS*, pages 616–648. Springer, 2016.

[BDF+17] Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel implementations of masking schemes and the bounded moment leakage model. In *EUROCRYPT 2017*, volume 10210 of *LNCS*, pages 535–566, 2017.

[BGI+18] Roderick Bloem, Hannes Groß, Rinat Iusupov, Bettina Könighofer, Stefan Mangard, and Johannes Winter. Formal verification of masked hardware implementations in the presence of glitches. In *EUROCRYPT 2018*, volume 10821 of *LNCS*, pages 321–353. Springer, 2018. `http://eprint.iacr.org/2017/897`.

[BGN+14a] Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Higher-order threshold implementations. In *ASIACRYPT 2014*, volume 8874 of *LNCS*, pages 326–343. Springer, 2014.

[BGN+14b] Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. A more efficient AES threshold implementation. In *AFRICACRYPT 2014*, volume 8469 of *LNCS*, pages 267–284. Springer, 2014.

[CDG+13] Jeremy Cooper, Elke De Mulder, Gilbert Goodwill, Joshua Jaffe, Gary Kenworthy, and Pankaj Rohatgi. Test Vector Leakage Assessment (TVLA) Methodology in Practice. International Cryptographic Module Conference, 2013.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In *CRYPTO '99*, volume 1666 of *LNCS*, pages 398–412. Springer, 1999.

[Cor14] Jean-Sébastien Coron. Higher order masking of look-up tables. In *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 441–458. Springer, 2014.

[CPR07] Jean-Sébastien Coron, Emmanuel Prouff, and Matthieu Rivain. Side channel cryptanalysis of a higher order masking scheme. In *CHES 2007*, volume 4727 of *LNCS*, pages 28–44. Springer, 2007.

[CPRR14] Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-order side channel security and mask refreshing. In *FSE 2013*, volume 8424 of *LNCS*, pages 410–424. Springer, 2014.

[CRB+16] Thomas De Cnudde, Oscar Reparaz, Begül Bilgin, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Masking AES with d+1 shares in hardware. In *CHES 2016*, volume 9813 of *LNCS*, pages 194–212. Springer, 2016.

[Dae16] Joan Daemen. Spectral characterization of iterating lossy mappings. In *SPACE 2016*, volume 10076 of *LNCS*, pages 159–178. Springer, 2016.

[DDF14] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. In *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 423–440. Springer, 2014.

[De 18] Thomas De Cnudde. Cryptography Secured Against Side-Channel Attacks. PhD thesis, KULeuven, 2018.

[DFS15] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete - or how to evaluate the security of any leaking device. In *EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 401–429. Springer, 2015.

[EWS14] Hassan Eldib, Chao Wang, and Patrick Schaumont. Formal verification of software countermeasures against side-channel attacks. *ACM Trans. Softw. Eng. Methodol.*, 24(2):11:1–11:24, 2014.

[FGP+18] Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and François-Xavier Standaert. Composable Masking Schemes in the Presence of Physical Defaults and the Robust Probing Model. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):?, 2018.

[GC17] Ashrujit Ghoshal and Thomas De Cnudde. Several masked implementations of the boyar-peralta AES s-box. In *INDOCRYPT 2017*, volume 10698 of *LNCS*, pages 384–402. Springer, 2017.

[GIB18] Hannes Groß, Rinat Iusupov, and Roderick Bloem. Generic low-latency masking in hardware. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(2), 2018.

[GJJR11] Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. A testing methodology for Side channel resistance validation. In *NIST Non-invasive Attack Testing Workshop*, 2011.

[GM17] Hannes Groß and Stefan Mangard. Reconciling d+1 masking in hardware and software. In *CHES 2017*, volume 10529 of *LNCS*, pages 115–136. Springer, 2017.

[GM18] Hannes Groß and Stefan Mangard. A unified masking approach. *J. Cryptographic Engineering*, 8(2):109–124, 2018.

[GMK16] Hannes Groß, Stefan Mangard, and Thomas Korak. Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. In *TIS@CCS 2016*. ACM, 2016. http://eprint.iacr.org/2016/486.

[GMK17] Hannes Groß, Stefan Mangard, and Thomas Korak. An efficient side-channel protected AES implementation with arbitrary protection order. In *CT-RSA 2017*, volume 10159 of *LNCS*, pages 95–112. Springer, 2017.

[GR17] Dahmun Goudarzi and Matthieu Rivain. How fast can higher-order masking be in software? In *EUROCRYPT 2017*, volume 10210 of *LNCS*, pages 567–597, 2017.

[GT03] Jovan Dj. Golic and Christophe Tymen. Multiplicative masking and power analysis of AES. In *CHES 2002*, volume 2523 of *LNCS*, pages 198–212. Springer, 2003.

[ISW03] Yuval Ishai, Amit Sahai, and David A. Wagner. Private circuits: Securing hardware against probing attacks. In *CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481. Springer, 2003.

[MPG05] Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-channel leakage of masked CMOS gates. In *CT-RSA 2005*, volume 3376 of *LNCS*, pages 351–365. Springer, 2005.

[MPL+11] Amir Moradi, Axel Poschmann, San Ling, Christof Paar, and Huaxiong Wang. Pushing the limits: A very compact and a threshold implementation of AES. In *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 69–88. Springer, 2011.

[MPO05] Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully attacking masked AES hardware implementations. In *CHES 2005*, volume 3659 of *LNCS*, pages 157–171. Springer, 2005.

[MS16] Amir Moradi and Tobias Schneider. Side-channel analysis protection and low-latency in action - - case study of PRINCE and midori -. In *ASIACRYPT 2016*, volume 10031 of *LNCS*, pages 517–547, 2016.

[NRS11] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of nonlinear functions in the presence of glitches. *J. Cryptology*, 24(2):292–321, 2011.

[PMK+11] Axel Poschmann, Amir Moradi, Khoongming Khoo, Chu-Wee Lim, Huaxiong Wang, and San Ling. Side-channel resistant crypto for less than 2, 300 GE. *J. Cryptology*, 24(2):322–345, 2011.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 142–159. Springer, 2013.

[RBN+15] Oscar Reparaz, Begül Bilgin, Svetla Nikova, Benedikt Gierlichs, and Ingrid Verbauwhede. Consolidating masking schemes. In *CRYPTO 2015*, volume 9215 of *LNCS*, pages 764–783. Springer, 2015.

[Rep15] Oscar Reparaz. A note on the security of higher-order threshold implementations. *IACR Cryptology ePrint Archive*, 2015:1, 2015.

[Rep16] Oscar Reparaz. Detecting flawed masking schemes with leakage detection tests. In *FSE 2016*, volume 9783 of *LNCS*, pages 204–222. Springer, 2016.

[RP10] Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of AES. In *CHES 2010*, volume 6225 of *LNCS*, pages 413–427. Springer, 2010.

[sak] Side-channel AttacK User Reference Architecture. `http://satoh.cs.uec.ac.jp/SAKURA/index.html`.

[SH18] Pascal Sasdrich and Michael Hutter. Protecting triple-des against DPA - A practical application of domain-oriented masking. In *COSADE 2018*, volume 10815 of *LNCS*, pages 207–226. Springer, 2018.

[SM15] Tobias Schneider and Amir Moradi. Leakage assessment methodology - A clear roadmap for side-channel evaluations. In *CHES 2015*, volume 9293 of *LNCS*, pages 495–513. Springer, 2015.

[SM16] Tobias Schneider and Amir Moradi. Leakage assessment methodology - extended version. *J. Cryptographic Engineering*, 6(2):85–99, 2016.

[SP06] Kai Schramm and Christof Paar. Higher order masking of the AES. In *CT-RSA 2006*, volume 3860 of *LNCS*, pages 208–225. Springer, 2006.

A On the Need of (Pipelining) Registers

In the second part of Section 8 we have shown that register stages are crucial ingredients for the construction of composable gadgets. However, up to this part it was only demonstrated that a lack of output registers can have serious consequences on the compositional security of locally secure gadgets. In this appendix, we provide experimental evidence for the fact that, even when output registers are employed, a lack of pipelining stages can lead to a reduction of the protection order as well. We illustrate this claim with UMA and DOM.

The starting point of our analysis is the illustration of Figure 10, where one could assume that the depicted registers are supposed to be enabled all at once and then be kept active for a determined number of clock cycles until the correct results are stable at the output. One might also assume that the registers are reset (e.g., to zero) before applying new values to the inputs, as it is a usual practice in hardware design (especially when pipelining is not a considered use case). We show that it is not possible to safely make those assumptions when composing UMA multiplications and argue that the same is true for the DOM-*indep* multiplier when implemented without pipelining registers.

In the case of UMA no pipelining registers (not even optional ones) are included in the paths for the inner-domain terms. Thus, the purple dashed registers in Figure 10 are not present in the UMA scheme as proposed by the authors. Accordingly, the result of the inner-domain terms will propagate to the output of a UMA gadget first. Let us assume for a moment that a state machine controlling this circuit iterates over the following three simple states. At first all registers are reset to zero, then the shared multiplication inputs are applied to the inputs of the circuit and afterwards all registers are enabled for 8 consecutive clock cycles. In this case $x_1 \cdot b_1$, $x_2 \cdot b_2$ and $x_3 \cdot b_3$ are evaluated right after the inputs are applied. Before being saved into the output register of the first multiplier these values are input to an XOR with zero (due to the reset of the registers), which does not change their value. Accordingly, after being enabled for one clock cycle, $x_1 \cdot b_1$, $x_2 \cdot b_2$ and $x_3 \cdot b_3$ are propagated to the second multiplier, where they are multiplied with all shares of b individually (i.e., two shares of b are combined in each multiplication without proper resharing). Thus, trivial univariate second-order leakage emerges due to the early propagation of partial results. This is confirmed by the univariate non-specific t-tests considering 200 million power traces in Figure 18, where we also performed a reset of all registers before each multiplication. When taking a look at the DOM-*indep* multiplier in Figure 13, it appears that omitting the green (and neighboring blue) registers leads to the same problem. To demonstrate this we also measured 100 million power traces of two composed DOM-*indep* multipliers without pipelining registers, but with output registers. The results of the t-test are depicted in Figure 19.

In fact in both of those cases the leakage is even more drastic when additionally removing the output registers as well. We have verified this again with experiments (100 million traces each), as apparent in Figures 20 and 21. We insist that we do not claim these bad combinations are the only possible ones. We just mean that the authors' guidelines are not strictly sufficient to avoid these issues. For example, if the gadgets are not as directly connected as in our examples, but with synchronization stages and other modules in between, these problems may not arise. Furthermore, in all the presented cases the security issues can easily be fixed by putting additional constraints on the registers that have to be observed by a state machine (e.g., not allowing a reset; only activating the output register after a certain number of clock cycles; not propagating b to the second multiplier before the first one is finished; ...). However, our results highlight that without an explicit guideline on how to treat the register stages, it is strongly advised to use fully pipelined circuits, even when pipelining is not a considered use case, in order to mitigate the early propagation of partial results. This directly complies to the fact that the so

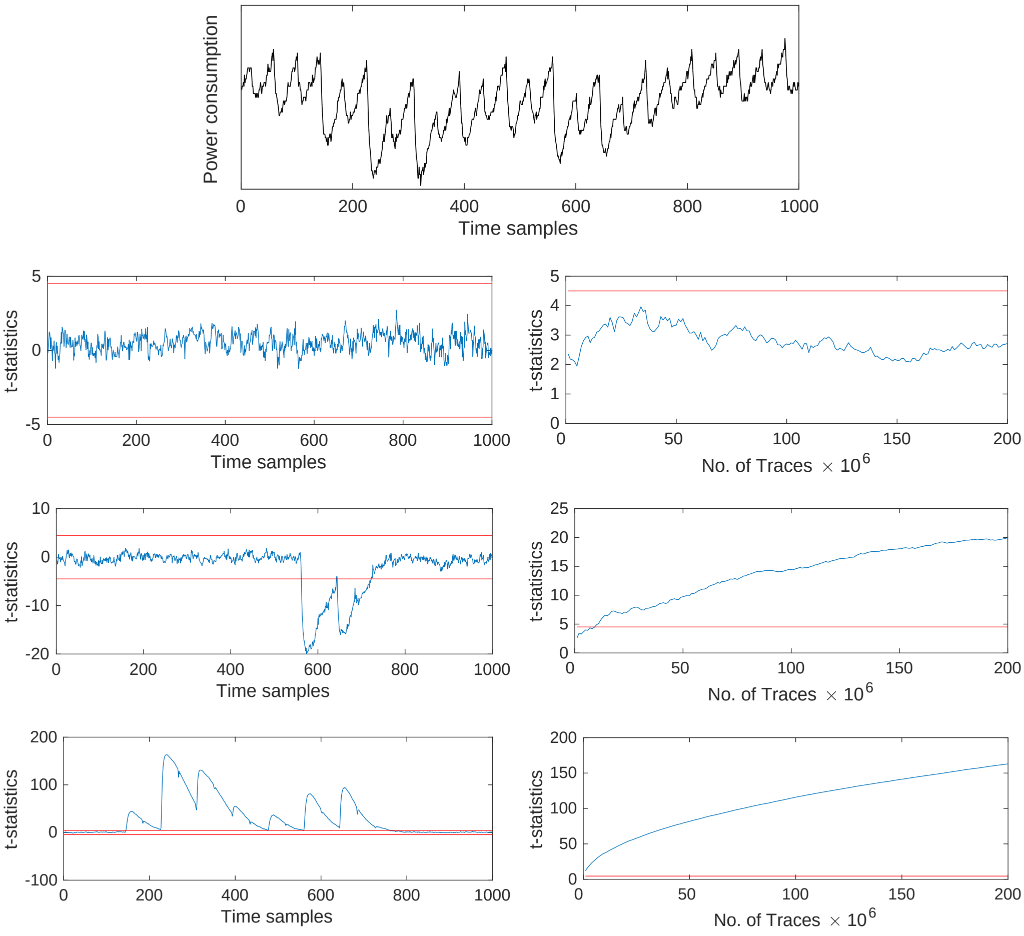


Figure 18: Sample power trace and univariate non-specific t-test results with 200 million measurements for two composed UMA multiplications ($d = 2$) based on $GF(2^4)$ multipliers without pipeline registers for the inner-domain terms. The second to fourth row show the t-statistics for the statistical moments 1 to 3, respectively.

far only multiplication gadget which has been proven secure in the robust probing model requires $(d + 1)^2$ registers to store the (refreshed) cross-products, and $(d + 1)$ registers to store the shared multiplication output. In this case no specific constraints have to be set on the registers and the gadget is suitable for all reasonable use cases.

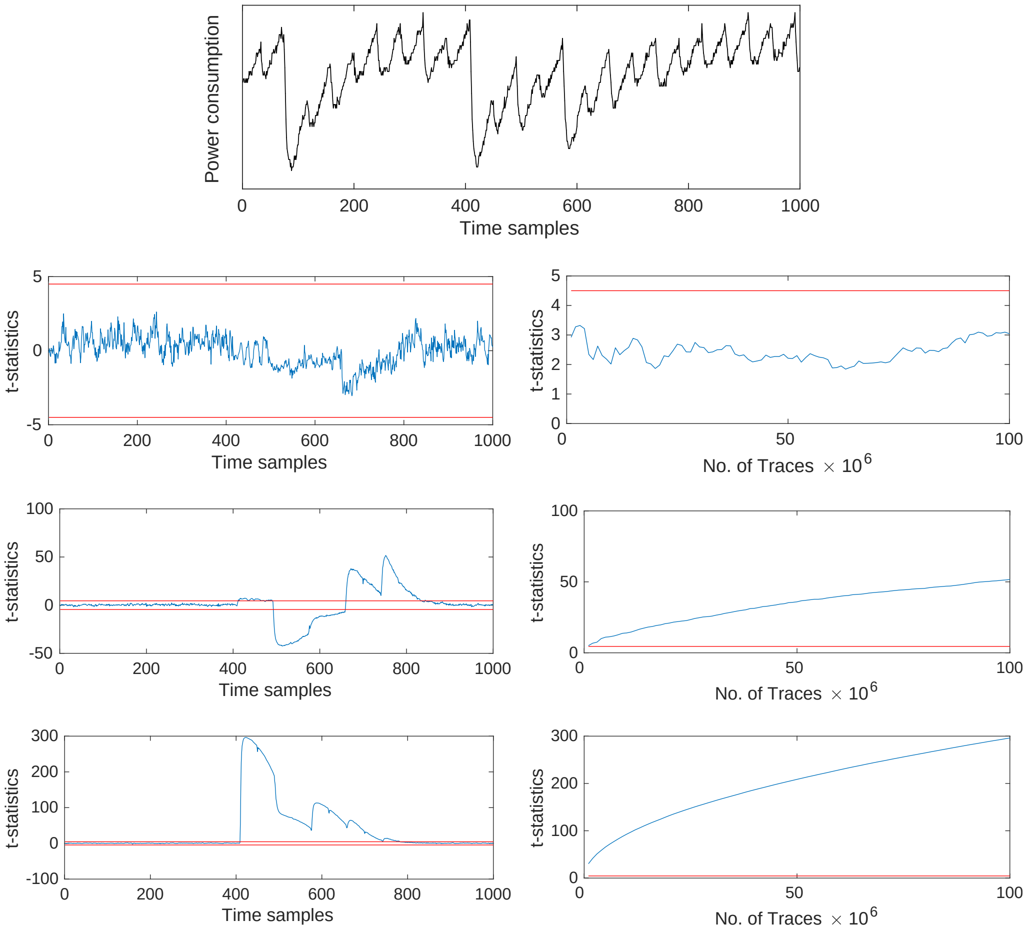


Figure 19: Sample power trace and univariate non-specific t-test results with 100 million measurements for two composed DOM-*indep* multiplications ($d = 2$) based on $GF(2^4)$ multipliers with output registers but without pipeline registers applied. The second to fourth row show the t-statistics for the statistical moments 1 to 3.

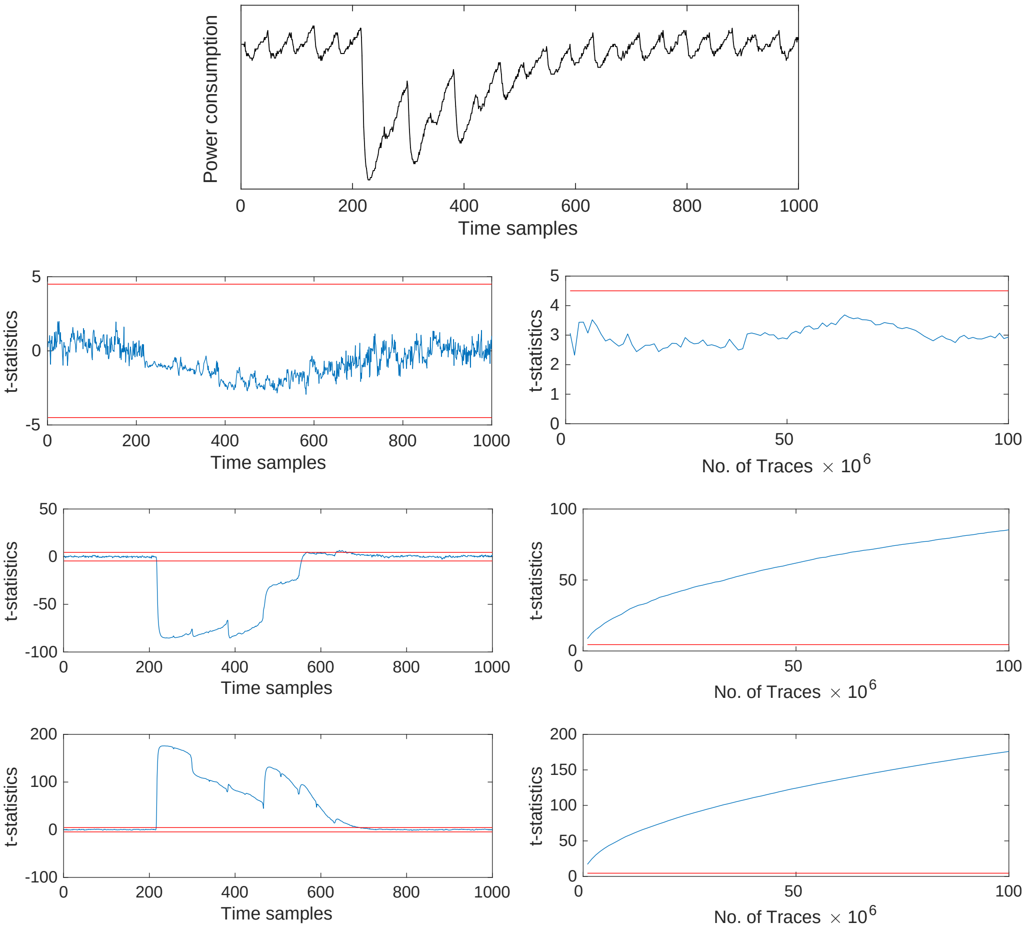


Figure 20: Sample power trace and univariate non-specific t-test results with 100 million measurements for two composed UMA multiplications ($d = 2$) based on $GF(2^4)$ multipliers with only the mandatory registers applied. The second to fourth row show the t-statistics for the statistical moments 1 to 3, respectively, arranged like before.

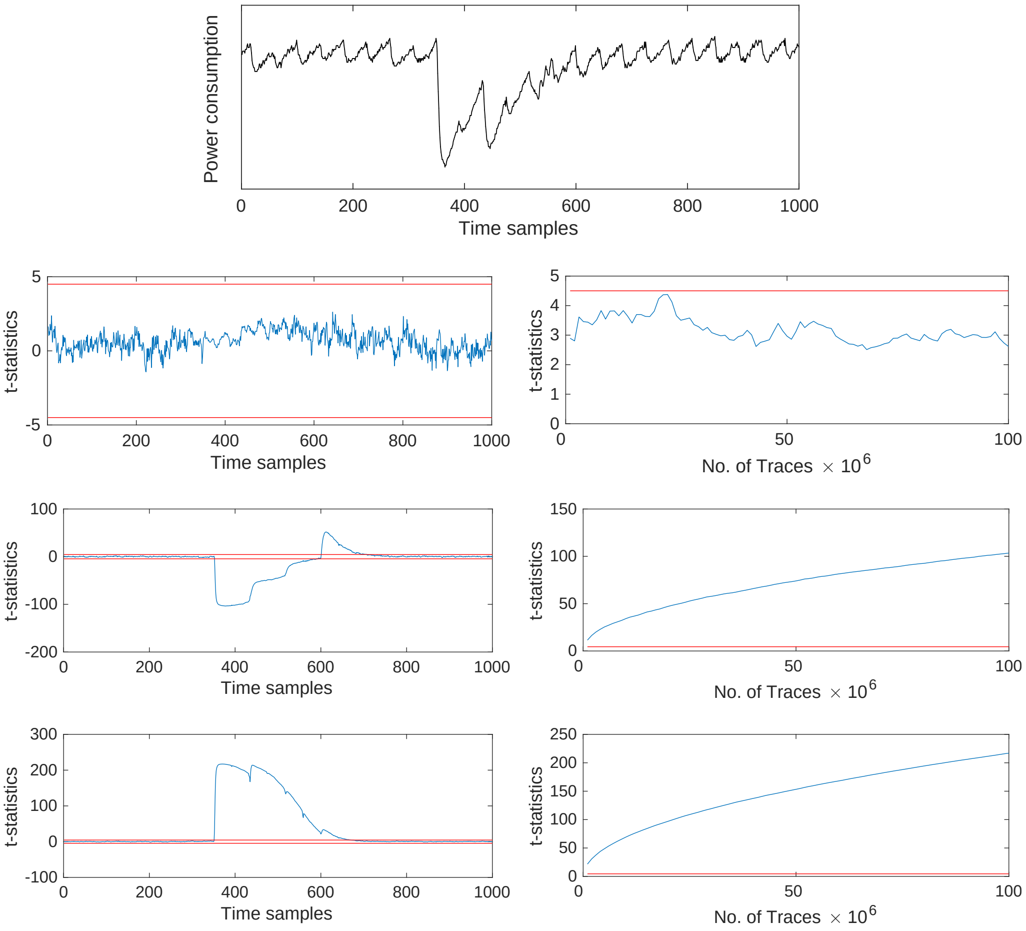


Figure 21: Sample power trace and univariate non-specific t-test results with 100 million measurements for two composed DOM-*indep* multiplications ($d = 2$) based on $GF(2^4)$ multipliers with only the mandatory registers applied. The second to fourth row show the t-statistics for the statistical moments 1 to 3, respectively, arranged like before.

4.3 Deep Learning Leakage Assessment

Publication Data

> Thorben Moos, Felix Wegener, and Amir Moradi. DL-LA: deep learning leakage assessment A modern roadmap for SCA evaluations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):552–598, 2021

The acceptance rate for Volume 2021 of the IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES) was **31,2%** [Acca].

Content This work explores whether deep learning can be used as an effective instrument to improve the performance of side-channel security evaluations or reduce their manual effort in complex scenarios. The developed approach for leakage assessment with deep neural networks is based on the concept of supervised learning. In particular, a network is trained on a sequence of labeled side-channel measurements recorded for different inputs to the targeted cryptographic implementation. Afterwards, the trained classifier is supposed to correctly categorize further unlabeled measurements based on the learned information such as generalizable data-dependencies. It is shown that this evaluation methodology is able to deal with many different types of side-channel leakage without any specific pre-processing or manual selection of points and that the provided confidence in the detected leakage significantly exceeds that of traditional approaches.

Contribution The author of this thesis is a principal author of this publication. In particular, the general idea of the work, the acquisition of the measurement data for almost all case studies, the evaluation of the univariate and multivariate t-test and χ^2-test results and a significant share of the DL-LA experiments and their evaluation have been contributed by the author of this thesis. The author also contributed substantially to the writing and the presentation of the results. The author would like to thank both co-authors for their substantial contributions to the network selection and the definition of the metrics and the methodology presented in this work.

DL-LA: Deep Learning Leakage Assessment

A modern roadmap for SCA evaluations

Thorben Moos*, Felix Wegener* and Amir Moradi

Ruhr University Bochum, Horst Görtz Institute for IT Security, Bochum, Germany
firstname.lastname@rub.de

Abstract. In recent years, deep learning has become an attractive ingredient to side-channel analysis (SCA) due to its potential to improve the success probability or enhance the performance of certain frequently executed tasks. One task that is commonly assisted by machine learning techniques is the profiling of a device's leakage behavior in order to carry out a template attack. At CHES 2019, deep learning has also been applied to non-profiled scenarios for the first time, extending its reach within SCA beyond template attacks. The proposed method, called DDLA, has some tempting advantages over traditional SCA due to merits inherited from (convolutional) neural networks. Most notably, it greatly reduces the need for pre-processing steps when the SCA traces are misaligned or when the leakage is of a multivariate nature. However, similar to traditional attack scenarios the success of this approach highly depends on the correct choice of a leakage model and the intermediate value to target. In this work we explore, for the first time in literature, whether deep learning can similarly be used as an instrument to advance another crucial (non-profiled) discipline of SCA which is inherently independent of leakage models and targeted intermediates, namely leakage assessment. In fact, given the simple classification-based nature of common leakage assessment techniques, in particular distinguishing two groups fixed-vs-random or fixed-vs-fixed, it comes as a surprise that machine learning has not been brought into this context, yet. Our contribution is the development of the first full leakage assessment methodology based on deep learning. It gives the evaluator the freedom to not worry about location, alignment and statistical order of the leakages and easily covers multivariate and horizontal patterns as well. We test our approach against a number of case studies based on FPGA, ASIC and µC implementations of the PRESENT block cipher, equipped with state-of-the-art SCA countermeasures. Our results clearly show that the proposed methodology and network structures are robust across all case studies and outperform the classical detection approaches (t-test and χ^2-test) in all considered scenarios.

Keywords: Leakage Evaluation · Side-Channel Analysis · Deep Learning

1 Introduction

In an ideal world, side-channel security evaluations would be able to provide a qualitative and confident answer (pass or fail) to the question whether the device under test (DUT) is vulnerable to physical attacks or not. However, the past has shown that this expectation is indeed a utopia. An exhaustive verification of the security of a DUT against all possible attack vectors is simply infeasible. Instead, the concept of leakage assessment has been introduced in order to answer a slightly, but explicitly, less informative question. Namely, whether any kind of input-dependent information can be detected in side-channel measurements of the device under test. Clearly, in case this question is answered positively,

---

*These authors contributed equally to this work.

no conclusions about the actual vulnerability of the device with respect to key recovery attacks can be drawn (although it is sometimes interpreted as an indication thereof). Yet, in case it is answered negatively (and no *false* negative occurs) the DUT should be sufficiently secure. In other words, leakage assessment is conceptually capable of providing the initially desired confidence in at least one of the two cases. This possibility inspired the quest for appropriate leakage assessment methods in academia and industry.

The most prominent leakage detection approach is certainly distinguishing two groups of measurements, one for fixed and one for random inputs, by means of the Welch's *t*-test [GJJR11, SM15]. Whenever these two groups are distinguishable with confidence one can conclude that the device reveals input-dependent information. However, this method has some severe limitations, especially when more sophisticated types of side-channel leakage need to be captured. First of all, since each point in time is evaluated independently, the approach inherently expects that any potential side-channel leakage is of a univariate nature and, more generally, that the detection of the leakage does not benefit from a combination of multiple points. Yet, many counterexamples to this assumption can be observed in reality. Although Schneider *et al.* [SM15] provide detailed information on how to perform the *t*-test at arbitrary order and variate, the performance of the test at higher variates either quickly runs into feasibility issues or its success depends highly on the expertise of the evaluator and the prior knowledge about the underlying implementation. On another note, a misalignment of the leaking samples between the individual traces leads to a significantly impaired detection as well. Thus, the Welch's *t*-test, as it is currently applied as a test vector leakage assessment (TVLA) methodology, is naturally unsuited to cover multivariate and horizontal leakages, as well as (heavily) misaligned traces. In addition to that, it has recently been pointed out that the separation of statistical orders, which is often seen as a beneficial feature of the *t*-test when seeking the smallest key dependent moment for example, may cause false negatives. This can be observed when masked implementations with (very) low noise levels are analyzed [Sta18, Moo19] or when the leakage is distributed over multiple statistical moments (as it is common for hardware masking schemes like Threshold Implementations) [MRSS18, Sta18]. Moradi *et al.* [MRSS18] suggested Pearson's χ^2-test as a natural complement to the Welch's *t*-test to aggregate leakages distributed over multiple orders and to analyze the joint information. By combining the two approaches the risk of false negatives, especially in the previously described cases, can significantly be reduced. Yet, in the same manner as the *t*-test, the χ^2-test analyzes the individual points in a leakage trace independently and therefore suffers from the same shortcomings when it comes to multivariate or horizontal patterns and misalignments. The core motivation for this work has been to extend the state of the art in such a way that the latter types of leakage can be covered, using a simple and easy to apply methodology.

Deep learning has been brought into the context of side-channel analysis mainly in order to improve the effectiveness of template attacks [HGM$^+$11]. In a template attack the adversary is in possession of a fully-controlled profiling device, learns the leakage function of a certain cryptographic operation and subsequently uses the acquired knowledge to reveal sensitive information on a structurally identical but distinct target device where the secrets are unknown. Apart from the general suitability of deep learning to build classifiers for profiled side-channel attacks, it has also been demonstrated that certain features and structures of the applied neural networks offer valuable advantages over classical template attacks. For example, it has been shown in [CDP17] that convolutional neural networks (CNNs) can lead to efficient classifiers even when the available side-channel traces suffer from a misalignment. Thus, due to their so-called translation invariance property, CNNs can be utilized to conquer jitter-based countermeasures. Recently, the first non-profiled deep learning based side-channel attacks have been demonstrated in literature [Tim19]. The proposed method, called DDLA, is based on guessing a part of the key, using it to compute the targeted key-dependent intermediate value, applying a leakage model and labeling the training data according to its result. Assuming that under the correct key hypothesis the differences between the classes implied by the leakage model correlate with the measured leakage traces (and for the incorrect guesses they do not), the impact of the correct key guess on the training loss and the training accuracy is visible and can easily be identified. Although, this approach depends on the correct choice of the targeted intermediate value and the applied leakage model as much as traditional attacks, it offers some tempting advantages. First of all, in case CNNs are used, the translation invariance property allows to analyze misaligned traces without any pre-processing. Secondly, when the leakage is of a multivariate nature or generally distributed over multiple points no recombination and no prior knowledge about the underlying implementation is required. Hence, deep learning is a powerful tool for non-profiled scenarios as well.

1.1 Our Contribution

For the first time in literature we evaluate whether deep learning is an eligible strategy for black box leakage detection. To this end, we have developed an approach that is based on the concept of supervised learning. We call it deep learning leakage assessment (DL-LA) in the following. Simply put, we train a neural network with a randomly interleaved sequence of labeled side-channel measurements that have been acquired while supplying the DUT with one of two distinct fixed inputs (fixed-vs-fixed). Afterwards, in the validation phase, the trained network is supplied with unlabeled measurements from both groups and supposed to correctly classify them. Of course, the training set and the validation set are disjoint. In case the network succeeds with a higher percentage of correct classifications than could be achieved by a randomly guessing binary classifier with a non-negligible probability, it can be concluded that indeed enough information was included in the *training* set to distinguish the two groups. In other words, given the percentage of correctly classified traces and the size of the validation set one can easily calculate a confidence value, i.e., a probability, that the correct classifications were not just a random statistical occurrence. In this way it is possible to directly compare the confidence values achieved by DL-LA with the confidence provided by classical leakage assessment approaches, such as the Welch's t-test and Pearson's χ^2-test. Aligning the appearance of DL-LA results to previous approaches is a valuable characteristic of our methodology and a contribution of this work.

Classical hypothesis tests rely on clearly defined formulas to estimate their statistics. DL-LA on the other hand grants a high level of freedom regarding its application due to the flexible choice of the network that shall be trained as a classifier. However, that freedom does not come without drawbacks. A complex network with a highly successful performance as a classifier on a certain set of measurements may not deliver satisfactory results on another set. Hence, the selection of the network may have a huge impact on the success of the DL-LA procedure. In such a case, an evaluator may be required to repeat the evaluation using many different networks to gain a confident result or choose suitable parameters based on prior knowledge about the underlying implementation. Clearly, both scenarios are undesirable for an initial analysis. Thus, in order to qualify as a simple and generic strategy for leakage assessment, it should be possible to select networks which offer a fairly robust and universal performance. In particular, it is desirable that the leakage detection capability of a network is as independent of the type of side-channel leakage to be detected as possible and largely independent of exterior parameters such as the trace length as well. This approach stands in stark contrast to common applications of deep learning in the area of physical security evaluations. Usually a network is deliberately tailored to a specific leakage pattern and measurement set in order to provide the best possible results. In our case, however, we want to use networks, at least for the initial analysis, that are general enough to deliver appropriate results when facing many different measurement sets and leakage behaviors. Thus, an important contribution of this work is the identification of network structures that perform consistently well when faced with different types of side-channel leakage and characteristics of the traces. After a long process of evaluating different networks (manually and automated) we have come to the conclusion that simplicity beats complexity. We evaluate and recommend two different simple network architectures, 1) a multi-layer perceptron (MLP) and 2) a convolutional neural network (CNN), the first with a given set of hyper-parameters and the second with a spectrum of hyper-parameters that proved successful. We are able to show that both networks provide excellent detection performance in a total of nine different case studies analyzing three distinct implementation platforms (FPGA, custom ASIC, and ARM Cortex-M0 µC). Each of the case studies features implementations of the PRESENT ultra-lightweight block cipher with different variations of masking and hiding countermeasures applied. The classification capability of our networks does not only withstand misaligned and noisy traces, but is able to deal with univariate and multivariate higher-order leakage as well. In all nine case studies we compare the success of our method to both the Welch's t-test and Pearson's χ^2-test and show that DL-LA outperforms the leakage assessment capabilities of the classical techniques in all considered scenarios (either by requiring fewer traces for confident detection or by providing a higher confidence using the same number of traces for detection[1]). We also present one scenario where both the univariate and the multivariate versions of the t-test and the χ^2-test fail to detect leaked information with confidence, while DL-LA still succeeds with only half of the available traces. As an unintended byproduct of our practical case studies, we provide the most detailed practical comparison between the Welch's t-test and Pearson's χ^2-test that has been reported in the literature so far.

The most outstanding advantage of our approach is clearly that the underlying network is free to combine as many points for the classification of the two groups as necessary. Thus, even in complex scenarios of purely multivariate or horizontal leakages, the traces can simply be fed as training data into the network without any trace-specific pre-processing or manual selection of points. Accordingly, neither a high expertise is demanded from the evaluator, nor is it required to obtain any prior information about the underlying implementation or the type of leakage that is expected. We believe that proving distinguishability of two sets of data by actually building a successful distinguisher is an elegant solution and more intuitive than the usual statistical arguments. We also compare our neural-network-based classifiers to multivariate Gaussian models which theoretically could capture distributed leakages as well. However, based on our experiments we conclude that such template-analysis methods can not provide the same level of flexibility as machine learning approaches. Compared to the common univariate distinction tests, DL-LA generally entails a lower risk of false positives as it provides a single confidence value to assess the distinguishability of the groups. Traditional point-wise methods would need to normalize their confidence values to the number of points in the traces to provide a meaningful confidence threshold. However, this inaccuracy is mostly disregarded in their respective methodologies. Even though, DL-LA provides only a single confidence value, the approach can still identify the points of interest in side-channel traces that contain leakage, by performing a Sensitivity Analysis (SA) on the trained network. Obviously, the average computation time to perform a DL-LA is significantly higher when compared to simple univariate tests (a comparison is provided in Section 4). However, as soon as more complex types of side-channel leakage need to be analyzed, the additional run time quickly pays off, since the effort that otherwise has to be spent in order to make traditional methods recognize those complex patterns (if possible) grows even bigger and contains several steps that are hard to automate.

---

[1] We explain in Section 3 that in the DL-LA procedure there is a distinction between the required number of traces for the detection (i.e., the training set) and the required number of traces to carry out the evaluation (i.e., the sum of the training set and the validation set).

1.2 Claims and Non-Claims

In order to avoid any potential confusion regarding our claims, or lack thereof, we explicitly list the most important statements below:

We *do not* claim that ...

- ✗ our chosen networks are optimal for leakage detection in general or for any of the considered case studies in particular. We are certain that there is room for improvement, especially when considering individual cases, as we intentionally optimized for robustness and simplicity instead of single case performance.
- ✗ our chosen networks necessarily lead to classifiers that outperform the t-test or the χ^2-test for any given side-channel traces.
- ✗ DL-LA is generally superior and should replace established leakage detection techniques.
- ✗ DL-LA generally causes none or fewer false negatives than the classical approaches.

We *do* claim that ...

- ✓ the chosen networks offer some basic universality and robustness. Their success in all nine case studies featuring three different implementation platforms (FPGA, ASIC, µC) is practical evidence for this claim.
- ✓ the chosen networks are able to learn first-order, higher-order, univariate, multivariate and horizontal leakages without requiring any trace-specific pre-processing or prior knowledge about the underlying implementation.
- ✓ DL-LA entails a much lower risk of false positives (if the same confidence threshold is chosen) since it provides one confidence per set of traces instead of one confidence per time sample in the trace set.

2 Background

In this section we introduce the necessary background with respect to the roots and state-of-the-art of leakage assessment, as well as deep learning and its applications to side-channel analysis.

2.1 Leakage Assessment

Ever since the introduction of side-channel attacks in 1999 [KJJ99] the standard approach for assessing the physical vulnerability of a device has been a more or less exhaustive verification of its resistance against known attacks while attempting to cover a broad range of intermediate values and hypothetical leakage models. This approach, however, became less feasible over the years due to the increasing amount of new attack methods and the higher complexity of potential leakage models due to the introduction of countermeasures against physical attacks. Another concern regarding this procedure is that it entails a significant risk of reporting physical security in favor of the DUT while in reality merely a certain attack vector was missed in the process (by mistake or because it was unknown at time of evaluation) that could indeed enable key recovery [SM15]. Hence, the need for a robust and reliable standard leakage assessment method independent of concrete attack scenarios, targeted intermediates and hypothetical leakage models grew consistently over the years. In an attempt to gather and evaluate promising candidates, the National

Institute of Standards and Technology (NIST) hosted a "Non-Invasive Attack Testing Workshop" in 2011. One of the most intriguing proposals at the workshop was the use of the non-specific Welch's t-test [GJJR11] for leakage detection. Leakage detection avoids any dependency on the choice of intermediates and leakage models by focusing on the detection of leakage only, without paying any attention to the possibility to exploit said leakage for key recovery. Simply put, the concept is based on supplying the device under test with different inputs, recording its leakage behavior and evaluating whether a difference can be observed. Thus, such a method is suitable for black box scenarios and allows certification of a device's physical security by third party evaluation labs without the need to test a multitude of different methods and parameter combinations. Seven years later, after some shortcomings of the moment-based nature of the t-test had been identified [Sta18], another popular statistical hypothesis test was proposed for leakage detection purposes, namely the Pearson's χ^2-test [MRSS18]. Both hypothesis tests, the t-test and the χ^2-test, are applied in the field of statistics in order to answer the question whether two sets of data are significantly different from each other. To be more precise, the evaluation of the tests examines the validity of the null hypothesis, which constitutes that both sets of data were drawn from the same population (i.e., they are indistinguishable) [SM15]. In side-channel analysis contexts, it is usually evaluated whether two groups of measurements can be distinguished with confidence. Traditionally, those two groups are acquired by supplying the DUT either with random (group Q_0) or a fixed input (group Q_1), selected by coin toss. Later, it has been demonstrated that the careful choice of two distinct fixed inputs (instead of maintaining one group for random inputs) usually leads to a lower data complexity for the distinction [DS16]. We provide the details on how to conduct the Welch's t-test and Pearson's χ^2-test below.

Welch's *t*-test. We denote two sets of data by Q_0 and Q_1, their cardinality by n_0 and n_1, their respective means by μ_0 and μ_1 and their standard deviations by s_0 and s_1. The t-statistics and the degrees of freedom v can then be computed using the following formulas.

$$t = \frac{\mu_0 - \mu_1}{\sqrt{\frac{s_0^2}{n_0} + \frac{s_1^2}{n_1}}} \qquad v = \frac{\left(\frac{s_0^2}{n_0} + \frac{s_1^2}{n_1}\right)^2}{\frac{\left(\frac{s_0^2}{n_0}\right)^2}{n_0-1} + \frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1-1}}$$

Afterwards, the confidence p to accept the null hypothesis can be estimated via the Student's t probability density function, where $\Gamma(.)$ denotes the gamma function [SM15, MRSS18].

$$p = 2\int_{|t|}^{\infty} f(t,v)dt \qquad f(t,v) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{\sqrt{\pi v}\,\Gamma\left(\frac{v}{2}\right)}\left(1 + \frac{t^2}{v}\right)^{-\frac{v+1}{2}}$$

In practice, for the sake of simplicity, it is common to only evaluate the t-statistics and to set the confidence threshold for distinguishability to $|t| > 4.5$. The statistical background of this threshold is that for $|t| > 4.5$ and $v > 1000$ the confidence p to accept the null hypothesis is smaller than 0.00001 which is equivalent to a 99.999 % confidence that the two sets were *not* drawn from the same population. Of course, when the degrees of freedom v are not explicitly evaluated, it can occur that the assumption $v > 1000$ does not hold. However, practice has shown that this procedure rarely produces false positive results in side-channel analysis contexts. Yet, calculating the actual confidence p is certainly preferable, scientifically correct and can still be efficiently implemented [MRSS18]. Since the Welch's t-test is designed to distinguish the means of two distributions, it can only be applied to first-order univariate analyses in its simplest form. Schneider *et al.* [SM15] extended the methodology to arbitrary orders and variates and provide the required formulas for incremental one-pass computation of all moments.

Pearson's χ^2-test. In order to mitigate some of the limitations and shortcomings of the moment-based nature of the Welch's t-test, in particular for higher-order analyses of masked implementations, Moradi *et al.* [MRSS18] suggested the Pearson's χ^2-test. In contrast to the t-test this hypothesis test analyzes the full distributions and can capture information that lies in multiple statistical moments. Thus, it prevents false negatives when moment-based analyses become suboptimal [MRSS18].

In a first step a contingency table F has to be constructed from the two sets Q_0 and Q_1 (basically two histograms). We denote the number of rows by r (= 2, when two sets are compared) and the number of columns by c (number of bins of the histograms). The χ^2-statistics x and the degrees of freedom v can then be computed using the following formulas.

$$\mathrm{x} = \sum_{i=0}^{r-1}\sum_{j=0}^{c-1} \frac{(F_{i,j} - E_{i,j})^2}{E_{i,j}} \qquad\qquad \mathrm{v} = (r-1)\cdot(c-1)$$

$E_{i,j}$ denotes the expected frequency for a given cell.

$$E_{i,j} = \frac{\left(\sum_{k=0}^{c-1} F_{i,k}\right)\cdot\left(\sum_{k=0}^{r-1} F_{k,j}\right)}{N}$$

Finally, the confidence p to accept the null hypothesis is estimated via the χ^2 probability density function, where $\Gamma(.)$ denotes the gamma function [MRSS18].

$$p = \int_{\mathrm{x}}^{\infty} f(x,\mathrm{v})dx \qquad\qquad f(x,\mathrm{v}) = \begin{cases} \frac{x^{\frac{\mathrm{v}}{2}-1}e^{-\frac{x}{2}}}{2^{\frac{\mathrm{v}}{2}}\Gamma(\frac{\mathrm{v}}{2})} & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

In contrast to the t-test this procedure can easily be extended to more than two sets of data ($r > 2$), which can be a valuable feature when used as distinguisher for key recovery attacks. Generally, it can be said that in cases where the χ^2-test provides a higher confidence to reject the null hypothesis than the t-test (on the same side-channel data), the analysis of the leakages requires some special attention. This is usually the case when masked implementations with low noise levels are analyzed [Sta18, Moo19] or when hardware-masking schemes like threshold implementations cause leakages in multiple moments due to physical defaults such as glitches [Sta18, MRSS18].

2.2 Deep Learning

We give a brief summary of the history and applications of deep learning and subsequently introduce definitions and explain the underlying principle.

History and Applications. Historically, the field of machine learning dealt with extracting meaningful information from data by applying relatively simple mathematical models, e.g., Bayes Classifiers, Support Vector Machines or Decision Trees to a sanitized version of the input data. This required manual and time-consuming *feature engineering* to predetermine which elements might be useful in a given set of raw data and how to best represent them, e.g., Canny edge detection as a first hard-coded step for image classification.

In contrast, deep learning methods are generally capable of learning from raw input data, thereby making the elaborate modeling process unnecessary. Since the breakthrough improvement of classification accuracy on the ImageNet data set in 2012 [KSH12], deep learning has been successfully applied to many diverse tasks such as speech recognition, drug discovery, natural language processing, visual art style transfer, image classification, autonomous driving and strategy games.

More recently, the side channel community discovered deep learning as a tool to perform profiled attacks [CDP17, HGM+11, MPP16, MDP19b] with competitive results compared to classical modeling techniques, e.g., based on a multivariate normal distribution. On the other hand, the run-time effectiveness of DL-based approaches over classical machine learning is sometimes questioned [PSK+18]. Apart from our present work, only few publications have investigated the use of deep learning for the non-profiled case, including [Tim19] and [PCBP21]. In the former article a method is introduced that exploits the correlation between a correct key guess and a steep learning rate to enable key recovery. The latter article introduces a novel framework based on unsupervised learning to improve horizontal attacks on (protected) implementations of public-key cryptosystems.

Principle and Definitions. In the following we limit ourselves to sequential neural networks (without recurrent elements) used for the purpose of classification. The aim of this description is to give brief definitions for the standard terms in deep learning, while the explanation of principles is intentionally very shallow.

A neural network is structured into multiple layers, each containing a matrix of learnable weights w that is linearly applied to its inputs x and a non-linear activation function applied to each coordinate of the result of this matrix multiplication. The output of this combined operation is taken as an input for the subsequent layer. Finally, the output layer of the neural network contains as many output coordinates as classes[2] (c) and uses softmax as an activation function

$$\text{softmax}(x_j) = \frac{e^{x_j}}{\sum_{i=1}^{c} e^{x_i}},$$

such that the sum over all outputs is always equal to one, thereby forming a probability distribution over the possible class labels.

Let us first assume the weights are initialized with some values before an evaluation of the network takes place by applying the function of the first layer to the input sample and subsequently propagating the computed values forward layer by layer until all layers have been evaluated. The prediction y' consists in the output coordinate of the final layer with the highest value.

In the beginning, the weights in a neural network are initialized with random values. To determine useful weights that achieve accurate prediction values, a *training phase* is necessary. First, the designer needs to define a metric to measure the distance between a prediction y' and the actual class label y. This metric is called a *loss function* which determines the *loss score*. To perform training, a data set with labeled inputs, i.e., a list of tuples (x, y), is separated into *batches* of a fixed size b. The neural network is evaluated simultaneously on all samples in a batch thereby producing loss scores. After each batch an optimization strategy based on *Backpropagation* is used to adjust all weights in the neural network dependent on the gradient of the loss function. Each iteration through the entire training set is called an *epoch*. To minimize the loss score, training over multiple epochs is performed in each of which the training data is randomly regrouped into new batches. For simplicity we assume that the training ends after a predetermined number of user-defined epochs.

To judge the quality of the classifier during and after training, the metrics *accuracy* and *validation accuracy* should be considered. While accuracy is related only to the training set, validation accuracy takes an entirely separate *validation set* into account, to ensure that the traits learned are actually generalizable opposed to rote learning of the specific training set (the latter phenomenon is called *overfitting*).

When choosing and training a deep learning model, the designer has to determine values for many so-called *hyper-parameters*[3], these include the depth of the network, the types and

---

[2] We limit ourselves to this variant called one-hot encoding.

[3] In distinction from the parameters, i.e., the concrete weights learned during training.

sizes of layers, their activation function, the loss function and the optimizer strategy. We provide the hyper-parameters we chose for both of our network architectures in Section 3.3.

3 DL-LA: Deep Learning Leakage Assessment

We introduce Deep Learning Leakage Assessment (DL-LA), a novel leakage assessment methodology based on deep learning. Our method is simple to apply and outperforms classical leakage detection approaches such as the Welch's t-test and the more recently proposed Pearson's χ^2-test in many cases due to its intrinsically multivariate nature. We specify two neural networks that showed excellent and robust detection performance in all our practical case studies. Further, we apply Sensitivity Analysis (SA), which has been used in other SCA contexts for leakage visualization, to leakage assessment and preempt several common pitfalls during adoption of DL-LA.

3.1 Core Idea of DL-LA

The aim of leakage assessment is to determine whether an attacker is able to extract information from side-channel measurements. The current state of the art for non-profiled adversary models is based on univariate statistical distinction tests (Welch's t-test, Pearson's χ^2-test) which are applied to two groups of side-channel measurements collected for two distinct fixed inputs processed by the target implementation (alternatively one group for random inputs and the other one for fixed).

DL-LA maintains the basic idea of distinguishing two groups of side-channel traces from each other (fixed-vs-fixed). Hence, from an evaluator's perspective the entire measurement setup and tool-chain can remain unchanged when adopting our methodology. We apply deep learning to the concept of leakage assessment by training a neural network to serve as a distinguisher between the two groups. This is done in a supervised-learning-based approach by applying labeled data from both groups to the network. The set of data applied to the network during this phase is then called the training set. Afterwards, the classification capabilities of the network are evaluated on a distinct validation set of labeled measurements without revealing the true labels to the network. The success rate of the classification on the validation set quantifies the amount of generalizable information that the neural network could extract from the *training* set during the *training* phase. In more detail, the network has learned generalizable features during the training when it can provide a better-than-random guess which of the two fixed inputs was processed to produce an individual validation trace. In case the classification of the whole validation set succeeds with a higher percentage than it could be achieved by a randomly guessing classifier with a non-negligible probability, it gives clear evidence for the fact that informative side-channel leakage is present. In this context we present a simple metric to determine an exact probability value p that quantifies the statistical confidence in the evidence.

3.1.1 Training vs. Validation Set

Please note that the number of required traces to detect the leakage is only related to the *training* set. The size of the validation set can be chosen completely independent and influences the result of the detection only if generalizable features (i.e., informative side-channel leakage) could be extracted from the *training* set. Otherwise the percentage of correct classifications will never be significantly different from 50%, no matter how large the validation set is. To be more precise, the information included in the validation set has no impact on the already trained classifier. It is merely an auxiliary data set required to test the quality of the classification ability.

Therefore, in DL-LA it has to be distinguished between the required number of traces for the detection (i.e., the training set) and the required number of traces to carry out the evaluation (i.e., the sum of the training set and the validation set). Only the detection (i.e., the training) traces represent the number of measurements available to an attacker. In other words, DL-LA evaluates the likelihood that an attacker is able to extract informative side-channel leakage from the finite set of training traces. While the evaluator clearly requires both sets to perform the evaluation, reporting the combined cardinality of both sets as the number of traces for a successful detection would be misguided and could lead to an incorrect impression about the security of the target. As an example, we assume an evaluator wants to know with high confidence whether a set of 1 000 traces contains a sufficient amount of information to detect a difference between the two underlying groups of measurements. The typical procedure would be to train a neural network as a classifier on those 1 000 traces over as many epochs as desired (while being mindful of overfitting) and then validating the classifier on an arbitrarily-sized validation set. Even if the validation set must be 100 000 000 traces large to overcome the confidence threshold desired, it still means that only 1 000 traces were required to find and learn generalizable features in the traces that allow better-than-random classification of new and unseen measurements into the two groups. Simply speaking, this means that 1 000 traces already leak confidently detectable information. However, 100 001 000 traces would be required for the evaluation. Of course this is an extreme and unlikely corner case. Typically, much more reasonable trade-offs between training set and validation set size can be achieved, which is also demonstrated in the practical case studies we present in Section 4. Especially when examining well-protected implementations which require millions of traces for a meaningful analysis the size of the validation set is typically not the prohibitive element and is often significantly smaller than the training set. Yet, it is important to make the distinction between detection and evaluation traces, since minimizing the combined set, namely training + validation, is not trivial and not a focus of this work. In the example detailed above the evaluator could simply increase the size of training set in hopes of improving the trained classifier and require a smaller validation set (and likely a smaller combined set) for a confident detection. However, this would not properly answer the question whether 1 000 traces allow extraction of information. Therefore, we do not explore strategies to find the minimum combined cardinality of the two sets although this might be an interesting topic for future research in the area. We provide further discussion on the partition strategy into training set and validation set to decouple the number of traces available to the attacker from the statistical confidence the evaluator wants to obtain in Section 5.

3.1.2 Fixed-vs-Fixed or Fixed-vs-Random

Traditionally, leakage detection methods have relied on distinguishing one group of measurements acquired when supplying the device under test with random inputs from another group recorded when the device received a fixed input over and over again (although both groups should be recorded in a randomly interleaved sequence [SM15]). Yet it was pointed out at EUROCRYPT 2016 [DS16] that a partitioning based on two different fixed inputs normally leads to a lower data complexity (i.e., fewer traces required for a successful detection). The arguments given by the authors are essentially the same that led us to suggest fixed-vs-fixed as the default partitioning strategy for DL-LA. In fact, the whole DL-LA concept is applicable to a fixed-vs-random grouping as well. However, in that case a larger data complexity has to be expected. Intuitively, this can be understood best when picturing the distributions for both groups of measurements at one individual sample point in the traces. In case of one group for fixed and one for random inputs, the two distributions will always overlap if their cardinality is sufficiently large, since the fixed input is also contained in the set of all inputs from which the random inputs are selected. A larger difference between the two distributions is possible for two distinct fixed inputs. In some cases the distributions may even be disjoint and allow perfect classification into the two groups. While this will not occur for measurements of securely masked implementations, similar arguments can be made for higher-order statistical moments. Generally, the maximum difference between two distributions at one sample point recorded for fixed inputs will always be larger than the difference between one fixed and one random group for a sufficiently large number of traces. When measuring the execution of a cryptographic primitive over an extended period of time for each trace, sample points with large differences between the two fixed classes will inevitably occur [DS16]. This is a conceptual difference to analysis techniques that do not record a trace over time, but rather take a single snapshot of the current state, such as static power SCA attacks [Moo19]. In those cases the two fixed inputs need to be selected with greater care in order to not accidentally choose two fixed classes which lead to very similar leakage distributions. In our experimental analysis, however, the leakage traces have a significant length in terms of collected sample points ($\geq 2\,000$) and covered clock cycles (> 20). Thus, following the arguments of [DS16], it is unlikely that such traces recorded for two fixed classes show a smaller maximum difference over the full length of sample points and clock cycles than traces recorded for a fixed class and a random class. Hence, we are confident, and our experimental attempts have confirmed this, that a fixed-vs-fixed partitioning strategy is preferable for the DL-LA (and t- and χ^2-test) methodology over a fixed and a random class.

3.2 Overall Methodology

We assume that the recorded traces have already been separated into a set of N training traces and a set of M validation traces, the latter of which should have an equal number of elements from both groups to maximize the statistical confidence value that can be obtained during the evaluation[4]. Initially, we determine the point-wise mean μ and standard deviation σ of the whole trace set and standardize both the training and the validation set by calculating

$$X_i^j := (X_i^j - \mu_i)/\sigma_i,$$

with j denoting the trace and i the time sample within the trace. This very lightweight and universal pre-processing step is necessary to reach a homogeneous range between all input points and weights thus enabling efficient training.

Afterwards, the evaluator has to pick a confidence level, i.e., an upper bound on the chance that a false positive occurs. We assume the common threshold in SCA evaluations of $p_{\text{th}} = 10^{-5}$. Now, let v be the validation accuracy obtained by the neural network, then the total number of correct classifications is computed as $s_M = v \cdot M$. Considering the null hypothesis $\mathcal{H}_0$ where the neural network did not learn anything and classifies randomly (coin flip model), this corresponds to modeling the total number of correct guesses as a random variable following a binomial distribution

$$\mathcal{H}_0 : X \sim Binom(M, 0.5).$$

The probability that at least s_M correct classifications occur in a purely random classifier is given by: $P(X \geq s_M)$. This probability is easily computed as

$$P(X \geq s_M) = \sum_{k=s_M}^{M} \binom{M}{k} 0.5^k 0.5^{M-k} = 0.5^M \sum_{k=s_M}^{M} \binom{M}{k}$$

Now, we say that the implementation leaks information about the processed data if

$$P(X \geq s_M) \leq p_{\text{th}}.$$

In this case the exact location of leakage can be determined subsequently by Sensitivity Analysis (cf. Section 3.4).

---

[4] We provide a discussion on the size of both sets in Section 5.

Table 1: Minimum validation set sizes calculated for different validation accuracy values v obtained in Step 1.

v	M'	$v \cdot M'$
75.00 %	76	57
60.00 %	470	282
56.00 %	1 300	728
51.00 %	45 600	23256
50.50 %	182 200	92011
50.10 %	4 549 000	2279049
50.05 %	18 194 000	9106097

3.2.1 Minimum Size of the Validation Set

As previously explained, there is always a trade-off between the size of the training set and the size of the validation set while the minimization of their combined cardinality is not trivial. Assuming the size of training set has been set to a fixed value N by the evaluator, for instance because it determines the lifetime of the key or the whole device, then choosing the minimum size of the validation set for a confident result can be approached by the following iterative procedure.

- *Step 1*: Choose $M = 1{,}000$ as the size of the validation set (or any other number of traces that can be recorded in a short period of time)
- *Step 2*: Perform DL-LA using N training and M validation traces, observe the validation accuracy v
- *Step 3*: Find smallest integer M' such that $p_{\text{th}} \geq 0.5^{M'} \sum_{k=v \cdot M'}^{M'} \binom{M'}{k}$
- *Step 4*: Perform DL-LA using N training and M' validation traces, observe the validation accuracy v'
- *Step 5*: If $p_{\text{th}} \geq 0.5^{M'} \sum_{k=v' \cdot M'}^{M'} \binom{M'}{k}$, the procedure terminates, otherwise set $v = v'$ and repeat from the Step 3

This approach can be useful to approximate the total number of traces the evaluator requires in addition to the training set in order to achieve a confident result, but only in case detectable leakage is present. It relies on the assumption that the validation accuracy which the trained classifier achieves on a comparably small validation set can approximately be maintained on a larger set. The smaller the initial value of M is, the likelier it is that this assumption can be incorrect. In such a case, multiple iterations may be required. We have listed exemplary results of the procedure in Table 1. Please note, in case the trace set does not contain enough information for distinction between the two groups or the two groups indeed belong to the same population (i.e., the null hypothesis is true), the procedure will never terminate and M' will approach infinity.

In general, minimizing the validation set will yield results where the confidence threshold is just overcome. Often it can be useful to increase the validation set beyond M' in order to achieve higher confidence values. We would like to insist that the number of validation traces may often be the bottleneck for reducing the number of evaluation traces (training + validation) when analyzing unprotected implementations or generally traces that show significant amounts of leakage. However, when evaluating SCA-protected implementations, it is not uncommon, in our experience, that tens or hundreds of millions of traces are required for a meaningful analysis. In such cases, the validation set is typically not the prohibitive element. In our experience, validation sets larger than 5 or 10 million traces should not be necessary for any regular analysis, while larger training sets will often be required. This is also showcased in some of our case studies in Section 4.

3.3 Proposed Network Structures

As already discussed in Section 1, our goal is to select and propose networks that perform robustly on many different sets of side-channel data instead of maximizing the performance towards one particular data set. In other words, we try to keep the network architectures generally applicable and as free of any assumptions about the leakage to be analyzed or the underlying implementation as possible. We have taken multiple approaches in order to find such networks. First of all we have collected side-channel data containing different types of leakage, such as first-order, higher-order, univariate and multivariate leakages, from different kinds of devices, FPGA, ASIC, µC, while simulating different levels of measurement quality, such as high signal-to-noise ratio, low signal-to-noise-ratio, aligned and misaligned leakage traces. Once this collection had been assembled we essentially followed a trial-and-error based approach in order to find the most suitable number of layers and number of neurons per layer to built a simple multi-layer perceptron (MLP) providing the best average classification performance across our data sets. The resulting network, which is described below, is even fairly robust to small changes to its hyper-parameters.
In addition to the MLP we also suggest a simple convolutional neural network (CNN). In order to find suitable hyper-parameters for this CNN we performed a hyper-parameter search with Talos [mea20] on our data sets corresponding to all case studies. As a result we suggest a network including a set of 8 different hyper-parameter combinations which are evaluated against each other in Section 4. In summary, the two network architectures proposed below have been selected because they proved to deliver a respectable level of universality across a number of experimental data sets. Of course, we do neither claim that the selected networks are the optimal solution for such purposes, nor that they necessarily provide appropriate performance on any given set of side-channel data. Yet, we are confident that they represent a good starting point for an investigation.
We have built, tested and evaluated both of the network architectures described below in the Python library Keras (keras-gpu version 2.4.3) using TensorFlow (tensorflow-gpu 2.1.0) as the backend.

3.3.1 Multi-Layer Perceptron (MLP)

The MLP network consists of four fully-connected layers (*Dense*) of 120, 90, 50 and 2 output neurons. The input layer and each of the inner layers use a *Rectified Linear Unit* (*ReLU*) as an activation function, while the final layer uses *softmax.* The four Dense layers are each separated by a *BatchNormalization* layer. In summary, the model can be defined in Python as:

```
model = Sequential([
    Dense(120, activation = ’relu’, input_shape= (tracelength,) ),
    BatchNormalization(),
    Dense(90, activation = ’relu’),
    BatchNormalization(),
    Dense(50, activation = ’relu’),
    BatchNormalization(),
    Dense(2, activation = ’softmax’)])
```

Further, we used the *mean squared error* as a loss function and *adam* as an optimizer with the default parameters provided by Keras[5]. We chose the batch size as 2 000 samples for

---

[5] $lr = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, decay = 0.0$

traces of length 5 000 points and 20 000 samples for traces of length 500 points and 1 00 for traces of length 200 000 points.

Justification. We chose $ReLU$ defined as

$$relu(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

as an activation function over other common possibilities, e.g., tanh or sigmoid, because of better results regarding validation accuracy in initial tests as well as for better computational performance when operating on large data sets (which is highly relevant for the evaluation of protected implementations). We chose the softmax activation function of the final layer to create a probability distribution over both classes as explained in Section 2.2. The purpose of each *BatchNormalization*-layer is to decouple the learning process of all Dense layers from each other and additionally provide a means of regularization to prevent overfitting [IS15].
We confirmed the suitability for univariate and multivariate leakage located in different statistical orders and for traces as short as one point and as long as 200 000 points which may be encountered during a typical leakage evaluation of symmetric cryptographic primitives and provide extensive depth on the performance of our leakage assessment approach in different case studies in Section 4.

3.3.2 Convolutional Neural Network (CNN)

In addition to the comparison between the MLP-based DL-LA and the classical detection approaches in our case studies in Section 4, we performed a hyper-parameter search for CNNs with Talos [mea20] on the data sets corresponding to all case studies. We utilized the following convolutional network with one convolutional, one pooling and a final dense layer

```
model = Sequential([
    Reshape((tracelength,1), input_shape = (tracelength,)),
    Conv1D(filters=filter, kernel_size=kernel_mult*peakdist, \
           strides=peakdist//strides, input_shape=(tracelength,1), \
           activation='relu'),
    MaxPooling1D(pool_size=pool),
    Flatten(),
    Dense(2, activation='sigmoid')])
```

to provide an extremely simple high-level network architecture that is not specialized for any particular power traces and then performed a search through the following eight hyper-parameter combinations:

```
    filter: [12]
    kernel_mult: [2, 3]
    strides: [2, 3]
    pool: [2, 4]
```

Note that the kernel size and stride distance is given relative to the distance of peaks, i.e., the length of a clock cycle, in the power trace. We chose the multiplicative factor larger than one to assure that the convolution combines information from multiple clock cycles. While we limited ourselves to the given eight hyper-parameter combinations, many choices greater than one for filter, kernel multiplier, strides and pool are theoretically sound and should lead to comparable results (cf. Section 4.1). In contrast to the MLP network we used the *binary_crossentropy* as a loss function and chose *sigmoid* as the activation function of the final layer.

3.4 Extracting Temporal Information

If leakage is detected, the hardware designer or evaluator is usually interested in exactly pinpointing the leakage locations to report or alleviate the shortcoming, e.g., masking flaws. By applying *Sensitivity Analysis (SA)* based on input activation gradients [SZ13, Tim19, MDP19a, PEC19], we can exactly locate all points of interest by quantifying how much they contributed to the leakage function learned by the neural network. In short, SA determines the partial derivates of one output coordinate of the neural network with respect to the network inputs, thereby characterizing the effect of a slight change in each individual input on the classification outcome.

We perform SA on the final network after training has completed by averaging the gradients of one output coordinate (with respect to the network inputs) weighted with the network inputs for all samples in the training set and subsequently take the absolute value. More precisely, let x_i denote the i-th input of our network, y_0 the first output coordinate of the network and X_i^j the value of the i-th input for trace j in the training set. Then the sensitivity can be determined as:

$$s_i = \left| \sum_j \frac{\partial y_0}{\partial x_i} \cdot X_i^j \right|.$$

While the actual value of this expression has to be determined via the chain-rule over all network layers, this process is fully-automated by TensorFlow such that the remaining effort for the evaluator is a single function call. Instead of considering the network inputs, a sensitivity analysis may also be performed based on the first layer weights [Tim19].

3.5 Common Pitfalls

We discuss the most important differences between the classical detection approaches and DL-LA and aim to preempt common pitfalls an evaluator might encounter with our leakage assessment:

Group Imbalance. While the classical TVLA based on the Welch's t-test as well as the χ^2-test can handle groups imbalanced in mean, variance and size, we want to stress that an equalization of group sizes in the validation set is extremely important for DL-LA. If the groups are imbalanced, the test statistic no longer follows the distribution $X \sim Binom(M, 0.5)$. Instead, always assigning the label of the more common group leads to a classifier which outperforms random guessing, without actually being able to distinguish the groups based on their traces. This discrepancy between actual and theoretical distribution of the test statistic – given a sufficiently large validation set – will lead to false positives. While the problem of severe group imbalance has previously been addressed by sampling techniques [PHJ$^+$19], we instead strongly advise pruning both groups in the validation set to an exact ratio of 50/50 to achieve the highest possible confidence.

Probability Adaption. An obvious idea to counteract the issue just addressed is the adaption of the success probability in the Binomial distribution. Assume a slight (or even significant) imbalance ϵ in group sizes

$$\frac{|G_0|}{|G_0| + |G_1|} = 0.5 + \epsilon.$$

The evaluator could simply adapt the distribution of the test statistic to

$$X \sim Binom(M, 0.5 + \epsilon).$$

While this might even show satisfactory results in the case of low noise and unprotected or severely flawed implementations, which in turn lead to a high validation accuracy, we caution against any alteration of the distribution. In all practically relevant cases (protected implementation, moderate noise) a change of the success probability severely lowers the confidence of the statistical test. More specifically, consider a validation set of size 500 000 traces over which a validation accuracy of $v = 0.506$ has been achieved. In case of a balanced validation set this event is highly statistically significant (10^{-17}). However, if the validation set contains a small bias of $\epsilon = 0.004$ no significance can be concluded as the remaining likelihood for this event is only 10^{-3}. It is obvious that the adapted test looses its statistical power in all interesting cases; hence, false negatives might occur. Therefore, we want to reinforce the previous point to prune the validation set to an exact 50/50 ratio.

Overfitting. We caution against using an overly complicated neural network as it might lead to overfitting, which is defined by a continuous rise of the training accuracy over the number of epochs while the validation accuracy begins to fall. The underlying cause of this effect is the memorization of the training set as opposed to learning generalizable features of the entire set. Hence, it can be prevented by using a network with a simple structure which does not contain excessively many weights and optionally includes Normalization, Regularization or Dropout layers (cf. Section 3.3).

4 Experimental Results

In the following we provide an experimental verification of the suitability of DL-LA as a black box leakage assessment strategy. We strive for a realistic benchmark of our approach with a clear real-world impact. For this reason we chose power measurements of multiple hardware and software implementations of the PRESENT-80 ultra lightweight block cipher [BKL+07] as the common target in our case studies. PRESENT has been developed for ubiquitous and resource constrained computing environments, which exactly constitutes the type of application that commonly requires side-channel security as a design goal and may be certified by third-party evaluations labs. In total, we target nine different implementations of the cipher on three different platforms (FPGA, ASIC, µC). As a first step we evaluate the MLP (see Section 3.3) on all nine case studies for one concrete choice of hyper-parameters. Those results are compared to the previously introduced state-of-the-art leakage detection methods Welch's t-test and Pearson's χ^2-test. Afterwards, in order to demonstrate that robust detection on trace sets from 3 different platforms is not limited to this particular MLP, we evaluate the CNN (see Section 3.3) for a whole spectrum of hyper-parameters. Among the nine different implementations that are tested, multiple are protected by masking and hiding techniques. The masked variants feature provable first-order security. Some of them even provide security at any order against univariate-only attacks. The results show that DL-LA with our choice of network architectures is able to confidently detect leakage in smaller trace sets or with a higher confidence using the same amount of traces compared to the conventional methods.

4.1 Measurement Setup

For the first 7 case studies, we have implemented the different instances of the PRESENT block cipher on a SAKURA-G board [sak] which has specifically been designed for SCA evaluations. The board features two Spartan-6 FPGAs, one as a target and the other as a control interface. Case studies 8 and 9 feature protected versions of the PRESENT cipher implemented on a 40 nm ASIC prototype and an ARM Cortex-M0 microcontroller respectively. In all cases we have measured the voltage drop over a 1 Ω shunt resistor in

Table 2: Measurement details for the nine different case studies.

	Case Study 1	Case Study 2	Case Study 3	Case Study 4	Case Study 5
Platform	FPGA	FPGA	FPGA	FPGA	FPGA
Alignment	aligned	misaligned	heavily misal.	aligned	misaligned
Sampling Rate	1 GS/s	1 GS/s	1 GS/s	1 GS/s	1 GS/s
Frequency	6 MHz	6 MHz	$\leq$ 24 MHz	6 MHz	6 MHz
No. of Traces	1 000	1 000	5 000	10 000 000	10 000 000
No. of Points	5 000	5 000	5 000	5 000	5 000

	Case Study 6	Case Study 7	Case Study 8	Case Study 9
Platform	FPGA	FPGA	40 nm ASIC	Cortex-M0 µC
Alignment	aligned	misaligned	aligned	aligned
Sampling Rate	100 MS/s	100 MS/s	2 GS/s	500 MS/s
Frequency	6 MHz	6 MHz	12 MHz	8 MHz
No. of Traces	50 000 000	50 000 000	50 000 000	100 000
No. of Points	5 000	5 000	2 000	200 000

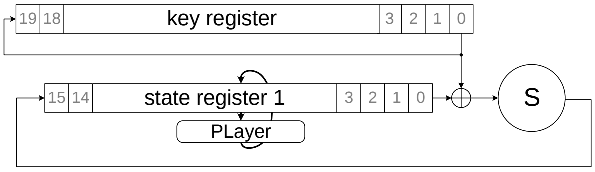


Figure 1: Unprotected serialized PRESENT architecture with a 4-bit data path.

the V_{dd} path of the target with a digital sampling oscilloscope. On the SAKURA-G board the measured signal is amplified through a built-in AC amplifier. The measurement details for each of the nine different case studies including sampling rate, operating frequency, number of traces and number of time samples per trace are listed in Table 2. For all case studies we measured side-channel traces in a fixed-vs-fixed manner for two arbitrarily selected fixed inputs. We have taken care to follow all rules that have to be considered to avoid false positives in leakage assessment [SM15], e.g., the measurements of the two groups are randomly interleaved and in the masked cases the communication between the control unit and the target is performed in a shared manner (in our case the same holds for the communication with the measurement PC).

Case Study 1: Unprotected PRESENT (FPGA), aligned Traces

In this first case study we target an unprotected serialized implementation of the PRESENT block cipher. The architecture can be seen in Figure 1 and is similar to profile 1 introduced in [PMK+11]. As a first step we evaluate the confidence to distinguish the two groups of measurements (fixed-vs-fixed) by conventional methods. The results of the first-order t-test and the χ^2-test can be seen in Figure 2. In both cases we plot the confidence values p instead of relying on the common (and less precise) approach of defining a threshold for the intermediate statistics (e.g., $|t| > 4.5$). The t-test succeeds in providing a confidence higher than 99.999 % for the distinguishability of the two groups after about 20 traces since it shows a probability below 10^{-5} to accept the null hypothesis. The χ^2-test requires approximately 90 traces to overcome the desired confidence threshold. In conclusion, none of the two methods faces any problems to distinguish the leakage distributions with a high confidence when 1 000 traces are considered.

When applying DL-LA to the same traces, the results in Figure 3 are achieved. We have to state here that a plot as depicted in Figure 3(b) is rather unnatural to obtain using DL-LA.

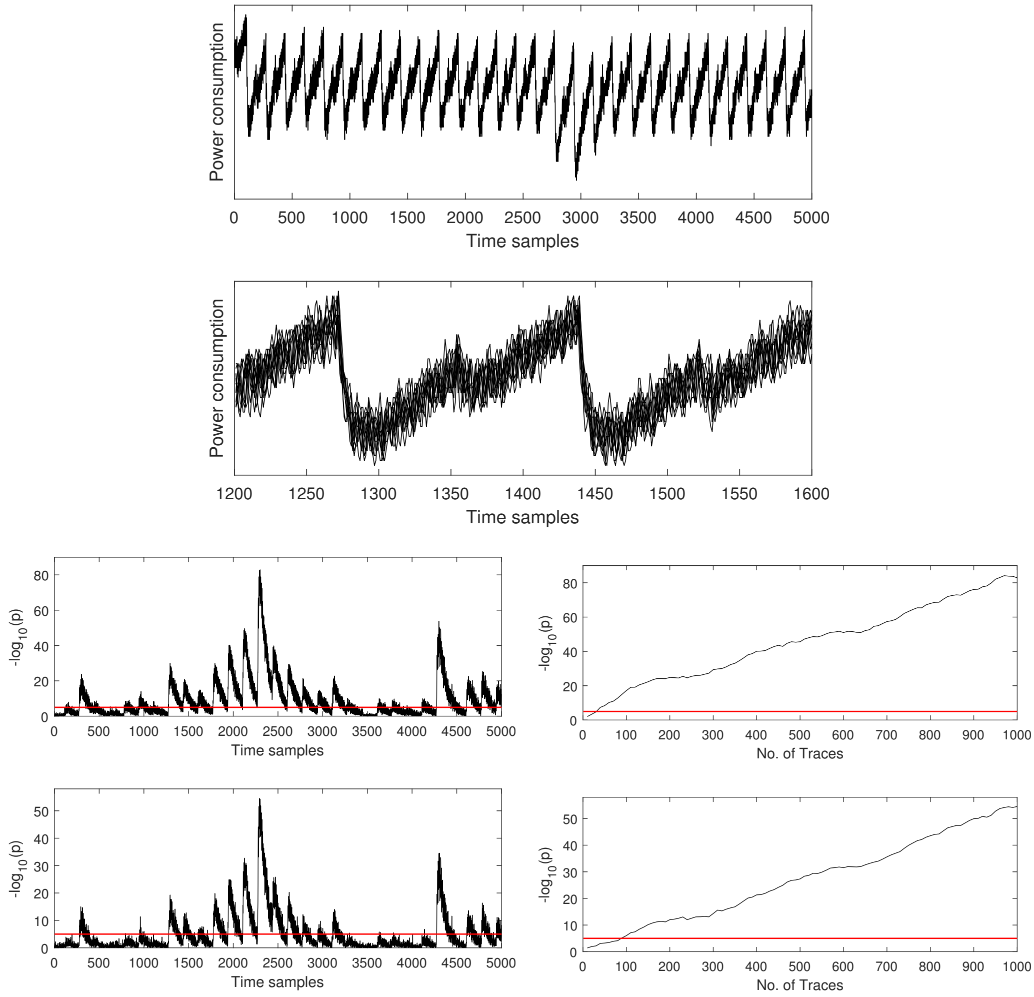


Figure 2: Univariate leakage assessment using 1 000 traces (step size 10) of an unprotected serialized PRESENT-80 implementation. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) t-test results, 4) χ^2-test results.

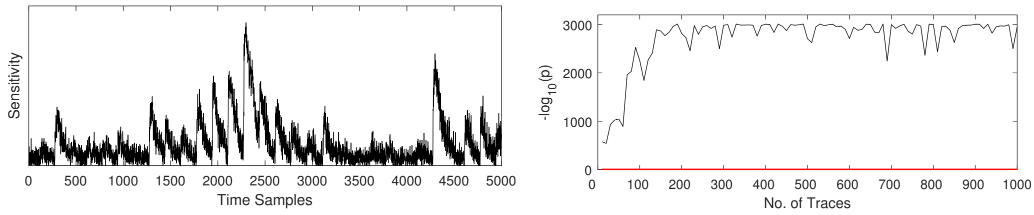


Figure 3: Sensitivity Analysis and DL-LA using 1 000 traces (step size 10) of an unprotected serialized PRESENT-80 implementation. For each p value 30 epochs and a validation set of 10 000 traces are considered.

Normally, training and validating the network results in a confidence value after each epoch. Thus, it would be more natural to train the network on a training set of fixed size and to show the p values over the number of epochs to determine how many are required to overcome the threshold. However, in order to offer the best possible comparison between the leakage assessment approaches we repeated this process 100 times for a fixed number of epochs (30) and a training set that increases by 10 traces per step and plotted the maximum confidence over the number of traces. The result shows that a network which is trained on only 10 traces is already capable of providing an extremely high confidence that the two groups are distinguishable (since large $-\log_{10}(p)$ values give confidence to reject the null hypothesis). By increasing the size of the training set the confidence is boosted significantly until the p values stagnate in a corridor between 10^{-2300} to 10^{-3011}. Please note that, as the validation set has a size of 10 000 traces, the maximum achievable p value is $0.5^{10\,000} = 10^{-3011}$. Thus, the stagnation in the corridor is simply caused by the fact that (almost) all of the traces in the validation set were classified correctly. By using a larger validation set the $-\log_{10}(p)$ values would rise even beyond 3011. We also perform a Sensitivity Analysis on the network to determine the points of interest and obtain a spatial resolution comparable to the univariate tests (cf. Figure 3(a)). The absolute values of the SA are not meaningful and cannot be compared to any threshold. Thus, they are omitted here. In summary, DL-LA outperforms the classical detection approaches in terms of required number of traces and absolute confidence provided. Of course, for the evaluation of DL-LA as performed in this case study, a validation set is required on top of the training set. However, please note that we only chose a validation set of 10 000 traces here in order to show the extremely high magnitude of achievable confidence values, even when considering very small training sets[6]. In fact, the indication of distinguishability relates only to the training set, and, in case the network learned generalizable features from it, the confidence can be arbitrarily boosted by increasing the validation set. If no generalizable features were learned (e.g., because no leakage is present) the percentage of correct classifications will not be different from 0.5 by a statistically significant magnitude. The advantages of decoupling the confidence from the number of traces (in the *training set*) are discussed in Section 5. In Figure 28 of Appendix A, we additionally provide DL-LA results for the first three case studies where the size of the union of the training and the validation set does not exceed the number of traces considered by the t- and the χ^2-test. Even in that case DL-LA outperforms the classical approaches.

Case Study 2: Unprotected PRESENT (FPGA), misaligned Traces

This case study is an exact replication of the previous one apart from the fact that we artificially created a misalignment of the traces, as apparent in Figure 4(b). This misalignment was achieved by forcing the oscilloscope to trigger the acquisition of the power traces close to the peak of the rising edge of the trigger signal (in our case at 2.48 V while the peak is at 2.5 V) as opposed to the more stable part in the middle of the edge. Thus, due to the electronic noise, the acquisition is in some cases triggered earlier than in others and the traces are not perfectly aligned anymore. Figure 4 shows that the t- and χ^2-test results do not seem to significantly suffer from this misalignment when considering the absolute magnitude of the $-\log_{10}(p)$ values. However, the number of traces to overcome the threshold is increased in comparison to the previous case study in both tests. DL-LA also performs similar as before, as apparent from Figure 5 and outscores the classical detection approaches in required traces and provided confidence. It seems that the slight misalignment of the traces does not significantly affect the detection

---

[6] In contrast, the minimum size of the validation set in order to be able to overcome the detection threshold is 17, as $-\log_{10}(0.5^{17}) > 5$. However, this assumes a 100% correct classification by the network, otherwise a larger set needs to be considered.

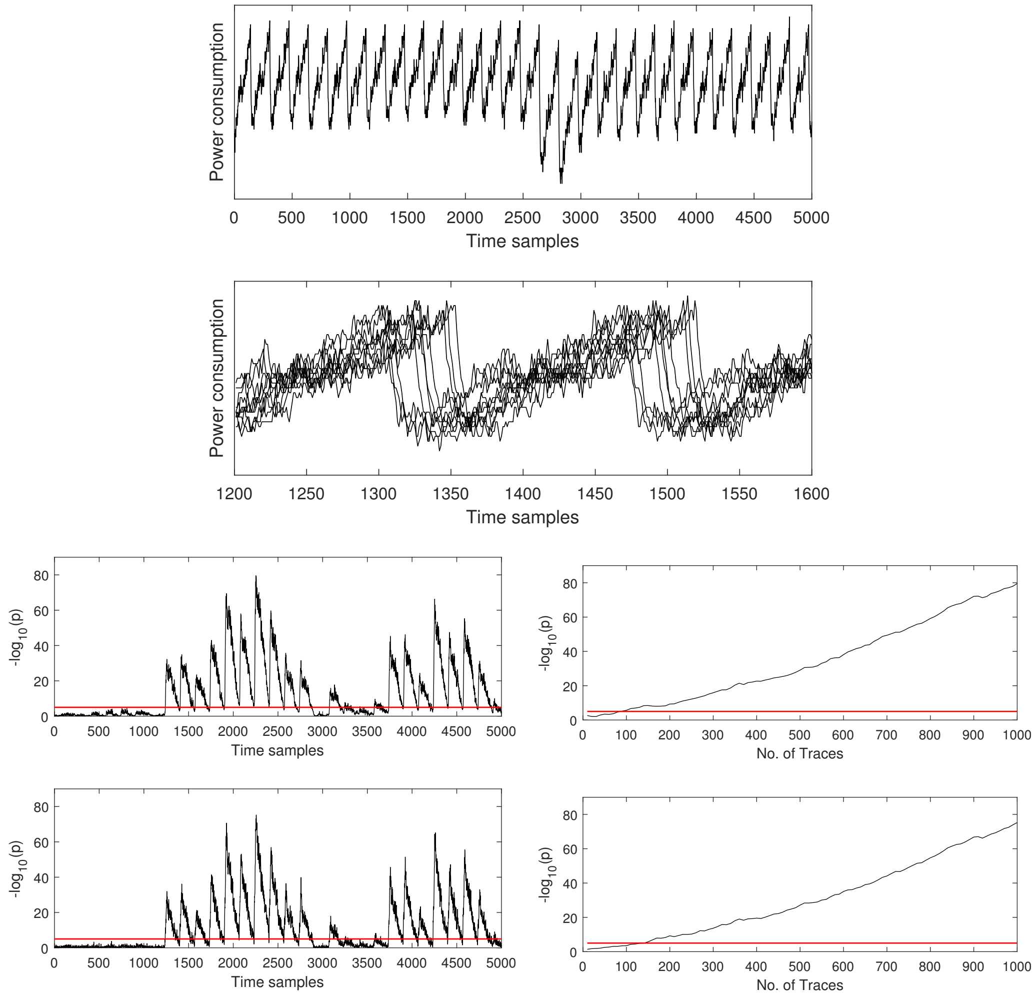


Figure 4: Univariate leakage assessment using 1 000 misaligned traces (step size 10) of an unprotected serialized PRESENT-80 implementation. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) *t*-test results, 4) χ^2-test results.

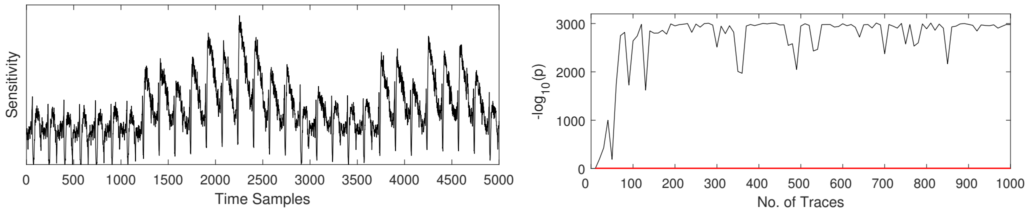


Figure 5: Sensitivity Analysis and DL-LA using 1 000 misaligned traces (step size 10) of an unprotected serialized PRESENT implementation. For each p value 30 epochs and a validation set of 10 000 traces are considered.

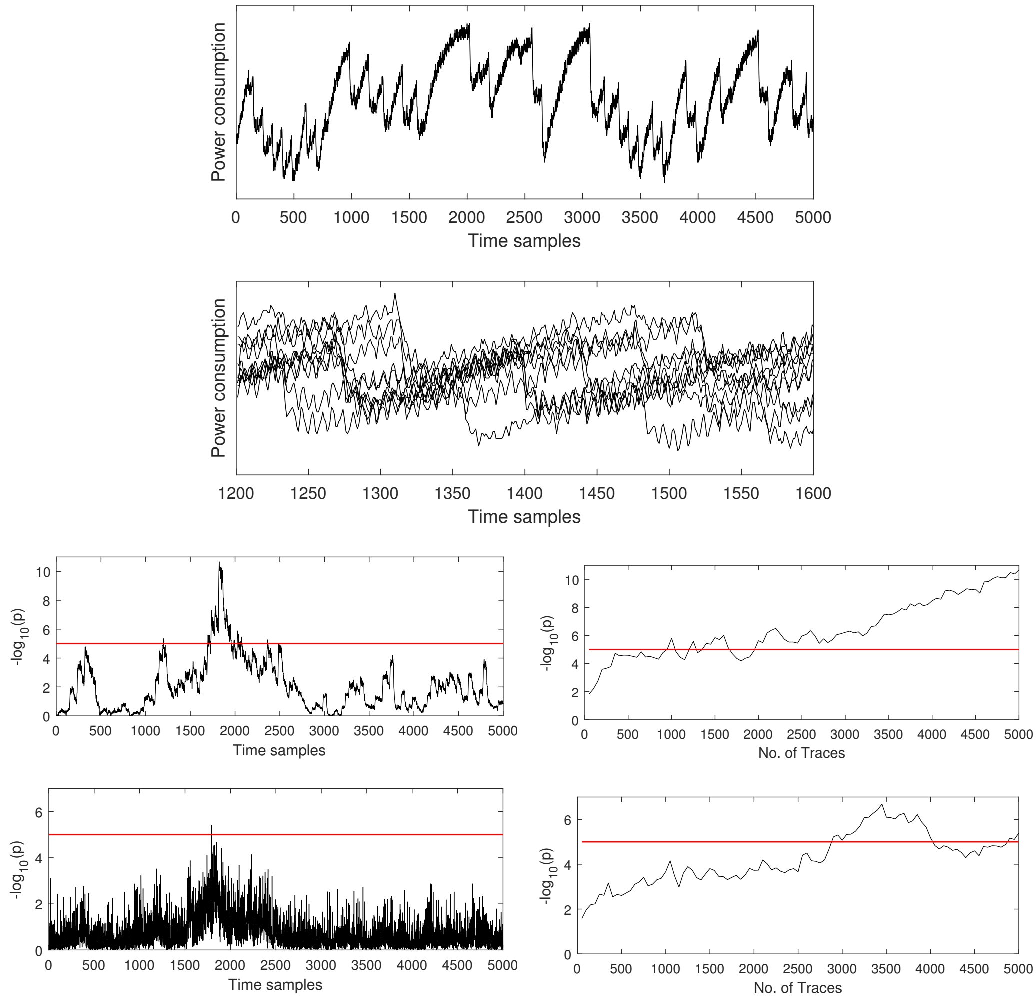


Figure 6: Univariate leakage assessment using 5 000 traces (step size 50) of a serialized PRESENT-80 implementation with clock randomization. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) t-test results, 4) χ^2-test results.

capabilities of any of the leakage assessment techniques when unprotected implementations are considered and the number of available traces is not chosen to be extremely small.

Case Study 3: (Unprotected) PRESENT (FPGA), randomized Clock

Since the artificial delay in the previous case study only slightly increased the data complexity of a leakage detection we now try to test a countermeasure that leads to much more heavily misaligned and noisy traces. In particular, we randomize the clock that drives the targeted PRESENT implementation. This is done by clocking the cipher with the output of a 64-bit LFSR. Hence, in each encryption (and therefore also in the power traces) the same intermediate computations are executed at different times, since the rising edges of the LFSR output occur in a random sequence. The input frequency of the LFSR was set to 24 MHz so that the number of rising edges in a certain frame of time is on average similar to being clocked by a stable 6 MHz clock. In this case the t- and χ^2-test struggle more significantly to detect leakage than in the previous experiments, as apparent in Figure 6. While the t-test requires about 2 000 traces for a detectable breach

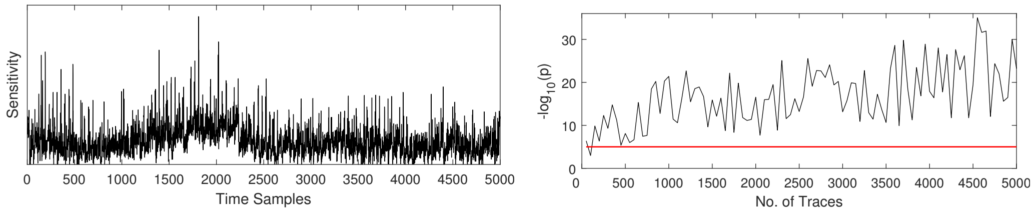


Figure 7: Sensitivity Analysis and DL-LA using 5 000 misaligned traces (step size 50) of an unprotected serialized PRESENT implementation with clock randomization. For each p value 30 epochs and a validation set of 10 000 traces are considered.

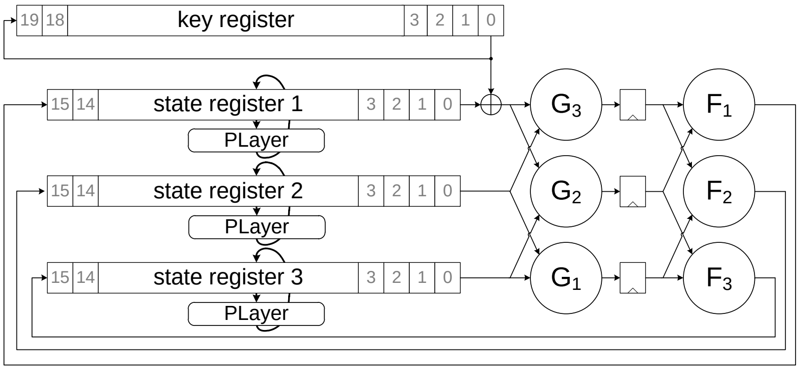


Figure 8: Serialized PRESENT threshold implementation architecture with 3 shares and a decomposed Sbox.

of side-channel security, the χ^2-test barely overcomes the threshold at all. DL-LA on the other hand is able to confidently state distinguishability after about 150 traces (cf. Figure 7). Although all three approaches suffer significantly from the misalignment and the added noise, DL-LA is still able to perform detection on a much smaller amount of traces. Please note that, if desired by the evaluator, the confidence can be made arbitrarily larger by increasing the size of the validation set.

Case Study 4: PRESENT TI (FPGA), aligned Traces

In this case study we target a serialized threshold implementation (TI) [NRR06] of the PRESENT block cipher. The architecture can be seen in Figure 8 and is similar to profile 2 introduced in [PMK$^+$11]. The PRESENT Sbox is decomposed into two quadratic functions F and G. Both of those decompositions are split into three component functions each according to the concepts of *correctness*, *non-completeness* and *uniformity* [NRR06]. As apparent from Figure 8 the three shares in the computation of the decomposed Sbox are evaluated in parallel. Thus, no first-order, but univariate higher-order (especially second- and third-order) leakage is expected. We evaluate this assumption in Figure 9. As expected the first-order t-test does not indicate detectable leakage, but the second- and third-order tests do. Interestingly, we can confirm the statements made by the authors of the χ^2-test proposal [MRSS18] regarding the shortcomings of the moment-based nature of the t-test. Unlike the situation in the previous case studies, the χ^2-test outperforms the t-test here. While the second-order and the third-order t-test require 3 000 000 and 1 100 000 traces for the detection respectively, the χ^2-test succeeds after only 600 000 traces and results in a much higher confidence over all.

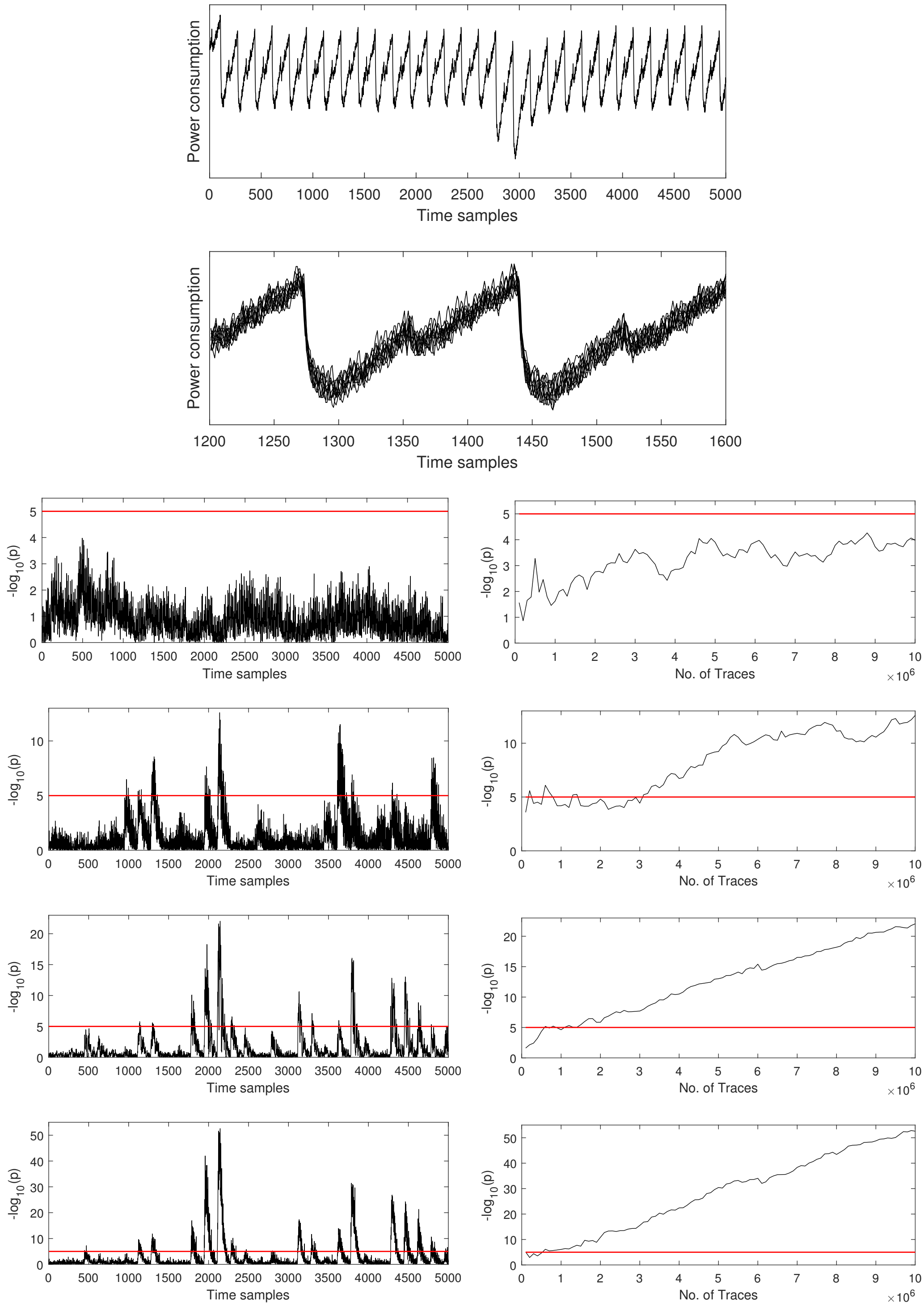


Figure 9: Univariate leakage assessment using 10 000 000 traces (step size 100 000) of a serialized PRESENT threshold implementation. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) first-order t-test results, 4) second-order t-test results, 5) third-order t-test results, 6) χ^2-test results.

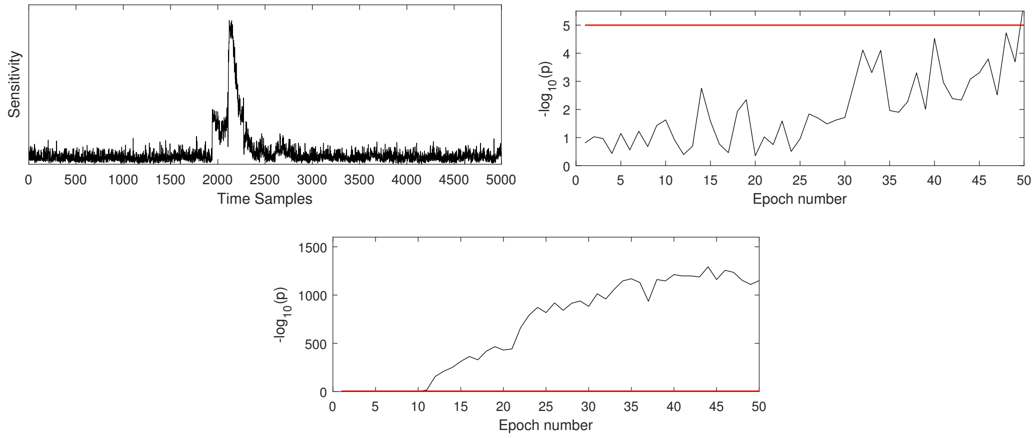


Figure 10: Sensitivity Analysis and DL-LA using 500 000 (Fig. 10(b)) and 3 000 000 (Fig. 10(a), 10(c)) traces of a serialized PRESENT threshold implementation respectively. For each p value a validation set of 1 500 000 traces is considered.

Please note that for this case study and the upcoming ones, where we analyze protected implementations, we change the visualization of the DL-LA results. Due to the large data sets involved it is not feasible to train many different classifiers with a steadily increasing size of the training set over many steps. Instead we visualize the $-\log_{10}(p)$ values over the number of epochs (instead of over training traces). We do this twice, once for a number of traces below the minimum required by the classical assessment method (here χ^2-test with 600 000 traces) and once for a larger training set in order to show that much larger confidence values can be achieved in the protected cases as well. Those results are presented in Figure 10. In case of a training set of size 500 000, the DL-LA succeeds only just in overcoming the confidence threshold. However, barring the possibility of a false positive results (which is highly unlikely), the confidence could be increased by an evaluator either by considering more epochs or by increasing the validation set. In the case of a training set including 3 000 000 measurements, the confidence that side-channel leakage is present becomes extremely large. In fact much larger than the confidence results achieved by the t- and χ^2-test given all 10 000 000 traces.

Case Study 5: PRESENT TI (FPGA), misaligned Traces

This case study is equivalent to the previous one apart from the fact that we artificially created a misalignment of the traces, as it was already done for case study 2. As a result of this misalignment the leakage detection approaches require slightly more traces to overcome the confidence threshold than in the aligned case. In particular, as shown in Figure 11, the second-order and the third-order t-test require 3 600 000 and 1 500 000 traces for the detection respectively, while the χ^2-test succeeds after only 800 000 traces and again results in a much higher confidence. The DL-LA is again the most powerful leakage detection mechanism and succeeds for both sizes of the training set (800 000 and 3 000 000) with much higher confidence values than any of the classical approaches (cf. Figure 12).

Case Study 6: PRESENT Multivariate TI (FPGA), aligned Traces

In the next case studies we concentrate on scenarios where the classical univariate detection approaches are naturally unsuited to detect leakage, namely purely multivariate higher-order leakages. These cases are the primary motivation to apply DL-LA in reality, as all

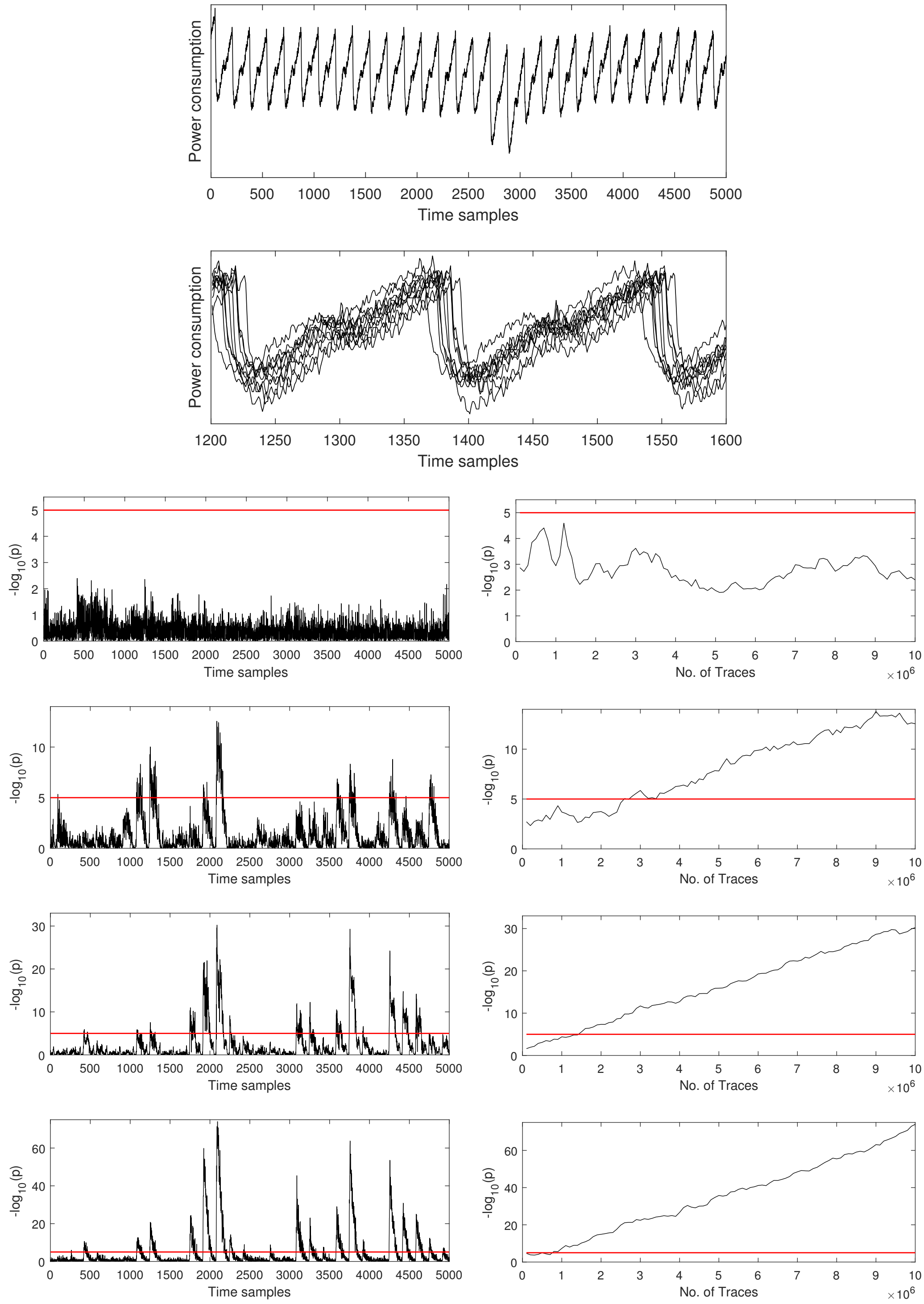


Figure 11: Univariate leakage assessment using 10 000 000 misaligned traces (step size 100 000) of a serialized PRESENT threshold implementation. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) first-order *t*-test results, 4) second-order *t*-test results, 5) third-order *t*-test results, 6) χ^2-test results.

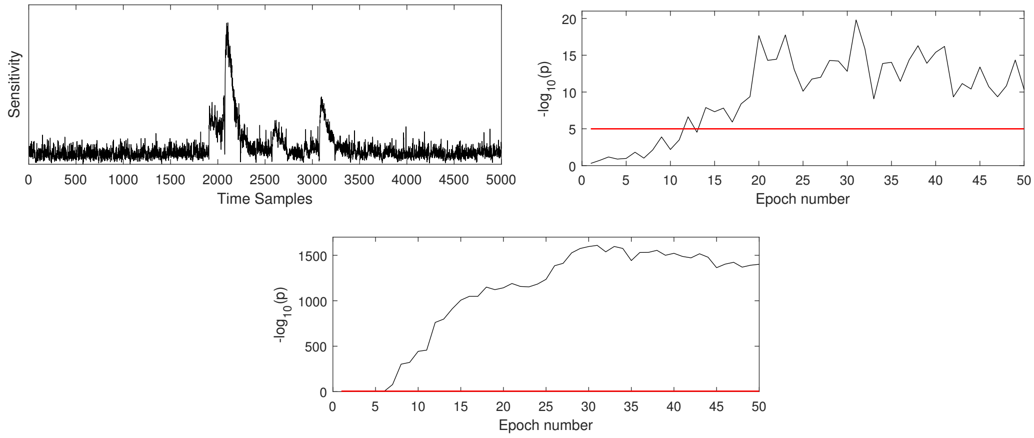


Figure 12: Sensitivity Analysis and DL-LA using 800 000 (Fig. 10(b)) and 3 000 000 (Fig. 10(a), 10(c)) misaligned traces of a serialized PRESENT threshold implementation respectively. For each p value a validation set of 1 500 000 traces is considered.

other commonly applied methods to the best of our knowledge fail to capture the whole amount of present side-channel leakage in these scenarios (at least without significant manual effort and knowledge about the underlying implementation). We provide evidence for this statement in the following.

We constructed a special version of the PRESENT threshold implementation architecture depicted in Figure 8, that does not offer univariate side-channel leakage. To this end we had to ensure that all six component function (G_1, G_2, G_3, F_1, F_2 and F_3) are evaluated sequentially and not in parallel. We did this by gating their respective inputs with `AND` gates which are controlled by a finite state machine (FSM). In addition to that we had to make sure that none of the state registers are clocked at the same time. Thus a single Sbox computation takes 7 clock cycles in the resulting hardware design. As expected, our univariate leakage assessment using the classical detection approaches does not indicate the presence of any side-channel leakage (cf. Figure 13). However, a multivariate investigation could still find higher order leakage if performed at the correct offsets. This requires either white-box knowledge about the implementation or must be determined by exhausting all possibilities. The result for the best offset leading to detectable multivariate leakage is illustrated in Figure 14 and shows leakage in the third order after more than 45 million traces. Please note that we have performed each multivariate second-order t-test, third-order t-test and χ^2-test with the correct offsets (as we know all the implementation details) and none of them was able to detect leakage with fewer traces.

In stark contrast, as apparent in Figure 15, DL-LA provides a very high confidence level of $-\log_{10}(p) > 150$ for the presence of side-channel leakage after training on only 20 million traces. We still retained the same network architecture as before and made absolutely no assumptions about the leakage and required no white-box knowledge about the offset of the individual evaluation of TI shares. Due to memory and time restrictions we pruned the traces to a length of 500 sample points (1281-1780). Validation took place on 5 million traces. Leakage becomes apparent after 25 epochs and continuously increases until our chosen threshold of 50 epochs has been reached.

Case Study 7: PRESENT Multivariate TI (FPGA), misaligned Traces

Our next case study is a replication of the previous one, but again we misaligned the traces through bad triggering. As shown in Figure 16 no univariate detection of leakage succeeds.

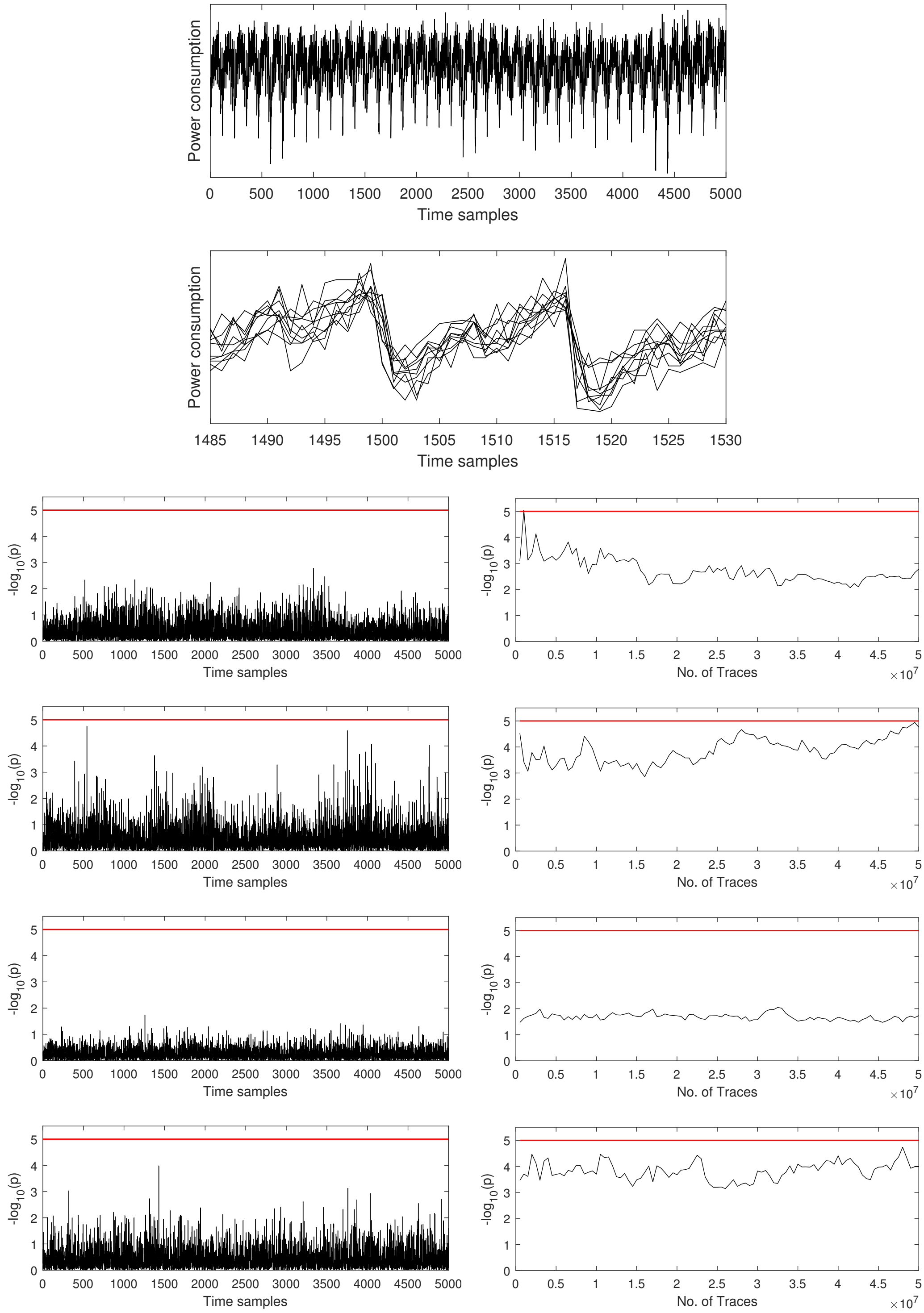


Figure 13: Univariate leakage assessment using 50 000 000 traces (step size 500 000) of a serialized multivariate PRESENT threshold implementation. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) first-order t-test results, 4) second-order t-test results, 5) third-order t-test results, 6) χ^2-test results.

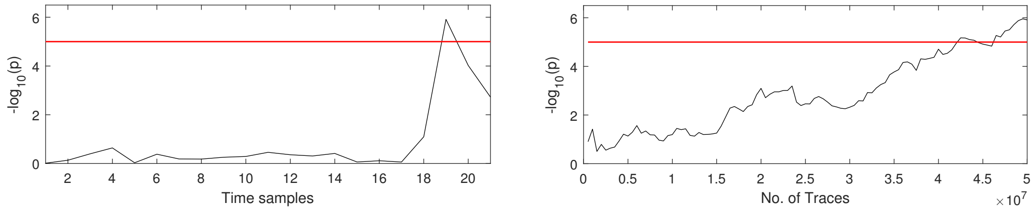


Figure 14: Multivariate third-order t-test using 50 000 000 traces (step size 500 000) of a serialized multivariate PRESENT threshold implementation.

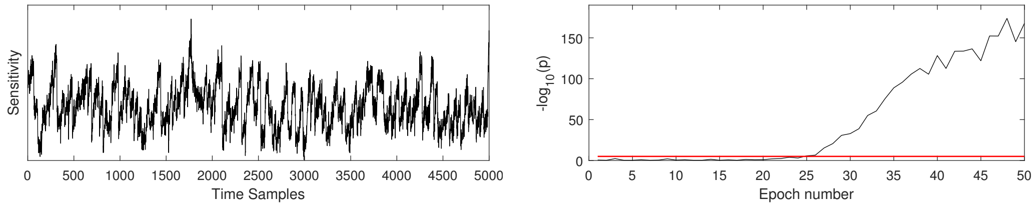


Figure 15: Sensitivity Analysis and DL-LA using 20 000 000 traces of a serialized multivariate PRESENT threshold implementation. For each p value a validation set of 5 000 000 traces is considered.

In this case however, even the multivariate third-order analysis with the best possible offset for leakage detection in the previous case study does not succeed (cf. Figure 17). In other words, the acquired set of traces does not allow detection of any leakage using conventional methods, at least in case the traces are not re-aligned before the analysis.
DL-LA however detects leakage with high confidence ($-\log_{10}(p) > 60$) after training on only half of the available traces (25 000 000). This result is depicted in Figure 18.

Case Study 8: PRESENT TI (ASIC), aligned Traces

As a complement to the FPGA-based case studies outlined on the previous pages, we have investigated a PRESENT threshold implementation realized in non-reconfigurable hardware as well, namely as part of a custom 40 nm ASIC prototype. The test chip has been developed for SCA evaluations and features several different cipher cores integrated into a larger control framework. The PRESENT implementation is the same nibble-serialized threshold implementation that has been investigated in case studies 4 and 5 already. We have measured 50 000 000 traces in a fixed-vs-fixed manner with 2000 sample points per trace. Results of a first-, second- and third-order univariate t-test as well as a χ^2-test are depicted in Figure 19. As expected, no first-order leakage can be observed, but second-order leakage can be detected consistently beyond the confidence threshold after approximately 8 000 000 traces. Since there is no detectable leakage present in the third-order, it is no surprise that the χ^2-test requires more traces to distinguish the distributions, namely about 15 000 000 traces.
However, DL-LA outperforms both methods by a large margin as it successfully classifies enough validation traces correctly to achieve a huge confidence after training on only 100 000 traces. This result is shown in Figure 20. It demonstrates that the leakage detection capability of our approach is not limited to the FPGA-based case studies. In fact, on the ASIC measurements it achieves one of the most impressive results compared to the classical detection approaches.

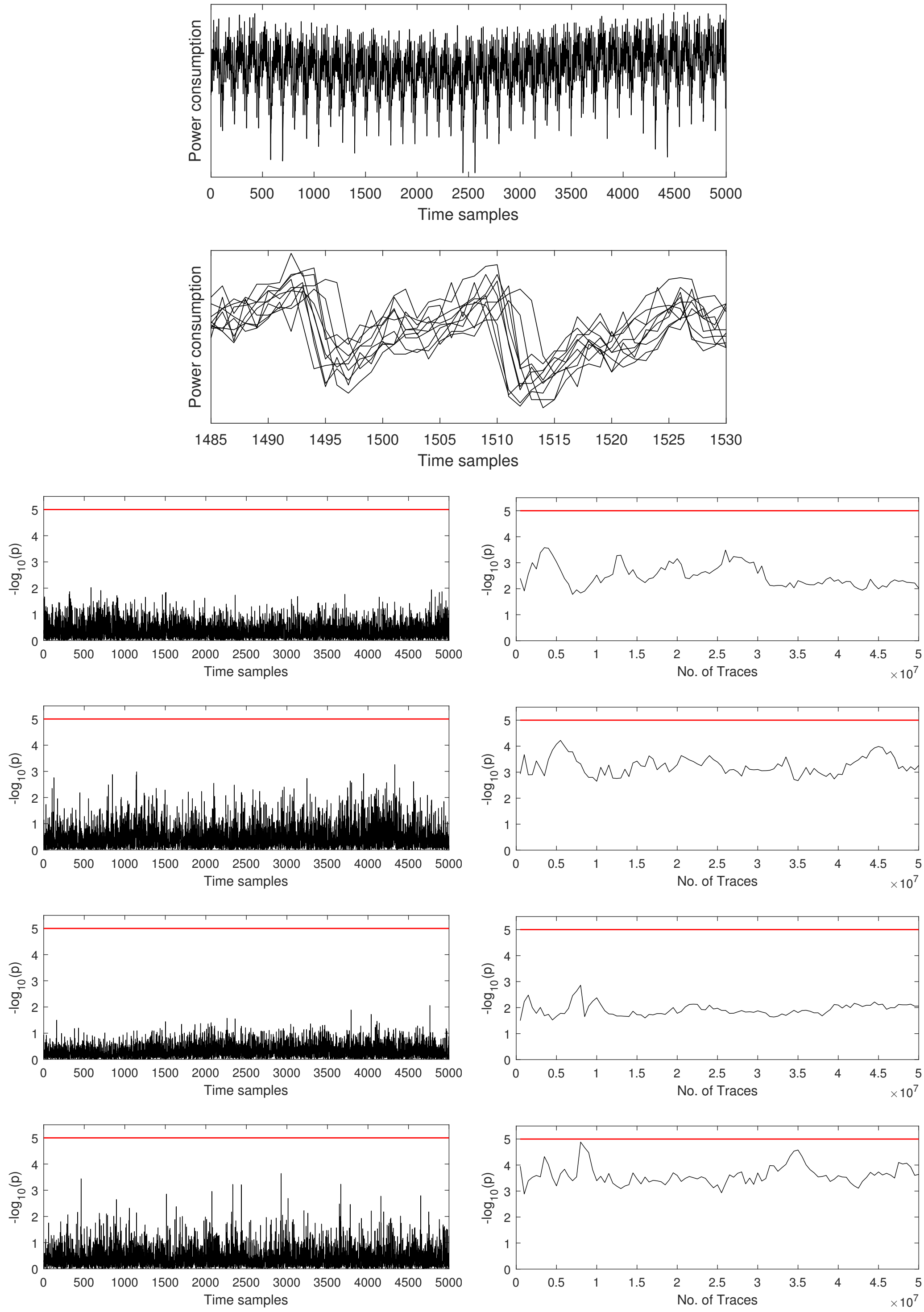


Figure 16: Univariate leakage assessment using 50 000 000 misaligned traces (step size 500 000) of a serialized multivariate PRESENT threshold implementation. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) first-order *t*-test results, 4) second-order *t*-test results, 5) third-order *t*-test results, 6) χ^2-test results.

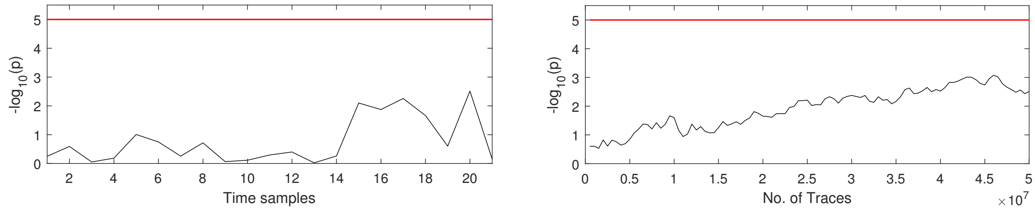


Figure 17: Multivariate third-order t-test using 50 000 000 misaligned traces (step size 500 000) of a serialized multivariate PRESENT threshold implementation.

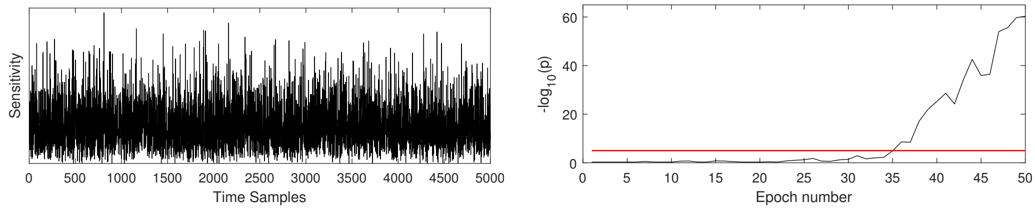


Figure 18: Sensitivity Analysis and L-LA using 25 000 000 misaligned traces of a serialized multivariate PRESENT threshold implementation. For each p value a validation set of 5 000 000 traces is considered.

Case Study 9: PRESENT TI (ARM Cortex-M0 µC), aligned Traces

Finally, we want to evaluate our methodology and network structure against a protected *software* implementation. As a target we have chosen a PRESENT threshold implementation in software, as suggested in [SBM18], and implemented the design on an ARM Cortex-M0 microcontroller. Naturally, side-channel traces recorded on software platforms are longer in terms of sample points due to the much larger number of clock cycles required to execute a cryptographic primitive. We have collected 100 000 traces with 200 000 sample points each, which do not even contain the full first round of the cipher execution. In case of properly masked implementations (see [SBM18]) usually no univariate leakage is exhibited, but multiple sample points need to be combined in order to find input-dependent information. Nevertheless, we start by applying the univariate distinction tests as a first step. The results are depicted in Figure 21. No consistently detectable leakage can be found by any of the four different methods (1st-, 2nd-, 3rd-order t-test, χ^2-test). However, when observing the progress of the maximum t-value over the whole number of points it is obvious that the confidence threshold is exceeded multiple times in all four of them. We claim that this effect is not caused by actual detectable side-channel leakage, but rather due to the excessive trace length of 200 000 sample points and the ineptitude of point-wise methods to estimate the confidence for a whole trace (without manual adjustments). This is discussed in more detail in Section 5.

While no univariate leakage is (robustly) detectable in the trace set, a second-order multivariate t-test applied with the correct offsets rejects the null hypothesis with confidence after merely 800 traces, as demonstrated in Figure 22. Since DL-LA can exploit multiple occurrences of such multivariate second-order leakage across the whole trace length at once, it again outperforms the classical approaches and requires a training set of only 500 traces to achieve a higher confidence. This result is shown in Figure 24. As a conclusion, neither the excessive trace length, nor the different architecture affects the detection capability of our approach negatively. While manual (or exhaustive) search for the correct offsets is required for the classical detection approaches, DL-LA does not require any additional

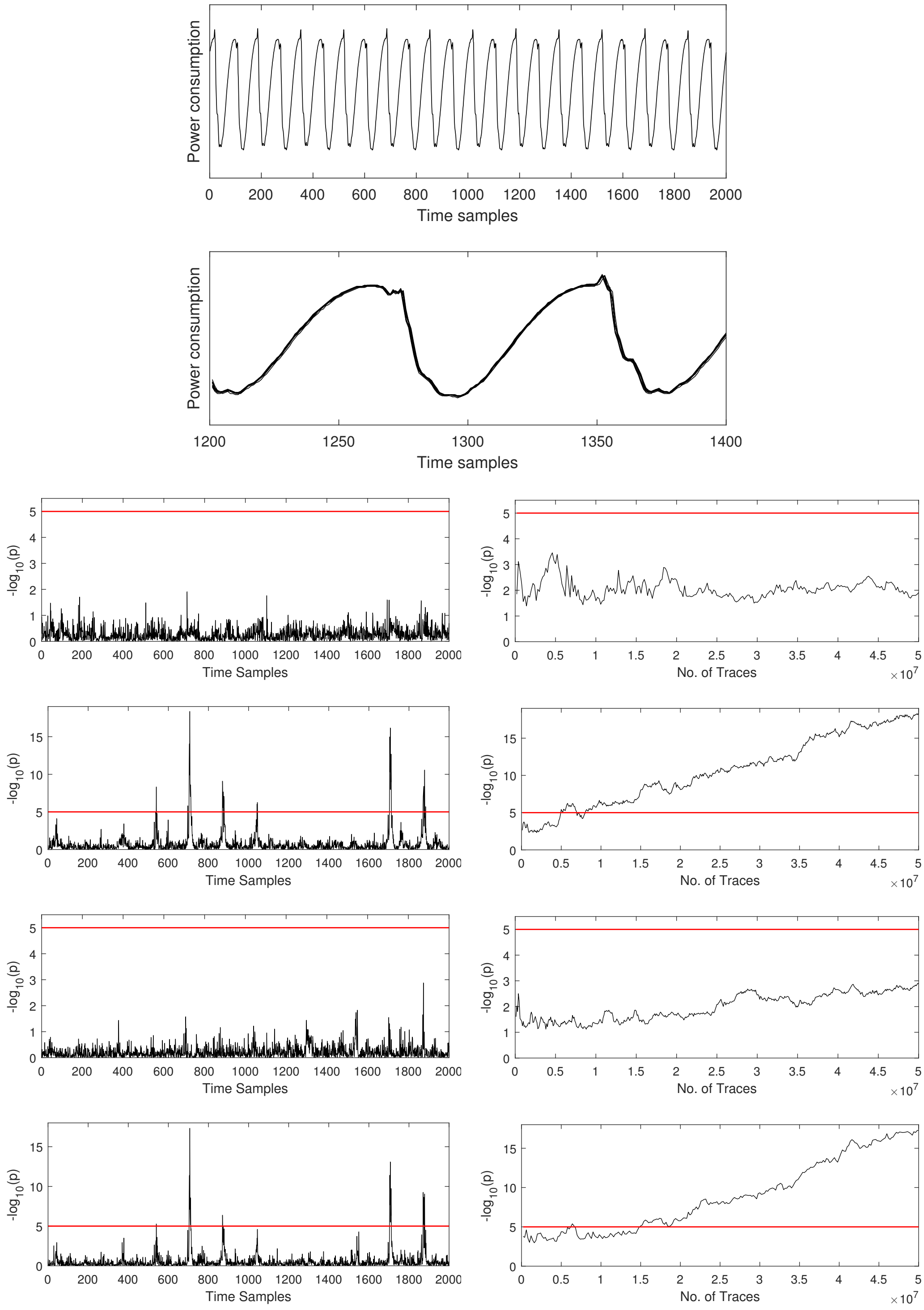


Figure 19: Univariate leakage assessment using 50 000 000 traces (step size 100 000) of a PRESENT threshold implementation on a 40 nm ASIC prototype. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) first-order t-test results, 4) second-order t-test results, 5) third-order t-test results, 6) χ^2-test results.

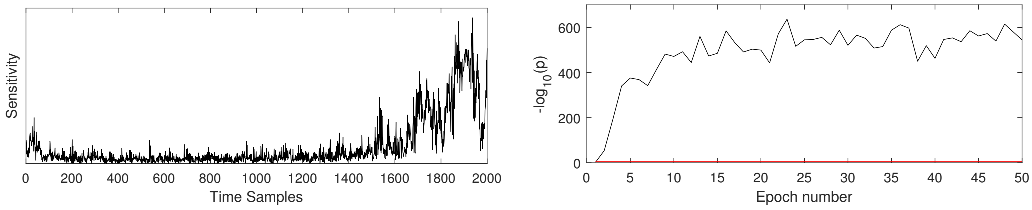


Figure 20: Sensitivity Analysis and DL-LA using 100 000 traces of a serialized PRESENT threshold implementation on a 40 nm ASIC. For each p value a validation set of 500 000 traces is considered.

Table 3: Comparison of the different leakage assessment techniques based on the number of required traces.

No. of Traces	Case Study 1	Case Study 2	Case Study 3	Case Study 4	Case Study 5	Case Study 6	Case Study 7	Case Study 8	Case Study 9
t-test	40	90	2 000	1 100 000	1 500 000	46 500 000	$>$ 50 000 000	8 100 000	800
χ^2-test	90	150	4 900	600 000	800 000	$>$ 50 000 000	$>$ 50 000 000	15 000 000	3 600
DL-LA (tr.)	$<$ 10	20	150	500 000	800 000	20 000 000	25 000 000	100 000	500
DL-LA (tr. + val.)	$<$ 10 010	10 020	10 150	2 000 000	2 300 000	25 000 000	30 000 000	600 000	80 500

information and trains a successful classifier on the raw data.

Overview

In order to enable an easy comparison between the classical methods and the MLP-based DL-LA results across all 9 case studies we have listed the required amount of traces for each analysis in Table 3. The table distinguishes between the detection traces (i.e., the training set) and the evaluation traces (i.e., the sum of the training and the validation set). The number of detection traces required for a confident result are lower or as low as that of the traditional methods. Yet, the number of traces required for the evaluation is often higher than that of the conventional methods, especially for the more trivial case studies. In the 3 case studies where the classical methods require the largest amount of traces for a detection (namely CS6, CS7 and CS8), even the combined set used for DL-LA is significantly smaller than the numbers required for t- and χ^2-test. This confirms that DL-LA is especially beneficial in the more complex, noisy and countermeasure-protected cases. Also, please note that in the case studies where only multivariate leakage is present (namely CS6, CS7 and CS9) DL-LA is compared to multivariate extensions of the t- and χ^2-test, which require manual effort and in-depth knowledge about the implementation, while DL-LA works on the raw traces without any additional information.

We also analyzed the computation times required for the classical leakage detection methods and DL-LA. Table 4 provides a comparison in that regard. We provide numbers achieved on a server that features 256 GB RAM and 2 × Intel Xeon E5-2650 v3 CPUs with 40 combined threads. For DL-LA we repeat the same evaluation while additionally utilizing a Tesla K80 GPU. All run times have been acquired by measuring the execution time of the respective C++ and Python scripts using the *std::chrono* library and the *time* module respectively and normalizing the resulting time periods by the number of traces

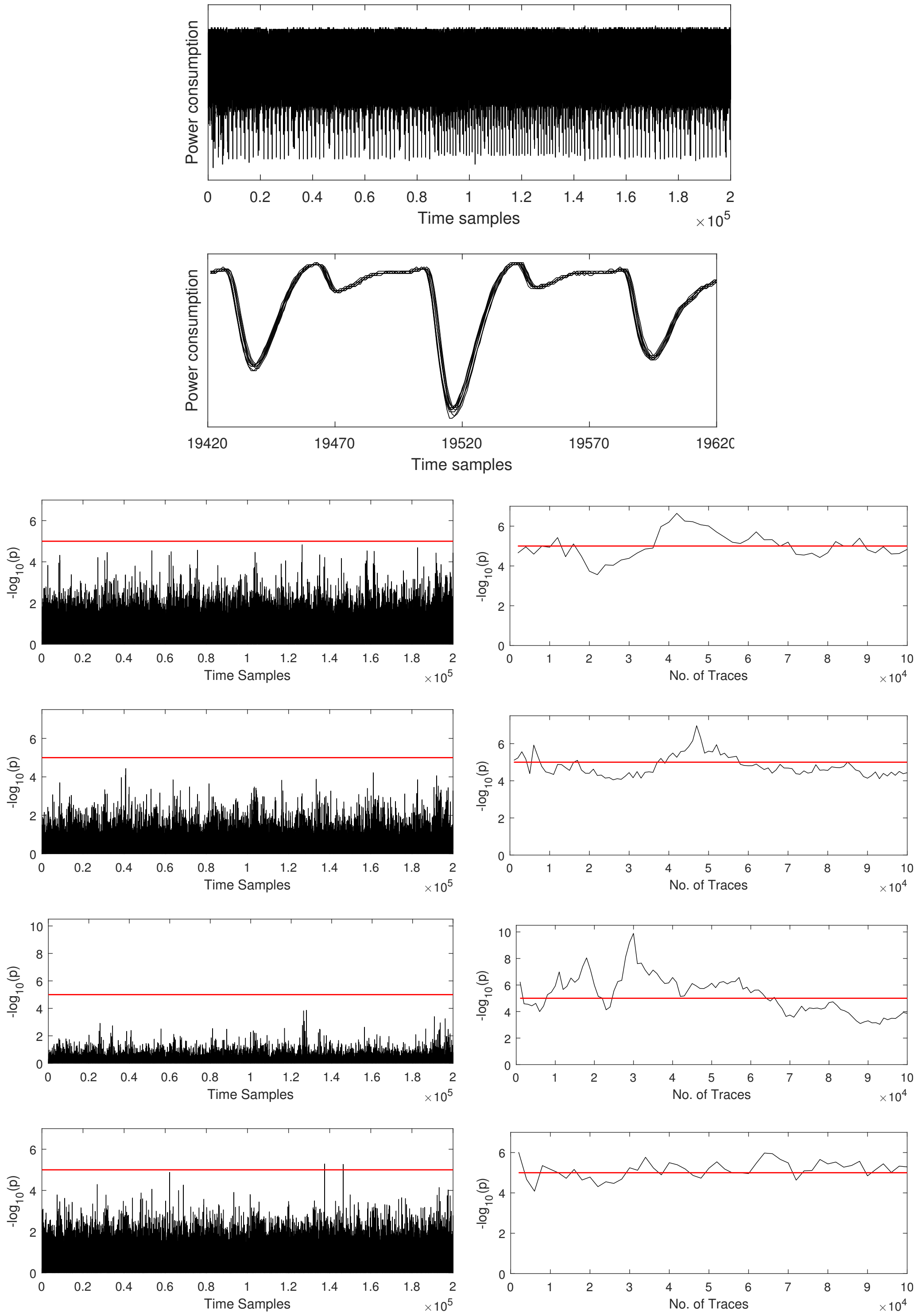


Figure 21: Univariate leakage assessment using 100 000 traces (step size 1 000) of a PRESENT threshold implementation in software (ARM Cortex-M0). From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) first-order *t*-test results, 4) second-order *t*-test results, 5) third-order *t*-test results, 6) χ^2-test results.

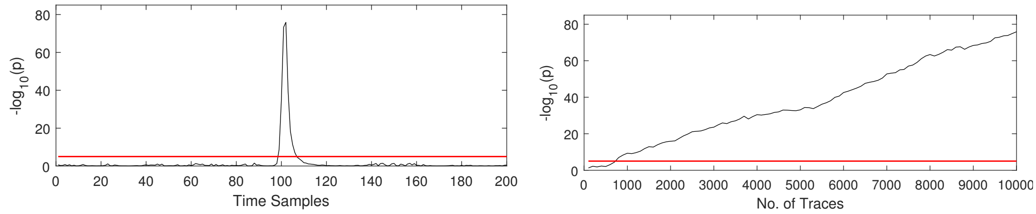


Figure 22: Multivariate second-order t-test using 10 000 traces (step size 100) of a PRESENT threshold implementation in software (ARM Cortex-M0).

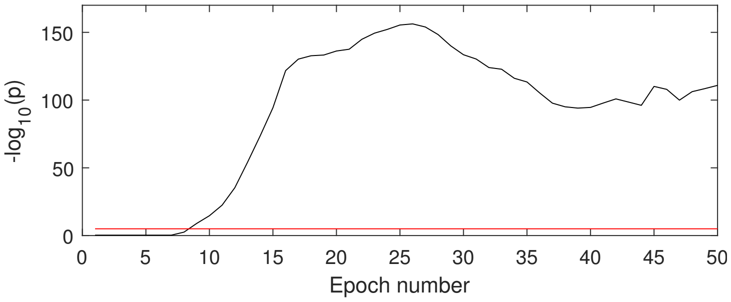


Figure 23: DL-LA using 500 traces of a PRESENT threshold implementation in software (ARM Cortex-M0). For each p value a validation set of 80 000 traces is considered.

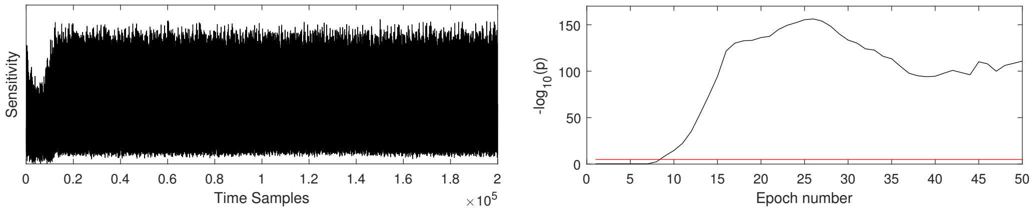


Figure 24: Sensitivity Analysis and DL-LA using 500 traces of a PRESENT threshold implementation in software (ARM Cortex-M0). For each p value a validation set of 80 000 traces is considered.

Table 4: Computation times for the different leakage detection methods on our hardware equipment.

Hardware	CPU 2 × Intel Xeon E5-2650 v3	GPU 1 × Tesla K80
univariate t-test (orders 1-3)	8.95 seconds / 1M traces	-
univariate χ^2-test	6.65 seconds / 1M traces	-
DL-LA loading/preparing traces	133.67 seconds / 1M traces	133.67 seconds / 1M traces
DL-LA training 1 epoch	21.47 seconds / 1M traces	11.75 seconds / 1M traces
DL-LA validation 1 epoch	13.06 seconds / 1M traces	9.57 seconds / 1M traces

1M = 1 000 000, assuming traces with 5 000 sample points and 8-bit resolution that have been processed. The *t*-test run times include fetching the traces from the hard drive, creating the histograms, computing the statistical moments 1-3, evaluating the *t*-statistics and degrees of freedom and finally calculating the confidence using multi-precision. Similarly, the χ^2-test numbers include fetching the traces from the hard drive, creating the histograms, evaluating the χ^2-statistics and degrees of freedom and finally calculating the confidence using multi-precision. For both implementations we made use of the C++ implementations using the Boost library provided by [MRSS18]. In that regard, please note that the run times depend on the decimal precision set in the Boost library, which affects the minimum p value that can be expressed. For DL-LA we have separated the run time into three different parts. In a first step the traces need to be loaded and prepared. This step has to be performed once per DL-LA evaluation and does not benefit from utilizing the GPU. The next two steps are training on the training set and validating on the validation set. Both of those run times are given *per epoch*. As an example, consider case study 4. Here, 10 million traces are analyzed by the *t*- and χ^2-test. Accordingly, the computation of the *t*-test (orders 1-3) took approximately 1 minute and 30 seconds and the χ^2-test took about 1 minute and 7 seconds. The DL-LA results (those with high confidence in Figure 10(c)) have been obtained by loading and preparing 4.5 million traces and then training on 3 million of those traces while validating on the remaining 1.5 million traces for 50 epochs. This equates to 1 hour, 20 minutes and 3 seconds[7] without GPU support or 51 minutes and 23 seconds[8] with GPU support. Clearly, DL-LA exceeds the run time for the classical univariate tests by orders of magnitude. However, the univariate tests can not detect any leakage in case studies 6, 7 or 9. We do not provide any run times for the multivariate extensions of *t*- and χ^2-test here, since their computational effort depends significantly on the amount of prior knowledge about the target implementation. Any exhaustive approach that does not require prior knowledge or manual effort, would exceed the run time of DL-LA by far. Therefore, we believe that the computation time of DL-LA is well spent in scenarios where leakage detection is not trivial.

Please note that we did not spend any particular time or effort to optimize the run time for any of the methods involved. We are certain that better performances can be achieved for all of the different techniques.

CNN Results and Hyper-Parameter Dependence

In addition to the MLP-based results presented in the individual case studies, we have applied our CNN on most of the respective measurement sets as well. In fact, we have done so for eight different hyper-parameter configurations introduced in Section 3.3.2. We varied the sizes of kernel and strides as a double or triple of the points in one clock cycle and the absolute pool size between two and four.

Figure 25 depicts our results: As before, all confidence results are given as $-\log_{10}(p)$ values. On the very left in light-green it can be observed that confidence values between 280 and 350 are achieved for 1000 training and 1000 validation traces on the trace sets corresponding to case study 1 (aligned traces of unprotected PRESENT on FPGA). Note that in this case 100% validation accuracy is often achieved and we limited the precision of the confidence computation to 10^{-350}. In leave-green we depict the confidence for 1000 training and 1000 validation traces corresponding to case study 2 (unprotected PRESENT with misaligned trigger on FPGA), which falls between 280 and 350 as well. Next, in dark-green the confidence achieved by the different CNN structures for 10 000 training and 70 000 validation traces corresponding to case study 3 (unprotected PRESENT with clock misalignment on FPGA) are shown. They range from 250 to 350. The confidence for

---

[7] 00:10:02 for loading/preparing 4.5 million traces, 00:53:41 for training on 3 million traces for 50 epochs and 00:16:20 for validating on 1.5 million traces for 50 epochs.

[8] 00:10:02 for loading/preparing 4.5 million traces, 00:29:23 for training on 3 million traces for 50 epochs and 00:11:58 for validating on 1.5 million traces for 50 epochs.

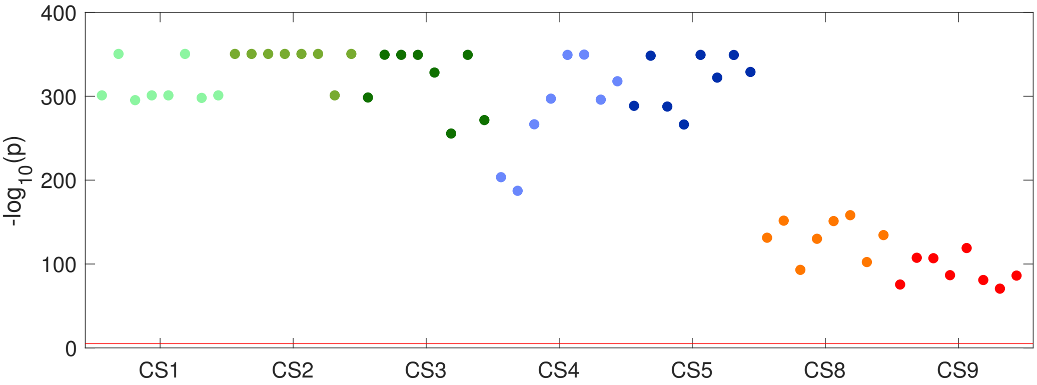


Figure 25: Confidence results achieved by the CNN network across eight hyper-parameter configurations for seven case studies. Each case study is depicted in a different color.

the aligned and trigger-misaligned PRESENT TIs (case studies 4 and 5) are depicted in light-blue and dark-blue based on 500 000 respectively 800 000 training traces and 500 000 validation traces each. Here, the confidence ranges from 180 to 350. Due to the high data complexity of case studies 6 and 7 (i.e., training on $\geq$ 25 000 000 traces) we did not evaluate them for all hyper-parameters and omitted them in this figure. Second to last, the confidence results for 100 000 training and 200 000 validation traces corresponding to the PRESENT TI on an ASIC (case study 8) range from 93 to 158 (orange). Lastly, we obtained $-\log_{10}(p)$ values between 70 and 119 for 1 000 training and 80 000 validation traces corresponding to case study 9 (Software TI of PRESENT) shown in red color. The red horizontal line indicates the confidence level of 10^{-5} that is commonly used as a leakage indication. In summary, the suggested CNN is able to detect leakage with high confidence in each of the tested case studies (from 3 different platforms) for any of the applied hyper-parameter configurations. These results showcase the robustness of DL-LA as a leakage assessment methodology across different network architectures and even different choices of hyper-parameters.

5 Discussion

Before coming to a conclusion we discuss a few important aspects about the evaluation.

False Positives. False positives commonly appear as a problem in classical leakage evaluations. We say that a t-test or χ^2-test result is falsely positive in a leakage detection scenario if the confidence threshold is exceeded for at least one sample point, despite the absence of leakage. In other words, a false positive occurs when the test decides to reject the null hypothesis for at least one sample point where it is in fact true [WO19]. This phenomenon is caused by the point-wise independent nature of classical detection methods. A threshold of $p_{\text{th}} = 10^{-5}$ set for each individual point will lead to an aggregation of the error probability over the length of the entire trace, thereby lowering the confidence. More formally, the likelihood that a false positive occurs at least once in a trace of length K can be described as (assuming independence between the tests [WO19]):

$$P(\text{false positive}) = 1 - (1 - p_{\text{th}})^K$$

For the typical value of $K = 5\,000$ in many of our case studies and the common threshold of $p_{\text{th}} = 10^{-5}$ this formula equates to 0.0488. Thus, the probability that the detection threshold is falsely exceeded for at least one sample point is roughly 5% (when using the common methodology for t- and χ^2-test). While the evaluator may have desired a confidence of $1 - 10^{-5} = 99.999\%$ in the reported leakage by setting $p_{\text{th}} = 10^{-5}$, the actual result can provide a confidence of only $1 - 0.0488 = 95.12\%$ when considering the full trace length. For longer traces the situation is even worse. Case study 9, for example, evaluates traces of $K = 200\,000$ samples points each. Assuming independence between the tests at each point, the probability for a false positive to occur is greater than 86%[9]. The impact of this large probability can be observed in Figure 21, where the point-wise methods exceed the threshold more often than not despite the apparent absence of univariate leakage. Hence, a manual investigation of the individual leakage points is often necessary when performing classical leakage detection to exclude false positives. Whitnall and Oswald suggest multiple different solutions to this fundamental problem in their recent work, including the Bonferroni correction, the Ŝidák correction and the Holm procedure [WO19]. They also conclude that these correction techniques inevitably increase the risk of false negatives, which is undesirable from an evaluators point of view. Hence, no perfect solution exists to fix point-wise methods in this regard.

In contrast, our deep learning based methodology does not produce several individual univariate statistical tests, but produces a single decision metric based on the entirety of points[10]. Hence, no correction of the confidence is required, independent of the length of the traces to be analyzed. When DL-LA suggests to reject the null hypothesis, then the methodology has already produced a classifier which is able to distinguish the two groups under analysis with statistical significance. In our belief, this is the most convincing evidence for distinguishability that any evaluator can hope for and goes beyond statistical arguments. To practically verify the resilience of DL-LA against false positives, we trained our networks several times on randomly generated data with random group assignments. In these evaluations we never observed any confidence exceeding $p = 10^{-2}$. Hence, we have a high confidence, that false positives are far less likely to occur with our methodology if a reasonable threshold is chosen, e.g., $p_{\text{th}} = 10^{-5}$.

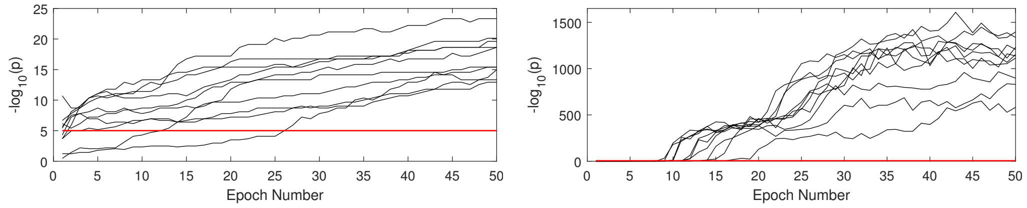
False Negatives. Any leakage evaluation methodology should primarily aim to prevent false negatives. We say that a leakage detection result is falsely negative if the procedure does not report leakage with a confidence level above the detection threshold, despite the presence of leakage, or even worse, despite the target being vulnerable to attacks. Statistically, a false negative occurs when the test decides to accept the null hypothesis while it is in fact false. Clearly, when relying purely on univariate distinction tests, any multivariate higher-order leakage causes a false negative, in the sense that the device leaks input dependent information while the test is unable to detect it as it can not be detected with univariate methods regardless of the amount of acquired traces (unless multiple consecutive cycles are interleaved due to frequency or setup manipulations [MM13]). This has been one of the core motivations for this work. When DL-LA is employed, the risk of overlooking temporally distributed leakages is significantly reduced. Although it is never possible to guarantee the absence of leakage, the confidence in the security of a device under test can be gradually increased by acquiring multiple sets of traces in a fixed-vs-fixed manner with different selections of the fixed inputs while analyzing those trace sets using the t-test, χ^2-test and DL-LA. If required, the DL-LA procedure can even be reiterated using different networks to be trained.

Despite the discussion above, it is noteworthy that the probabilistic nature of machine learning procedures adds an additional, potentially false-negative-causing, element to the analysis which does not exist in deterministic methods. Namely, when training a classifier in the DL-LA procedure over multiple epochs, the training data is randomly regrouped in

---

[9] Please note that the nature of leakage measurements makes it unlikely that the tests are indeed fully independent in practice, so the real probability could be lower [WO19].

[10] Note that pinpointing leakage in the time dimension is still possible due to sensitivity analysis as demonstrated in our case studies.



(a) using SCA traces from case study 1 (b) using SCA traces from case study 4

Figure 26: Repeating DL-LA multiple times on the same data using the same network structure and parameters. Left) training on 50 traces and validating on 300 traces; Right) training on 3 000 000 traces, validating on 1 500 000 traces. Differences are caused by the probabilistic learning procedure.

each epoch. Therefore, when performing DL-LA using the same network and parameters multiple times on the same data, the results will not be exactly the same. The order in which the shuffled traces are seen by the training procedure causes small differences in the trained weights due to nature of the backpropagation strategy. Hence, there is a probabilistic element which has an impact on the detection success and the resulting confidence. Of course, it would be highly undesirable for an evaluator if this situation causes the number of required traces for the detection to vary significantly when analyzing the same data several times. Thus, in order to evaluate whether this may become a problem on our data sets, we have exemplarily analyzed two of our case studies in this regard, namely case studies 1 and 4. The results can be seen in Figure 26. We have repeated the DL-LA procedure 10 times for each of the case studies for a fixed set of parameters. The differences in the detection performance are obvious, despite all parameters and the underlying data being identical. The results clearly show that for a small number of epochs (< 20 here) it can definitely occur that for a fixed size of training and validation set, the procedure succeeds in some iterations in detecting leakage and in others not. However, this inaccuracy can be avoided in parts by increasing the number of epochs. When choosing the number of epochs sufficiently large (we suggest ≥ 50), it becomes unlikely that the random shuffling during the training procedure causes false negative results. The trade-off between the computational effort (large number of epochs) and the risk of receiving false negatives (small number of epochs) can be controlled by the evaluator.

Confidence Boosting. In a common leakage evaluation a statistical test (t-test, χ^2-test) is performed on the entirety of collected traces. Thereby, two different metrics, (i) the number of traces required to extract meaningful information, and (ii) the level of confidence the evaluator wants to achieve are tightly intertwined. More specifically, under realistic noise conditions t-test and χ^2-test are fundamentally unable to answer the question: Given a very high confidence threshold of $p = 10^{-50}$ can the attacker extract information given only very few traces? In stark contrast, our deep learning based methodology operates on two sets: One training set of size N and one validation set of size M. Here, N and M can be chosen independently from each other. While N represents the actual amount of traces available to an attacker, M should be chosen sufficiently large to reach the desired level of statistical confidence, e.g. note that the maximum level of statistical confidence that can be achieved with a given validation set equals 0.5^M and might be much lower under realistic noise conditions.

Sensitivity Analysis. As seen in Section 4 computing the gradient of an output component of the neural network with respect to the input values can provide an insight into the dependence of the classification result on each individual time sample. While this seems similar to the result of classical univariate hypothesis tests, which illustrate independent statistical tests on each point in time, there are some crucial differences: DL-LA learns a function depending on the inputs in some way that minimizes the given loss function. This leads to two effects: (1) Points that do not contribute to leakage may still receive a non-zero component in the gradient, (2) Points that contribute to leakage, but correlate heavily with other points contributing to leakage might not be learned, as there is no intrinsic incentive for the neural net to learn redundant information. However, our practical case studies show, that the highest values in the Sensitivity Analysis typically correspond to leakages which are also found by traditional analyses. Yet, we want to caution against the idea that all leakage locations can be found with a single SA (c.f. CS4, CS5, CS8). Instead, the process is more iterative: After a design flaw has been identified and fixed, the DL-LA of the next design iteration might reveal new leakage locations of flaws that already persisted in the initial evaluation, but were simply not learned by the classifier.

Validation Accuracy in Isolation. Commonly, neural networks are applied to classification tasks in which the user is actually interested in obtaining a good classifier, e.g., obtain a network to distinguish cat pictures from dog pictures. In those cases a very high validation accuracy $(0.99 + \epsilon)$ is expected from a suitable neural network as each individual sample is noise free and can easily be assigned to one specific group. In contrast, when evaluating side-channel traces, especially of masked implementations, the randomized intermediate values lead to an impossibility to precisely assign each individual sample to a group with high accuracy.[11] In contrast, the aim of the attacker can only be to distinguish different processed intermediate values statistically, i.e., *on average*. This leads to a very different expectation (compared to the image classification problem): The aim is to find a network that works better than chance (validation accuracy > 0.5) and does so consistently over a large validation set. Hence, we caution the evaluator to disregard seemingly small values for the validation accuracy, e.g. 0.505. Instead, the size of the (perfectly balanced) validation set should always be taken into account by computing the correct p-value according to the Binomial distribution.

Test and Validation Set. In deep learning, there is a common distinction between the validation set, used as a feedback mechanism to adjust the hyper parameters and the test set, another completely independent set that is used to access the accuracy of the final network. This approach is used to prevent implicit information leakage from the validation set into the trained model (through the adjustment of hyper parameters). For our case studies this distinction is not needed, because we performed all evaluations on networks with identical hyper parameters and the chosen network architectures are not adjusted or specialized by any means.

Misalignment. As seen in our case studies, DL-LA is resilient against slight misalignment through bad triggering. However, this robustness is shared with the t- and χ^2-test. When operating on severely misaligned traces due to clock randomization both DL-LA based on MLPs and classical tests lose orders of magnitude of confidence compared to an aligned evaluation. Fortunately, this can be partially offset by increasing the validation set to perform *Confidence Boosting*. Alternatively, the loss of confidence can be compensated by performing DL-LA based on a CNN architecture, as showcased in Figure 25.

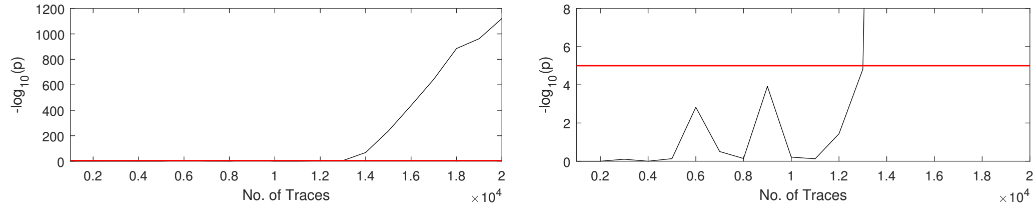
Targeted Block Cipher. For consistency reasons we have analyzed implementations of the PRESENT block cipher exclusively in all of our 9 case studies on 3 different target platforms. This brings the advantage that it is easier to compare the effectiveness of

---

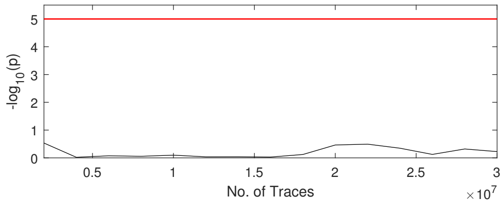
[11] In fact, if we find a neural network with validation accuracy equal to 1.0 an attacker would most likely be able to not only succeed with DPA, but mount a successful Simple Power Analysis (SPA).

countermeasures and the results from different devices with each other. However, we would like to stress that DL-LA is by no means limited to this choice. In fact, like the state-of-the-art methods for leakage assessment, namely t-test and χ^2-test, DL-LA is entirely independent of the cipher to be analyzed. Leakage assessment techniques simply try to distinguish two sets of measurements from each other based on statistical differences in the leakage distributions. If no dedicated protections are in place, such differences occur whenever physically manipulating different data values on a device. For leakage detection methods it is not important which particular operation causes such a difference or which exact data value is processed by that operation at any given moment in time. No modeling of the leakage of any specific operation or implementation part is required. It is not even important whether such a difference has any dependency on a secret variable (remember, leakage detection is not supposed to extract the secrets from an implementation). Leakage assessment is simply a tool for an evaluator to test whether - and what kind of - a dependency between the input given to an implementation and the recorded leakage exists. Therefore, the cipher running on the target device has no qualitative impact on the analysis. Since PRESENT, as an ultra-lightweight block cipher, is one of the most area and energy efficient cryptographic primitives [BKL$^+$07], it could be argued that more general block ciphers, like the Advanced Encryption Standard (AES), usually cause a larger power consumption per clock cycle. In that regard, the choice of the cipher may have a quantitative impact on the detection success, e.g., requiring fewer traces. However, this affects all leakage assessment techniques in the same manner and should not notably influence the comparison of different methods presented in this work. We believe it generally holds true that the choice of the countermeasure applied to a cipher has a much larger impact on the success of the leakage detection as it directly affects the noise and signal amplitude, the order and the variate of the leakage than the choice of the cipher itself. For this reason we have concentrated on one block cipher as a target in this work, but analyzed multiple countermeasures and device technologies.

Template Comparison. It is fair to wonder whether deep neural networks are the only viable solution to build the kind of classifiers required for the leakage detection procedure introduced in this work. In fact, any method that allows to build a binary classifier based on a set of labeled (fixed-vs-fixed) side-channel measurements which succeeds in classifying traces from a separate set with unknown labels better than randomly is theoretically applicable and can be plugged into our methodology. However, we suppose that it is difficult for any method not based on machine learning to provide the same flexibility and universality with respect to the type of leakage to be expected that DL-LA does. In order to investigate this expectation in more depth we provide a case study based on template analysis here. Using multivariate Gaussian templates to model the leakage patterns exhibited by a target implementation when different data values are processed is a well established technique in the side-channel community typically used for template attacks [CRR02]. Yet, the same principles can be applied to our leakage assessment procedure. The strategy is simple. For each of the two (fixed-vs-fixed) groups a multivariate Gaussian template over all time samples is created using all measurements in the training set that have been recorded for this particular input. Then, each measurement in the validation set is compared to the two templates and the likelihood for a match is calculated. A binary classifier is then achieved by simply assigning the trace to the group with the higher likelihood. Given the number of correct classifications and the size of the validation set, the confidence that leakage is detected can be calculated by the formulas given in Section 3. This method obviously shares some of the advantages of DL-LA. First of all, in contrast to univariate distinction tests it bases its classification on the whole trace at once and not on one individual time sample. In that regard, it also reduces the risk of false positives (as discussed earlier in this section) and is naturally



(a) using SCA traces from case study 1 (b) using SCA traces from case study 1 (zoomed)



(c) using SCA traces from case study 4

Figure 27: Multivariate leakage assessment using Gaussian templates. For the result on the top a validation set of 10 000 traces has been used. For the result on the bottom a validation set of 1 500 000 traces has been used.

capable of capturing multivariate and horizontal leakages. However, there are also some drawbacks, highlighted in the following example. We have applied the described analysis on two of our case studies from Section 4, namely case study number 1, based on the unprotected PRESENT core, and case study number 4, based on the PRESENT threshold implementation. Our templates are built using a multivariate Gaussian distribution. In the training phase we first compute the sample mean vector and the sample covariance matrix for the two groups. Since the underlying traces for both case studies have a length of $l = 5\,000$ sample points, the sample mean vectors $\overline{m_0}$ and $\overline{m_1}$ for the two fixed input classes are elements of $\mathbb{R}^l = \mathbb{R}^{5000}$ while the sample covariance matrices C_0 and C_1 are elements of $\mathbb{R}^{l \times l} = \mathbb{R}^{5000 \times 5000}$. In the validation phase, the profiles consisting of mean vector and covariance matrix are applied to a single validation trace x by computing the Gaussian probability density function pdf for both of the templates:

$$\text{pdf}(x, 0) = \frac{1}{\sqrt{(2 \cdot \pi)^l \cdot |C_0|}} \cdot \exp(-\frac{1}{2} \cdot (x - \overline{m_0})' \cdot C_0^{-1} \cdot (x - \overline{m_0}))$$

$$\text{pdf}(x, 1) = \frac{1}{\sqrt{(2 \cdot \pi)^l \cdot |C_1|}} \cdot \exp(-\frac{1}{2} \cdot (x - \overline{m_1})' \cdot C_1^{-1} \cdot (x - \overline{m_1}))$$

A binary classifier is then built by assigning each validation trace to the group with the higher likelihood. The results are depicted in Figure 27. Please note that we have used as many validation traces as for the DL-LA results presented in the respective case studies in Section 4. Clearly, the template method succeeds in detecting leakage in the data set associated to case study 1 and produces a very high confidence to reject the null hypothesis. However, as apparent in Figure 27(a) and Figure 27(b), the data complexity to overcome the detection threshold is very large compared to both, the univariate tests and the DL-LA results (13 000 training traces with a fixed validation set of 10 000 traces vs less than 10 training traces using the same validation set[12]). The reason for this is simply that the multivariate Gaussian templates span over all time samples in the traces without explicitly giving more weight to certain areas of the trace which could allow

---

[12] Please note that the results in the Appendix A show that less than 10 training traces are also sufficient when using a validation set of 500 traces for DL-LA on the data sets corresponding to case study 1.

straightforward classification. Since many time samples include more noise than useful information, this prevents successful classification with templates that are built from a small amount of training traces. Only after enough traces are considered in the training set to average out a sufficient amount of noise, the templates actually become useful. Of course, this disadvantage of the template method can be circumvented in multiple ways. One example is to first select a number of points of interest in the trace and to only build and match the templates based on these points. However, in this work we are explicitly interested in methods that make any kind of pre-selection or pre-processing, especially any manual effort, unnecessary. While DL-LA fulfills this criterion and is able to succeed with a small data complexity on the raw traces without any pre-selection or pre-processing, the template approach requires a significantly higher data complexity than the traditional methods as well as DL-LA under the same conditions.
The template analysis performs even worse on the data set associated to case study 4. While univariate tests and DL-LA succeed with less than 1 500 000 traces, the template method fails to produce a classifier that performs better than randomly guessing even with a training set of 30 000 000 traces (evaluated on a validation set of 1 500 000 traces). This result is not unexpected since multivariate Gaussian models typically extract only the mean vector and covariance matrix from the leakage traces. Clearly, this is insufficient to properly capture higher-order leakages. Again, a pre-processing of the traces (e.g., mean-free square) and pre-selection of certain points of interest may enable the detection. However, as discussed before, such additional steps depending on the type of leakage to be expected are supposed to be unnecessary when using DL-LA. Given that the template method is unable to detect univariate higher-order leakage, even with a large amount of available traces, there is no reason to believe that the method could succeed when faced with multivariate higher-order leakages. In such cases, the required effort to pre-process the traces is even larger, as it requires in-depth knowledge about the device to combine the correct samples in a trace by a combination function like the mean-free product. Therefore, we conclude that templates are no suitable candidate for replacing the neural networks used to build classifiers in DL-LA. For first-order horizontal leakages the technique may have some value, but apart from that we do not believe that the approach can contend with DL-LA in terms of flexibility and data complexity. From an efficiency standpoint the template method is not beneficial either, rather the contrary. While the template creation took less than 10 minutes for the data set corresponding to case study 1, it took almost 9 full days (210 hours and 53 minutes) of computation using 2 × Intel Xeon E5-2650 v3 CPUs with 40 combined threads to build the templates for case study 4 on 30 000 000 training traces with 5 000 time samples each (compare to Table 4). In general, the complexity of building and inverting the covariance matrix grows *at least* quadratically with the number of points in the traces, making this approach even less suitable for measurements with significantly longer traces (e.g., case study 9).

Availability and Reproducibility. Sample implementations of DL-LA based on Keras and TensorFlow using both proposed network architectures, including sensitivity analysis and a multi-precision calculator of the log probabilities are freely available at GitHub (https://github.com/Chair-for-Security-Engineering/DL-LA). For reproducibility of (a part of) the experimental results presented in this work we have hosted the underlying leakage traces for two of our nine case studies, namely CS3 and CS5, publicly online. The download links can be found in the above-given GitHub repository. Due to the large number of side-channel measurements required for the analysis in many of our case studies it is not possible to host all trace files online. CS3 and CS5 were chosen as sample data sets since they allow interesting and non-trivial analyses, but are still moderate in size (and computational complexity). For access to further data sets or the underlying software or hardware feel free to contact the authors.

6 Conclusion

We introduced Deep Learning Leakage Assessment (DL-LA), the first methodology to perform side-channel leakage detection by training a classifier based on deep neural networks. We detail all steps that are required to perform such an analysis on a target device or measurement set and develop a metric that allows to compare its results to conventional leakage detection approaches like the t-test and χ^2 test. We propose and evaluate two different network structures that deliver universal performance across nine different case studies based on real-world power traces measured on three different implementation platforms, FPGA, ASIC and µC. Our experimental analysis and the extensive comparison to traditional leakage detection methods demonstrate that DL-LA is capable of detecting side-channel leakage in smaller data sets than the competition and results in confidence values that are orders of magnitude higher than what traditional methods deliver.

In the case of multivariate leakage DL-LA effortlessly learns an accurate classifier, while multivariate extensions of the t- and χ^2-test require (i) exhaustive search over all time offsets or (ii) expert-level domain knowledge to choose the correct offset. Most importantly, we demonstrate a case study in which the classical hypothesis tests cannot detect any leakage despite having white-box knowledge about the underlying implementation while DL-LA indicates the insecurity with overwhelming confidence in a black box setting, requiring only a part of the available traces.

Our method unifies horizontal and vertical side-channel evaluation, is simple to use, broadly applicable and produces results with high statistical confidence. We believe that it can be a valuable addition to the evaluator's toolbox (as a complement to the t-test and χ^2-test) to severely reduce false negatives in multivariate and horizontal settings.

Acknowledgments

The work described in this paper has been supported in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and through the project 271752544 "NaSCA: Nano-Scale Side-Channel Analysis".

References

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

[CDP17] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Convolutional neural networks with data augmentation against jitter-based countermeasures - profiling attacks without pre-processing. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 45–68. Springer, 2017.

[CRR02] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic*

Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.

[DS16] François Durvaux and François-Xavier Standaert. From improved leakage detection to the detection of points of interests in leakage traces. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 240–262. Springer, 2016.

[GJJR11] G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi. A testing methodology for side channel resistance validation. In *NIST non-invasive attack testing workshop*, 2011.

[HGM+11] Gabriel Hospodar, Benedikt Gierlichs, Elke De Mulder, Ingrid Verbauwhede, and Joos Vandewalle. Machine learning in side-channel analysis: a first study. *J. Cryptographic Engineering*, 1(4):293–302, 2011.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[MDP19a] Loïc Masure, Cécile Dumas, and Emmanuel Prouff. Gradient visualization for general characterization in profiling attacks. In Ilia Polian and Marc Stöttinger, editors, *Constructive Side-Channel Analysis and Secure Design - 10th International Workshop, COSADE 2019, Darmstadt, Germany, April 3-5, 2019, Proceedings*, volume 11421 of *Lecture Notes in Computer Science*, pages 145–167. Springer, 2019.

[MDP19b] Loïc Masure, Cécile Dumas, and Emmanuel Prouff. A comprehensive study of deep learning for side-channel analysis. Cryptology ePrint Archive, Report 2019/439, 2019.

[mea20] mikkokotila et al. Autonomio talos: Hyperparameter optimization for keras. Source Code Repository, 2020. `http://github.com/autonomio/talos`.

[MM13] Amir Moradi and Oliver Mischke. On the simplicity of converting leakages from multivariate to univariate - (case study of a glitch-resistant masking scheme). In Guido Bertoni and Jean-Sébastien Coron, editors, *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, volume 8086 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2013.

[Moo19] Thorben Moos. Static power SCA of sub-100 nm CMOS asics and the insecurity of masking schemes in low-noise environments. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):202–232, 2019.

[MPP16] Houssem Maghrebi, Thibault Portigliatti, and Emmanuel Prouff. Breaking cryptographic implementations using deep learning techniques. In Claude Carlet, M. Anwar Hasan, and Vishal Saraswat, editors, *Security, Privacy, and Applied Cryptography Engineering - 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings*, volume 10076 of *Lecture Notes in Computer Science*, pages 3–26. Springer, 2016.

[MRSS18] Amir Moradi, Bastian Richter, Tobias Schneider, and François-Xavier Standaert. Leakage detection with the x2-test. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):209–237, 2018.

[NRR06] Svetla Nikova, Christian Rechberger, and Vincent Rijmen. Threshold implementations against side-channel attacks and glitches. In Peng Ning, Sihan Qing, and Ninghui Li, editors, *Information and Communications Security, 8th Int. Conf., ICICS 2006, Raleigh, NC, USA, Dec, 2006, Proceedings*, volume 4307 of *Lecture Notes in Computer Science*, pages 529–545. Springer, 2006.

[PCBP21] Guilherme Perin, Lukasz Chmielewski, Lejla Batina, and Stjepan Picek. Keep it unsupervised: Horizontal attacks meet deep learning. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(1):343–372, 2021.

[PEC19] Guilherme Perin, Baris Ege, and Lukasz Chmielewski. Neural network model assessment for side-channel analysis. *IACR Cryptol. ePrint Arch.*, 2019:722, 2019.

[PHJ$^+$19] Stjepan Picek, Annelie Heuser, Alan Jovic, Shivam Bhasin, and Francesco Regazzoni. The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(1):209–237, 2019.

[PMK$^+$11] Axel Poschmann, Amir Moradi, Khoongming Khoo, Chu-Wee Lim, Huaxiong Wang, and San Ling. Side-channel resistant crypto for less than 2, 300 GE. *J. Cryptology*, 24(2):322–345, 2011.

[PSK$^+$18] Stjepan Picek, Ioannis Petros Samiotis, Jaehun Kim, Annelie Heuser, Shivam Bhasin, and Axel Legay. On the performance of convolutional neural networks for side-channel analysis. In Anupam Chattopadhyay, Chester Rebeiro, and Yuval Yarom, editors, *Security, Privacy, and Applied Cryptography Engineering - 8th International Conference, SPACE 2018, Kanpur, India, December 15-19, 2018, Proceedings*, volume 11348 of *Lecture Notes in Computer Science*, pages 157–176. Springer, 2018.

[sak] Side-channel AttacK User Reference Architecture. `http://satoh.cs.uec.ac.jp/SAKURA/index.html`.

[SBM18] Pascal Sasdrich, René Bock, and Amir Moradi. Threshold implementation in software - case study of PRESENT. In Junfeng Fan and Benedikt Gierlichs, editors, *Constructive Side-Channel Analysis and Secure Design - 9th International Workshop, COSADE 2018, Singapore, April 23-24, 2018, Proceedings*, volume 10815 of *Lecture Notes in Computer Science*, pages 227–244. Springer, 2018.

[SM15] Tobias Schneider and Amir Moradi. Leakage Assessment Methodology - A Clear Roadmap for Side-Channel Evaluations. In *CHES 2015*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.

[Sta18] François-Xavier Standaert. How (not) to use welch's t-test in side-channel security evaluations. In Begül Bilgin and Jean-Bernard Fischer, editors, *Smart Card Research and Advanced Applications, 17th International Conference, CARDIS 2018, Montpellier, France, November 12-14, 2018, Revised Selected Papers*, volume 11389 of *Lecture Notes in Computer Science*, pages 65–79. Springer, 2018.

[SZ13] Simonyan and Vedaldiand Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[Tim19] Benjamin Timon. Non-profiled deep learning-based side-channel attacks with sensitivity analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):107–131, 2019.

[WO19] Carolyn Whitnall and Elisabeth Oswald. A cautionary note regarding the usage of leakage detection tests in security evaluation. *IACR Cryptol. ePrint Arch.*, 2019:703, 2019.

A Appendix

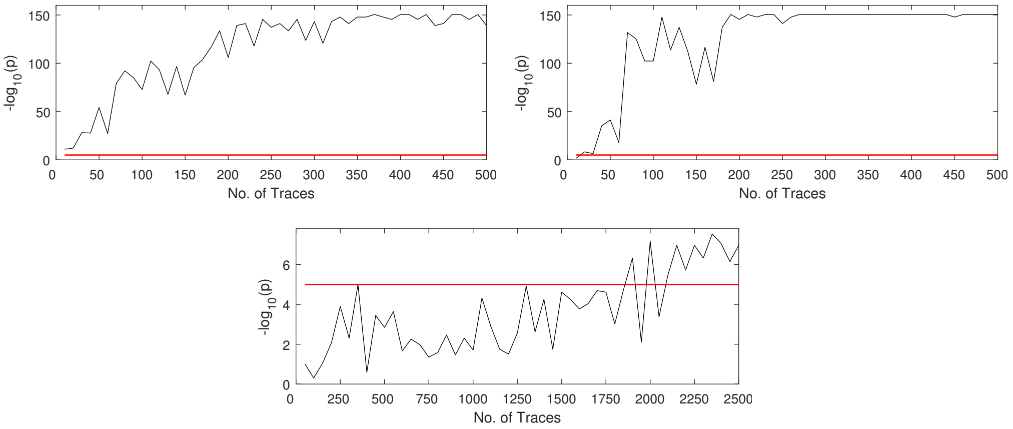


Figure 28: DL-LA results targeting an unprotected serialized PRESENT-80 implementation, using (up to) half the traces as training set and half the traces as validation set. From top to bottom: 1) aligned traces, 2) misaligned traces, 3) randomized clock. For 1) and 2) the training set ranges from 10 to 500 traces in steps of 10, while the validation set is 500 traces large. For 3) the training set ranges from 50 to 2 500 traces in steps of 50, while the validation set is 2 500 traces large.

Chapter 5

Low-Latency Block Ciphers

In this chapter we present the peer-reviewed publications accumulated in this thesis with relation to the design of low-latency block ciphers. In total, this chapter covers one paper published at the International Conference on Selected Areas in Cryptography (SAC) and another paper in the IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES).

Contents of this Chapter

5.1 PRINCEv2 295
5.2 The SPEEDY Family of Block Ciphers 327

5.1 PRINCEv2

Publication Data

> Dusan Bozilov, Maria Eichlseder, Miroslav Knezevic, Baptiste Lambin, Gregor Leander, Thorben Moos, Ventzislav Nikov, Shahram Rasoolzadeh, Yosuke Todo, and Friedrich Wiemer. Princev2 - more security for (almost) no overhead. In Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn, editors, *Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers*, volume 12804 of *Lecture Notes in Computer Science*, pages 483–511. Springer, 2020

The acceptance rate at the International Conference on Selected Areas in Cryptography (SAC) 2020 was **51,9%** [Accb].

Content This work proposes the `PRINCEv2` low-latency block cipher which is a re-design of the `PRINCE` block cipher. `PRINCEv2` benefits from an improved key schedule and a slight change to the middle round in order to improve the security of the design without sacrificing performance and without changing the number of rounds or the round operations. The comparison of the resource consumption and performance of `PRINCEv2` to other low-latency block ciphers shows that the new design is very competitive.

Contribution The author of this thesis contributed the hardware implementations of the analyzed ciphers including `PRINCEv2` and `PRINCE+v2` together with their corresponding evaluation and the extensive resource consumption and latency comparison of low-latency block ciphers. He also contributed substantially to the writing of Section 4. The author would like to thank all co-authors for their significant contributions to the design of the cipher and its security analysis.

PRINCEv2

More Security for (Almost) No Overhead

Dušan Božilov[1,2], Maria Eichlseder[3,4], Miroslav Knežević[1], Baptiste Lambin[4],
Gregor Leander[4,5], Thorben Moos[4], Ventzislav Nikov[1],
Shahram Rasoolzadeh[4], Yosuke Todo[6,4], and Friedrich Wiemer[4,5]

[1] NXP Semiconductors, Leuven, Belgium `firstname.lastname@nxp.com`
[2] COSIC KU Leuven and imec, Leuven, Belgium `dusan.bozilov@esat.kuleuven.be`
[3] Graz University of Technology, Graz, Austria `maria.eichlseder@iaik.tugraz.at`
[4] Ruhr-Universität Bochum, Bochum, Germany `firstname.lastname@rub.de`
[5] cryptosolutions, Essen, Germany `friedrich,gregor@cryptosolutions.de`
[6] NTT Secure Platform Laboratories, Tokyo 180-8585, Japan
`yosuke.todo.xt@hco.ntt.co.jp`

Abstract. In this work, we propose tweaks to the `PRINCE` block cipher that help us to increase its security without changing the number of rounds or round operations. We get substantially higher security for the same complexity. From an implementation perspective, `PRINCEv2` comes at an extremely low overhead compared to `PRINCE` in all key categories, such as area, latency and energy. We expect, as it is already the case for `PRINCE`, that the new cipher `PRINCEv2` will be deployed in various settings.

Keywords: `PRINCE` · low latency · lightweight · block cipher

1 Introduction

During the last several years we have been witnessing a very rapid deployment of secure microcontrollers in IoT, automotive and cloud infrastructures. Various technology fields, including industrial automation, robotics as well as the 5th generation mobile network urge for real-time operation and require low-latency execution while preserving the highest levels of security. Low-power and low-energy requirements are of equal importance, especially when considering the IoT market where the majority of devices are in a low-power mode during most of their lifetime. Those devices occasionally wake up, carry out a quick computation, the result of which they store or communicate securely and then go back to sleep. A life most humans are craving to have.

Securing a microcontroller involves securing the following three groups of assets:

- End-user data, including dynamic data such as personal information.
- Original equipment manufacturer (OEM) intellectual property, including firmware and software.

The article is the revised version of the final version [BEK+20] submitted by the authors, available online at
`https://sac2020.ca/files/preproceedings/08-PrinceV2.pdf`. The confcerence presentation is available at `https://www.youtube.com/watch?v=YBxb1oWb8OQ`.

– Silicon manufacturer intellectual property, including hardware, ROM code, crypto libraries and various drivers.

Securing each of these assets is a complex problem and requires a multi-layered approach. Starting with trust provisioning and then enabling secure boot, secure debug access, secure firmware update, secure test and life-cycle management is essential for creating a secure execution environment. One of the crucial aspects of a secure execution environment is secure storage, a security mechanism used for keeping confidentiality and integrity of valuable code and data stored on the device. Our paper focuses on memory encryption, which is a fundamental building block of secure storage.

Memory encryption has been used in the PC world for a long time already. Some examples include IBM's SecureBlue++, Intel's SGX and AMD's SEV encryption and integrity mechanisms. Those mechanisms are used to protect valuable assets against an attacker capable of monitoring and/or manipulating the memory content. The attacker is assumed to be employing various software tools running on the targeted platform as well as being able to manipulate the integrity of any underlying hardware by using invasive methods such as probing, voltage glitching, electromagnetic fault injection (EMFI), etc. While the device must remain secure in the hands of such adversary, the level of protection differs depending on whether the memory is volatile or non-volatile or whether it is internal or external to the system on chip (SoC). Securing the SoC-external memory is especially challenging while, at the same time, the advances of CMOS technology lead to increased production of FLASH-less microcontrollers. Add to this the everlasting requirement to minimize power and energy consumption as well as the never-ending race for higher performance and it will become clear why designing an efficient memory encryption scheme is a serious challenge. We are looking for a solution where a single clock cycle encryption is one cycle too many.

`PRINCE` [BCG$^+$12] is the first publicly known low-latency family of block ciphers that got scrutinized by the cryptographic community.[7] As a result, `PRINCE` has been deployed in a number of products including LPC55S of NXP Semiconductors [NXP20], which is a family of highly constrained general purpose IoT microcontrollers.

As pointed out in [KNR12,BCG$^+$12], the ultimate goal of low-latency block cipher design is to encrypt a block of data in a single clock cycle. The best illustration of the importance of meeting this goal is to look at the comparison of `PRINCE` and `AES` in the low-latency setting. When implemented fully unrolled, `PRINCE` occupies 4 times less silicon area while, at the same time, reaching an 8 times higher clock frequency.

Although some design principles have been explored during the design of `PRINCE`, there has been little work going on to determine the design choices that lead to the lowest-latency and most energy-efficient cipher architecture. Several parameters contribute to the efficiency of a given cipher design: area, latency, throughput, power, and energy. Several other designs, including `Midori` [BBI$^+$15],

---

[7] see `https://www.emsec.ruhr-uni-bochum.de/research/research_startseite/prince-challenge/`

`MANTIS` [BJK+16] and `QARMA` [Ava17] have been particularly optimized for one or more of these parameters.

This document describes the result of our efforts to increase the security margins of `PRINCE` without significantly increasing the latency, area, power or energy consumption. We recall the `PRINCE` security claims as follows: an adversary who has 2^n chosen plaintext-ciphertext pairs (obtained under the same key) needs at least 2^{126-n} calls to the encryption function to recover the secret key. This security level is sufficient for most of the aforementioned applications, yet we set ourselves to explore design opportunities when facing the security requirements NIST put forward in its lightweight crypto competition [NIS]. Therefore, the targeted security level for `PRINCEv2` is 112 bits [NIS18]; precisely, we claim that there is no attack against `PRINCEv2` with data complexity below 2^{47} (chosen) plaintext-ciphertext pairs (obtained under the same key) and time-complexity below 2^{112}. It has to be noted that the NIST lightweight crypto competition does not focus on the design of low-latency block ciphers. Instead, it focuses on Authenticated Encryption with Associated Data (AEAD) schemes which are, in general, too slow or too big compared to dedicated block ciphers, thus failing to meet the aforementioned design challenges.

One last design constraint we put in front of ourselves is to be able to implement `PRINCEv2` on top of the existing `PRINCE` architecture without adding a significant area overhead nor increasing the latency.

Our Contribution

Starting with the last design constraint mentioned, we tried to minimize the difference from `PRINCEv2` to `PRINCE`. Besides minimizing the overhead of implementing one on top of the other, this has the convenient benefit that a lot of the security analysis that `PRINCE` received can then either directly be transferred to `PRINCEv2` or transferred with small modifications.

To achieve the requested higher security level, a different key schedule is strictly necessary, as without a change there the generic bound of the FX construction applies. Through a carefully crafted and analyzed key schedule we can get a secure cipher meeting the NIST security requirements. Besides the change in the key schedule, we only add a single XOR in the middle rounds. This middle round was unkeyed in `PRINCE`. From an aesthetic point of view, the new key schedule has the drawback that the α-reflection property is slightly weakened. That is, decryption is not simply encryption with a modified key as in `PRINCE`, but requires slightly more effort.

Besides being beneficial from a security point of view, our minimal changes result in only minimal performance changes in all the aforementioned dimensions. This makes `PRINCEv2`, in the unrolled setting we aim at, nearly as efficient as `PRINCE`, while achieving a higher security level, echoing the title of our work.

We want to emphasize that the problem we are trying to solve is quite general, yet not an easy task. It touches the least understood part of block cipher design, namely the design of the key-scheduling. For an existing cipher with a potentially non-optimal key-scheduling (here `PRINCE`), this translates to the question on how to increase security while minimizing the resulting overhead. For the interesting – being deployed in several products – case of `PRINCE`, we came up with an elegant yet simple and efficient solution to this problem. This simplicity is an advantage concerning our objective. We venture to say that the smaller the change to the original `PRINCE` design, the higher the value of our contribution.

Outline of the Paper

We specify the design of `PRINCEv2` in Section 2, highlighting the differences to `PRINCE` and explaining our choices in Section 3. As noted above, the main changes are in the key schedule. By not using the FX construction for this new design, we also follow the advice in [Din15].

In Section 4 we report on our findings when implementing `PRINCEv2` and compare with `PRINCE`, `Midori`, `MANTIS` and `QARMA`. As we will explain there, those comparisons are naturally difficult as, for example, `MANTIS` and `QARMA` provide a tweak, which `PRINCEv2` does not.

We discuss the security analysis in Section 5. As `PRINCE` has attracted quite some third party analysis, e.g. [CFG$^+$14a,DZLY17,Mor17,RR16b], we can build on significant previous work. Besides confirming our belief that `PRINCEv2` indeed provides the requested security level, as a side result, we derive some new insights in `PRINCE` as well.

2 Specification

As discussed above, we aim to keep the changes to `PRINCE` minimal. To achieve this, we use the same round function and only change the middle layer, key schedule and the round constants compared to `PRINCE`. To be self-contained, we quickly recall `PRINCE`'s general structure and the round function, before giving the updated parts for `PRINCEv2`.

2.1 `PRINCE`

`PRINCE` is a family of block ciphers with block size of 64 and key size of 128 bits. The encryption function iterates the round function R five times, then applies the middle layer R', followed by five applications of the inverse round function R^{-1}. The round function itself applies an S-box layer `SB`, followed by a linear layer consisting of a MixColumns operation `MC` and a ShiftRows `SR`. The S-box in `PRINCE` family, can be chosen from one of the 8 Affine equivalent classes given in the proposal paper [BCG$^+$12, Tab. 3]. The S-box used in the `PRINCE` proposal that is given in Tab. 1(a), the ShiftRows permutation applied in `SR` in Tab. 1(b). While the ShiftRows permutation is the same used in the AES, the MixColumns operation is built from the following four 4×4 matrices:

$$M_1 = \begin{pmatrix} 0&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 1&0&0&0 \\ 0&0&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix}, \quad M_3 = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&0 \\ 0&0&0&1 \end{pmatrix}, \quad M_4 = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&0 \end{pmatrix},$$

Table 1. The S-box and ShiftRows permutation used in `PRINCE` and `PRINCEv2`. Note that the S-box for `PRINCE` family can be chosen from 8 Affine equivalent classes and the one given here is the one suggested in the `PRINCE` proposal paper.

(a) 4-bit S-box of `SB`

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	B	F	3	2	A	C	9	1	6	7	8	0	E	5	D	4

(b) Permutation of `SR`

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	5	10	15	4	9	14	3	8	13	2	7	12	1	6	11

or in other words M_i is the 4×4 identity matrix where the ith row is replaced by the zero vector. From these M_i we build the matrices $\widehat{M}^{(0)}$, $\widehat{M}^{(1)}$ and M':

$$\widehat{M}^{(0)} = \begin{pmatrix} M_1 & M_2 & M_3 & M_4 \\ M_2 & M_3 & M_4 & M_1 \\ M_3 & M_4 & M_1 & M_2 \\ M_4 & M_1 & M_2 & M_3 \end{pmatrix}, \ \widehat{M}^{(1)} = \begin{pmatrix} M_2 & M_3 & M_4 & M_1 \\ M_3 & M_4 & M_1 & M_2 \\ M_4 & M_1 & M_2 & M_3 \\ M_1 & M_2 & M_3 & M_4 \end{pmatrix}, \ M' = \begin{pmatrix} \widehat{M}^{(0)} & 0 & 0 & 0 \\ 0 & \widehat{M}^{(1)} & 0 & 0 \\ 0 & 0 & \widehat{M}^{(1)} & 0 \\ 0 & 0 & 0 & \widehat{M}^{(0)} \end{pmatrix},$$

i.e., M' is the 64×64 block diagonal matrix with blocks $(\widehat{M}^{(0)}, \widehat{M}^{(1)}, \widehat{M}^{(1)}, \widehat{M}^{(0)})$. Finally, the `MC`-layer multiplies the state with M' and is an involution.

The round function application is interleaved with additions of the round key $\oplus_{k_i}$ and constant $\oplus_{RC_i}$, where

$$\oplus_{k_i}(x) := x + k_i \qquad \text{and} \qquad \oplus_{RC_i}(x) = x + RC_i \ .$$

Overall for `PRINCE` we thus have the structure shown in Fig. 1 (top), where

$$R = \mathtt{SR} \circ \mathtt{MC} \circ \mathtt{SB} \ , \quad R'_{\mathtt{PRINCE}} = \mathtt{SB}^{-1} \circ \mathtt{MC} \circ \mathtt{SB} \quad \text{and} \quad R^{-1} = \mathtt{SB}^{-1} \circ \mathtt{MC} \circ \mathtt{SR}^{-1} \ .$$

As we modify the `PRINCE` key schedule and round constants, we do not give more details about them.

2.2 `PRINCEv2`

`PRINCEv2` is also a family of block ciphers in the same way as `PRINCE` that the applied S-box can be chosen from 8 Affine equivalent classes. For `PRINCEv2`, we keep the forward round R and backward round R^{-1} with their operations `SB`, `MC`, and `SR`. The structure for one full encryption is shown in Fig. 1 (bottom) and in full detail in Figs. 9 and 10 in the appendix.

Middle Layer We change R', which was a key-less operation, to

$$R' = \mathtt{SB}^{-1} \circ \oplus_{RC_{11}+k_1} \circ \mathtt{MC} \circ \oplus_{k_0} \circ \mathtt{SB} \ .$$

Key Schedule Given the 128-bit master key $k = (k_0 \parallel k_1)$, we define the ith round key as

$$k_i := \begin{cases} k_0 & i \in \{0, 2, 4, 6, 8, 10\} \\ k_1 & i \in \{1, 3, 5, 7, 9, 11\} \end{cases} ,$$

that is, we alternate between the two parts of the master key k_0 and k_1.

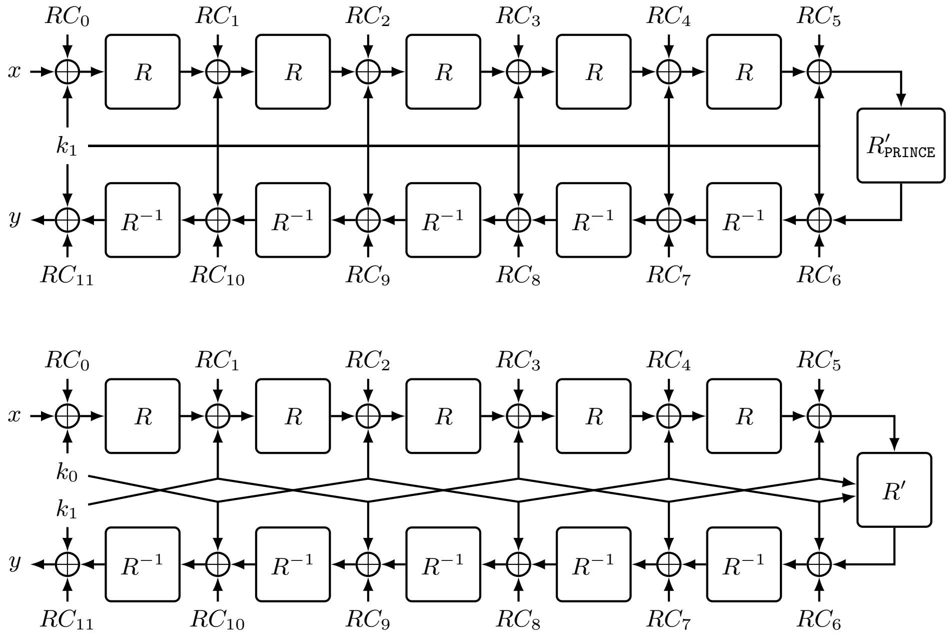


Fig. 1. (Top) `PRINCE` core structure, leaving out the FX construction; (bottom) `PRINCEv2` structure. Note that values of RC_7, RC_9 and RC_{11} in `PRINCEv2` are different than the ones in `PRINCE`.

Round Constants The round constants are derived as for `PRINCE`, but instead of adding the same α for every round constant in the second half of the encryption process ($i \geqslant 6$), we alternate adding α and β as defined in Tab. 2.

2.3 Encryption vs. Decryption

While `PRINCEv2` does not fulfil the α reflection property anymore, the choice of round keys and constants allows to implement both encryption and decryption with only a small area and delay overhead. This is shown in Fig. 10. In this figure the extra control signal **dec** switches between encryption and decryption. In particular, the `Swap` function is defined as

$$\texttt{Swap}(k_0, k_1, \text{dec}) = \begin{cases} k_0, k_1 & \text{if dec} = 0 \\ k_1 \oplus \beta, k_0 \oplus \alpha & \text{if dec} = 1 \end{cases} .$$

3 Design Rationale

The main objectives almost immediately results in clear design rationales. The first design rationale is to leave the round function, and not less important the

Table 2. Round constants used in `PRINCEv2`.

Constants	
$RC_0 =$ `0000000000000000`	$RC_6 =$ `7ef84f78fd955cb1` $= RC_5 \oplus \alpha$
$RC_1 =$ `13198a2e03707344`	$RC_7 =$ `7aacf4538d971a60` $= RC_4 \oplus \beta$
$RC_2 =$ `a4093822299f31d0`	$RC_8 =$ `c882d32f25323c54` $= RC_3 \oplus \alpha$
$RC_3 =$ `082efa98ec4e6c89`	$RC_9 =$ `9b8ded979cd838c7` $= RC_2 \oplus \beta$
$RC_4 =$ `452821e638d01377`	$RC_{10} =$ `d3b5a399ca0c2399` $= RC_1 \oplus \alpha$
$RC_5 =$ `be5466cf34e90c6c`	$RC_{11} =$ `3f84d5b5b5470917` $= RC_0 \oplus \beta$
$\alpha =$ `c0ac29b7c97c50dd`	$\beta =$ `3f84d5b5b5470917`

number of rounds, (almost) unchanged, and only change the key-scheduling. Note that the main reason for wanting to keep the number of round unchanged is mainly because having more rounds will necessarily increase latency and cause an overall loss in performances. As we are able to improve the security margin as in `PRINCE` (see Section 5), we believe that this is the correct choice to make. Here, the round-constants are thought of as part of the key-scheduling, even if it is not presented this way in the original `PRINCE` paper. This rationale (leaving the round function as is) both drastically simplified the design process and made it much more challenging. The simplification is due to the choices being narrowed down to a key-scheduling that has to be picked. More challenging, the analysis has to be much more precise and careful, as clearly the security margin would decrease. It is important to highlight that the security margin is relative to the claimed security level, not in absolute terms.

The only compromise from the goal of leaving the round function unchanged is the middle round. Some attacks that have been developed since the publication of `PRINCE` take explicit advantage of the symmetry and the key-less middle rounds. To make those attacks, particularly meet-in-the-middle attacks and accelerated exhaustive search procedures, less of a concern, it seems to be a good trade-off to spend two (actually one as will be explained in the implementation section) additional XOR on the critical path. It is noteworthy to mention that the idea of the keyed middle round is previously used in the design of `QARMA` block cipher.

For the key-scheduling, we were again highly restricted by the requirement of limiting the overhead of implementing decryption on top of encryption. This implies, as it did in `PRINCE`, that a complicated key-update is not a proper choice but a simple, up to the constants, periodic key-scheduling is best.

We opted for one of the simplest possible options: iterated round-keys.

Originally, in `PRINCE`, the round keys derived from the 128-bit key $(k_0 \parallel k_1)$ correspond to

$$k_0 \oplus k_1 \ , \ k_1 \ , \ k_1 \ , \ k_1 \ , \ k_1 \ , \ k_1 \ , \ k_1 \oplus \alpha \ , \ k_1 \oplus \alpha \ , \ k_1 \oplus \alpha \ , \ k_1 \oplus \alpha \ , \ k_1 \oplus \alpha \ , \ k_0' \oplus k_1 \oplus \alpha \ ,$$

where k_0' is the result of a simple and efficient bijective linear mapping from k_0, and α is a constant value.

In particular, the value of k_0 is used only in the whitening keys, limiting the security generically. The α-reflection property, that is the fact that decryption is encryption with a modified key, follows as replacing k_1 by $k_1 \oplus \alpha$ reverts the order of the round keys (except for the outer whitening keys).

In `PRINCEv2`, using a master key $(k_0 \parallel k_1)$, we decide to choose

$$k_0\ ,\ k_1\ ,\ k_0\ ,\ k_1\ ,\ k_0\ ,\ k_1\ ,\ k_0\ ,\ k_1{\oplus}\beta\ ,\ k_0{\oplus}\alpha\ ,\ k_1{\oplus}\beta\ ,\ k_0{\oplus}\alpha\ ,\ k_1{\oplus}\beta\ ,\ k_0{\oplus}\alpha\ ,\ k_1{\oplus}\beta\ ,$$

as the round keys where α and β are constant values. The constants are chosen as digits of $\pi = 3.1415\ldots$, as they were done in `PRINCE`. The new constant β is simply the next in line looking at the binary digits of π, see the appendix for a sage code to reproduce the constants used. Besides, it is noteworthy to mention that due to the key additions in the middle round of `PRINCEv2`, it has two more round keys than the one for `PRINCE`.

Here, replacing k_0 by $k_1 \oplus \beta$ and k_1 by $k_0 \oplus \alpha$ does ensure that the first rounds (the first seven round keys) of the encryption circuit perform decryption as required. However, when reaching the middle round (second key addition of the middle round), an additional modification is required to ensure the second half (the second seven round keys) works as well. For the second half of the round keys, we need to XOR all of these round keys with the constant value $\alpha \oplus \beta$. While replacing k_0 by $k_1 \oplus \beta$ and k_1 by $k_0 \oplus \alpha$, needs to implement 64 multiplexers in the critical path, modifying round keys of the second half does not affect the latency. As we will show in Section 4, combining decryption together with the encryption circuit does not significantly harm performance.

The only case where this would not be necessary is when α equals β. However, this would introduce a set of weak-keys. Namely, if $k_1 \oplus k_2 = \alpha$, then encryption would equal decryption, that is, the whole cipher would be an involution.

Finally, let us explicitly state the claim we want our design to be tested against:

Security Claim: We claim that there is no attack against `PRINCEv2` with data complexity below 2^{50} bytes – 2^{47} (chosen) plaintext-ciphertext pairs obtained under the same key – and time-complexity below 2^{112}. We do not claim any security in the related-key setting and related-keys have to be avoided at the protocol level.

This claim is backed up by the extensive security analysis. It is interesting to see how the advance in the state of the art has made the analysis more precise (e.g., for Boomerang-attacks using connectivity tables and for integral attacks using the division property) and simpler (using mainly MILP-based tools). Those improvements are an important tool to enable a cipher design with a very tight security claim: For a cipher optimized for low-latency, a large security margin is nothing but wasted performance.

Note that as `PRINCE` did not have any claim regarding security against side-channel and fault attacks, we chose to not make such claims either. Moreover to our knowledge, protecting a fully unrolled primitive against such attacks is not a well-researched area so far.

4 Implementation

The primary objective of `PRINCE` and `PRINCEv2` is to offer low-latency single-cycle encryption and decryption. This objective requires a short critical path in round-unrolled non-pipelined hardware implementations. In other words, the ciphers aim for a small logic depth in circuit representation. Furthermore, adding decryption functionality to an encryption circuit should induce minimal area and latency overhead. `PRINCE` achieves this goal in part due to the so-called α-reflection property [BCG$^+$12]. This property has been imitated by several other low-latency constructions (e.g. `MANTIS` [BJK$^+$16] and `QARMA` [Ava17]) and mandates that decryption with one key corresponds to encryption with a related key. Due to the modified key schedule, `PRINCEv2` does not fulfil the α-reflection behaviour of `PRINCE`. Yet, it implements a modified version that keeps the decryption overhead in hardware fairly small.

A secondary design goal is keeping its unrolled implementation cost-efficient, including a small hardware footprint (little occupied chip area) and a low energy consumption. In fact, the costs should be lower compared to unrolled implementations of other lightweight block ciphers. According to the original proposal, unrolled `PRINCE` with decryption and encryption capability can be clocked at frequencies up to 212.8 MHz when synthesized in NanGate 45 nm, an open-source standard cell library, and requires as little as 8260 Gate Equivalents (GE) of area [BCG$^+$12]. `PRINCEv2` aims to achieve similar performance figures while providing stronger security guarantees overall. Staying close to the initial design of `PRINCE` enables us to recycle and build upon the extensive security analysis it has already received. Furthermore, it allows us to construct circuits that can perform encryption and decryption in both the new `PRINCEv2` *and* original `PRINCE` at low overhead. This provides needed legacy support and backward compatibility for a variety of applications and environments where `PRINCE` is already deployed.

In the following, we compare our novel `PRINCEv2` design to the original `PRINCE` concerning the minimum latency and minimum area achieved by unrolled implementations. Since gate count and delay numbers depend on the particular technology used, we provide synthesis results from 4 different commercial standard cell libraries of feature sizes between 90 nm and 28 nm. This redundancy minimizes the influence of a single technology on the comparison's interpretation. All 4 standard cell libraries contain multiple classes of gates, namely a high threshold voltage (*hvt*) class, a standard threshold voltage (*std*) class and a low threshold voltage (*lvt*) class. These distinct classes allow to fully explore the latency-vs-leakage tradeoff. More specifically, when placing a tight constraint on the latency of a circuit, primarily *lvt* cells are chosen due to their high speed. On the other hand, when synthesizing without tight timing constraints, *hvt* cells will be chosen due to the lower energy loss through leakage currents. Using manufacturable standard cell libraries from a commercial foundry instead of open-source libraries for a design comparison is often preferable since the reported numbers are more accurate in all key categories, such as area, latency and energy. They are especially superior in power and energy estimation, as common open-source libraries fail to provide industry quality characterization in that regard. How-

Table 3. Area, latency and energy characteristics of unrolled `PRINCE` and `PRINCEv2` when constrained for minimum latency.

Techn.	**Mode**	**Cipher**	**Area** [GE]	**Latency [ns]**	**Energy** [pJ]
90 nm LP*	ENC	`PRINCE`	16244.25	**4.101177**	1.993172
		`PRINCEv2`	17661.25	**4.047311**	2.230068
	ENC/DEC	`PRINCE`	17808.00	**4.106262**	2.213275
		`PRINCEv2`	18888.75	**4.151113**	2.424250
65 nm LP*	ENC	`PRINCE`	19877.75	**2.866749**	1.602513
		`PRINCEv2`	18798.25	**2.944367**	1.492794
	ENC/DEC	`PRINCE`	19966.00	**2.946442**	1.594025
		`PRINCEv2`	21171.25	**2.930153**	1.696559
40 nm LP*	ENC	`PRINCE`	17177.00	**2.521302**	0.617719
		`PRINCEv2`	16556.50	**2.509131**	0.592155
	ENC/DEC	`PRINCE`	17377.50	**2.541220**	0.630223
		`PRINCEv2`	17799.50	**2.583466**	0.648450
28 nm HPC**	ENC	`PRINCE`	38145.33	**1.108886**	1.258586
		`PRINCEv2`	33470.33	**1.103273**	1.108789
	ENC/DEC	`PRINCE`	35297.67	**1.119593**	1.181171
		`PRINCEv2`	38962.33	**1.148693**	1.299172

* LP = Low Power
** HPC = High Performance Computing

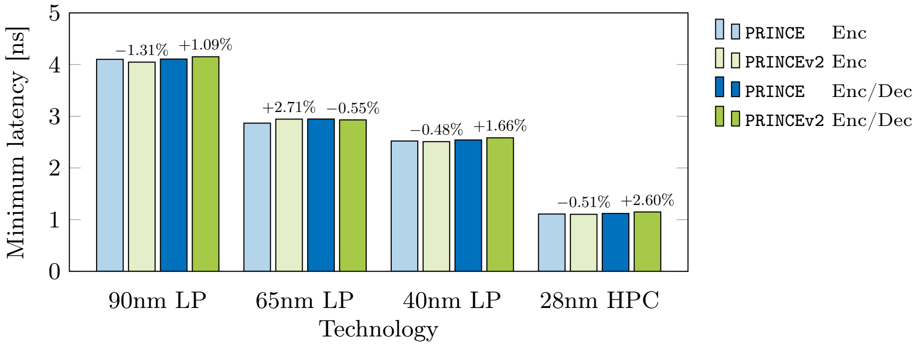
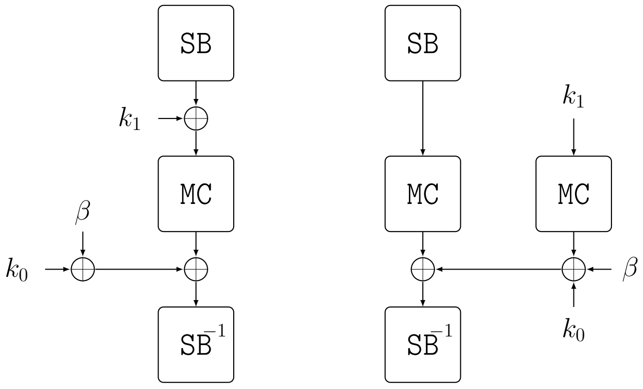


Fig. 2. Minimum achievable latency of unrolled `PRINCE` and `PRINCEv2` across different technologies.

ever, to keep our results reproducible and make comparisons to existing works easy we provide a comparison of all our unrolled `PRINCE` and `PRINCEv2` circuits to several other low-latency and low-energy constructions in NanGate 45 nm and 15 nm Open Cell Libraries (OCL) later in this section.

We consider the typical process and operating conditions in all our synthesis results, i.e., the typical PVT (process, voltage, temperature) corner case, with a nominal supply voltage and a working temperature of 25 °C. For synthesis, we



(a) regular (b) optimized

Fig. 3. Simple latency optimization strategy in the middle round of `PRINCEv2` that removes one key addition from the critical path.

have used *Synopsys Design Compiler Version O-2018.06-SP4* with three stages of the `compile_ultra` command (two incremental). As a first step, we have constrained our circuits for minimum latency. The results are given in Table 3 and visualized in Figure 2. We distinguish between circuits that can only encrypt (ENC) and circuits that can decrypt as well (ENC/DEC).

Several interesting observations can be made. Firstly, all four distinct circuits perform decidedly similar in terms of minimum latency. The difference falls in the range of single-digit picoseconds in several cases. To gain a better overview, Figure 2 provides the differences between corresponding `PRINCE` and `PRINCEv2` circuits as percentages on top of the bar graphs. Interestingly, the encryption-only version of `PRINCEv2` outperforms `PRINCE` in terms of minimum latency in three of four technologies. This may be counter-intuitive, as `PRINCEv2` adds two key additions to the middle round. However, as shown in Figure 3, those two key additions can be merged into a single one regarding the critical path by calculating and adding $\texttt{MC}(K_1)$ in parallel.

This optimization not only improves the minimum latency but also saves area.[8] One may expect the synthesis tool to perform such an optimization implicitly by itself, as two key additions and a `MC` operation in the middle essentially result in a sequence of four consecutive `XORs` per bit. Yet, our results suggest that it is indeed required to perform the optimization algorithmically in the RTL code. Additionally, it has to be noted that the original `PRINCE` design applies two key additions (whitening and round key) to the input before the first Sbox stage. `PRINCEv2`, on the other hand, applies only one. Hence, the difference between the latency of `PRINCE` and `PRINCEv2` in encryption-only mode comes down to whether the synthesizer implements the additional key `XOR` at the input more

---

[8] Area is saved by this optimization since slower cells with a lower drive strength can be selected for the parallel calculations. Those cells have a smaller area footprint.

efficiently or the one in the middle round. More often than not, the middle round key addition is more efficient latency-wise since the synthesizer has more freedom to move that `XOR` stage around (e.g., to the input, output or intermediate signals of the `MC` operation), while the two key `XORs` at the input have a fixed location and cannot be optimized beyond instantiating a three-input `XOR` per bit, since all inputs to the operation (key and plaintext) arrive at approximately the same time. Original `PRINCE` also requires an additional key `XOR` at the output. Yet, since the last round's output arrives much later than the keys, the key parts will be added to each other beforehand and only one of the `XOR` stages affects the critical path.

At this point, we should stress that differences in synthesis results of such a small magnitude may sometimes go beyond algorithmic considerations and can not always be understood in detail without having insight into the proprietary optimization algorithms used by EDA tools. Sometimes a latency optimization with a big area penalty is deemed worth it by the synthesizer, sometimes not. Thresholds for such decisions are unknown and therefore the outcome can not always be predicted. One example for such a case in Table 3 is the difference between the full variant of `PRINCEv2` and its encryption-only version in 65 nm LP technology. For unknown reasons, the full variant achieves a lower latency than the encryption-only one, but at the price of a significantly increased area, indicating costly latency optimizations. However, the majority of our reported figures directly corresponds to the algorithmic differences in the analyzed ciphers and modes.

Regarding the cipher variants with decryption capability, the situation is slightly different compared to the encryption-only versions. The more complex key-multiplexing in `PRINCEv2` required to choose between encryption and decryption, as apparent in Figure 10, induces additional delay. In the worst case, this results in an overhead of 2.6%, but on average the overhead is about 1.2%. Table 3 also reports area and energy numbers for the highly constrained circuits. The energy values correspond to the average energy consumed by one evaluation of the unrolled circuits at maximum clock frequency (corresponding to minimum latency). While `PRINCEv2` is often more area and energy-efficient than `PRINCE` in encryption-only mode, `PRINCEv2` with decryption capability consistently requires the largest area and consumes the most energy. Yet, the margins are still very thin. In summary, `PRINCE`'s most important property and main selling point, namely high-speed single-cycle encryption, is well preserved by `PRINCEv2`. For scenarios that require no decryption but only encryption, it may even be slightly improved.

As a second step, we evaluate the minimum area that can be achieved by the unrolled circuits. In this regard, we have executed the same synthesis scripts as before, but without the tight timing constraints. Our results are reported in Table 4 and depicted as a bar graph in Figure 4.

In contrast to the latency results, a consistent overhead for minimum area can be observed for the `PRINCEv2` circuits. This is expected, due to the additional operations in the middle round and the more complex key-multiplexing to decide

Table 4. Area, latency and energy characteristics of unrolled `PRINCE` and `PRINCEv2` when constrained for minimum area.

Techn.	**Mode**	**Cipher**	**Area [GE]**	**Latency** [ns]	**Energy** [pJ]
90 nm LP*	ENC	`PRINCE`	**7937.50**	12.859 908	0.569694
		`PRINCEv2`	**8111.25**	12.856 450	0.574683
	ENC/DEC	`PRINCE`	**8183.00**	14.015 245	0.616671
		`PRINCEv2`	**8440.75**	15.513 536	0.628298
65 nm LP*	ENC	`PRINCE`	**8316.00**	11.434 771	0.433378
		`PRINCEv2`	**8385.25**	11.504 968	0.430286
	ENC/DEC	`PRINCE`	**8547.75**	12.349 355	0.440872
		`PRINCEv2`	**8792.75**	13.376 949	0.456154
40 nm LP*	ENC	`PRINCE`	**8563.75**	10.144 847	0.212027
		`PRINCEv2`	**8608.50**	10.063 908	0.207317
	ENC/DEC	`PRINCE`	**8780.00**	10.886 960	0.217739
		`PRINCEv2`	**9039.75**	11.798 657	0.226534
28 nm HPC**	ENC	`PRINCE`	**8197.00**	3.599 936	0.127798
		`PRINCEv2`	**8292.00**	3.682 593	0.127786
	ENC/DEC	`PRINCE`	**8426.33**	4.260 999	0.131239
		`PRINCEv2`	**8844.67**	4.323 993	0.134909

* LP = Low Power
** HPC = High Performance Computing

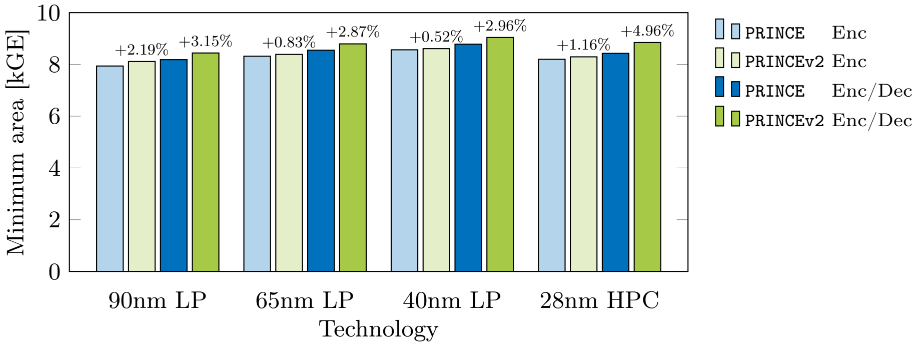


Fig. 4. Minimum achievable area of unrolled `PRINCE` and `PRINCEv2` across different technologies.

between encryption and decryption. Yet, the average overhead is less than 1.2% for the encryption-only and less than 3.5% for the full versions. This is a rather small price to pay for the additional security `PRINCEv2` provides. When comparing the low-latency and low-area implementations in Tables 3 and 4 respectively, it can be seen that area increases 2 to 4 times from low-area to low-latency constraint. Latency scales down 3 to 4 times. Energy consumption increases between 4 and 10 times, due to a dependency on both factors (caused by leakage currents). These metrics should be carefully considered when choosing operating frequency and target technology for a given application.

Finally, we compare `PRINCE` and `PRINCEv2` to other lightweight block ciphers proposed in the literature. Only a few cryptographic primitives have made low latency a primary design objective. To the best of our knowledge, none of those who have share all design goals and security claims with `PRINCE` or `PRINCEv2`. Hence, the following comparison involves ciphers with different security and performance claims and is only supposed to put their hardware efficiency in relation to each other, without concluding the superiority of one or the other. In particular, we compare `PRINCE` and `PRINCEv2` to the low-latency tweakable block ciphers `MANTIS` [BJK$^+$16] and `QARMA` [Ava17]. Yet, since both of those constructions are tweakable, unlike all `PRINCE` versions, they can not easily achieve the same latency and area as `PRINCE` and `PRINCEv2`. We also include `Midori` [BBI$^+$15] in the comparison, as the authors have partially aimed for a low logic depth as well. However, `Midori` primarily targets energy efficiency in round-based implementations and has no claim to provide low latency in unrolled representation. Additionally, we have developed a combination of `PRINCE` and `PRINCEv2`, which we call `PRINCE+v2`. This combined cipher offers a control signal to select whether the input should be processed according to the `PRINCE` or the `PRINCEv2` specification. In environments where `PRINCE` is already deployed, this can be useful for backward compatibility and legacy support. We have analyzed all 6 ciphers in two modes each (ENC and ENC/DEC) in NanGate 45 nm and 15 nm Open Cell Libraries and evaluate their area-vs-latency tradeoff. The result is depicted in Figure 5 for NanGate 45 nm and in Figure 6 for 15 nm technology. The exact performance figures used to create these graphs can be found in the Appendix in Tables 5 and 6. The results in these two libraries demonstrate that `PRINCE` and `PRINCEv2` are the most suitable choices for high-speed encryption, as long as a tweak input is not required. All `PRINCE` and `PRINCEv2` variants outperform the other ciphers both in terms of minimum latency and minimum area. `PRINCEv2` in encryption-only mode is roughly 20 percent faster than `Midori` and `MANTIS`, and 40 percent faster than `QARMA`. At the same time, its minimum area is about 15 percent smaller than `Midori`, 30 percent smaller than `MANTIS` and 40 percent smaller than `QARMA`. The results are similar when comparing both encryption and decryption implementations, except for `Midori` being significantly larger and slower. This outcome is unsurprising since all compared ciphers except `Midori` are reflection ciphers, i.e., they use a variant of the α-reflection property introduced by `PRINCE`. Please note that for this reason the full version of `Midori64`, including decryption, is barely visible in Figures 5 and 6 as it simply does not fit in the frame due to its much higher latency and area caused by the extra multiplexers required in each round. Yet the full figures for that implementation can be found in Tables 5 and 6 in the Appendix. We chose the particular instances $\texttt{MANTIS}_7$ and $\texttt{QARMA}_7$-64-σ_1 for the comparison as they are supposed to offer a similar security level as `PRINCE` and `PRINCEv2`, while being tweakable.

Table 5. Full comparison of unrolled block ciphers in NanGate 45 nm Open Cell Library.

`PRINCE`				`PRINCEv2`			
ENC		**ENC/DEC**		**ENC**		**ENC/DEC**	
Lat. [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]
4.059997	9873.33	4.119023	10486.33	4.077636	10332.00	4.245165	10780.67
4.500000	8421.67	4.500000	8807.00	4.500000	8526.67	4.500000	9488.00
5.000000	7837.00	5.000000	8213.33	5.000000	8013.00	5.000000	8659.67
5.500000	7684.33	5.500000	7959.33	5.500000	7865.67	5.500000	8328.67
6.000000	7620.00	6.000000	7874.00	6.000000	7812.33	6.000000	8196.00
6.500000	7620.00	6.500000	7868.67	6.500000	7812.33	6.500000	8144.00
7.000000	7620.00	7.000000	7868.67	7.000000	7812.33	7.000000	8141.67

`PRINCE+v2`				**Midori64**			
ENC		**ENC/DEC**		**ENC**		**ENC/DEC**	
Lat. [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]
4.353092	11258.33	4.469554	12395.67	4.934847	10755.67	7.111567	25058.33
4.500000	9588.33	4.500000	10709.00	4.500000	-	4.500000	-
5.000000	8622.67	5.000000	9870.33	5.000000	10353.67	5.000000	-
5.500000	8276.00	5.500000	9356.00	5.500000	9223.67	5.500000	-
6.000000	8169.67	6.000000	9091.67	6.000000	8858.00	6.000000	-
6.500000	8156.33	6.500000	8990.33	6.500000	8792.33	6.500000	-
7.000000	8155.33	7.000000	8982.00	7.000000	8748.33	7.000000	-
7.500000	8155.33	7.500000	8969.33	7.500000	8748.33	7.500000	19733.00
8.000000	8155.33	8.000000	8969.33	8.000000	8748.33	8.000000	18381.00
9.000000	8155.33	9.000000	8969.33	9.000000	8748.33	9.000000	16241.67
10.000000	8155.33	10.000000	8969.33	10.000000	8748.33	10.000000	14877.67
11.000000	8155.33	11.000000	8969.33	11.000000	8748.33	11.000000	14476.33

`MANTIS`$_7$				`QARMA`$_7$**-64-**σ_1			
ENC		**ENC/DEC**		**ENC**		**ENC/DEC**	
Lat. [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]
5.036228	14481.67	5.235198	14810.67	5.756122	17096.67	5.794558	18085.67
5.500000	12445.33	5.500000	12931.67	5.500000	-	5.500000	-
6.000000	11613.33	6.000000	12082.67	6.000000	14821.33	6.000000	15449.67
6.500000	11246.67	6.500000	11786.67	6.500000	13886.00	6.500000	14866.33
7.000000	11134.33	7.000000	11529.00	7.000000	13139.67	7.000000	14019.33
7.500000	11064.67	7.500000	11397.67	7.500000	12698.33	7.500000	13467.33
8.000000	11019.33	8.000000	11322.67	8.000000	12326.67	8.000000	13012.33
8.500000	11019.33	8.500000	11305.33	8.500000	12106.33	8.500000	12801.67
9.000000	11019.33	9.000000	11305.33	9.000000	12039.33	9.000000	12677.00
9.500000	11019.33	9.500000	11305.33	9.500000	12039.33	9.500000	12614.67
10.000000	11019.33	10.000000	11305.33	10.000000	12039.33	10.000000	12610.33
10.500000	11019.33	10.500000	11305.33	10.500000	12039.33	10.500000	12609.00

In the original proposal, the maximum achievable frequency in NanGate 45 nm of unrolled `PRINCE` was given as 212.8 MHz [BCG$^+$12]. Our unrolled `PRINCE` can be clocked at 242.8 MHz for the full variant and 246.3 MHz for the encryption-only version in the same technology, which corresponds to a 12.4% or 15.7% higher performance respectively. The minimum area was given as 8260 GE in the original proposal, while our implementations are as small as 7868.67 GE

Table 6. Full comparison of unrolled block ciphers in NanGate 15 nm Open Cell Library.

PRINCE				PRINCEv2			
ENC		**ENC/DEC**		**ENC**		**ENC/DEC**	
Lat. [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]
0.389144	13291.00	0.400530	13468.00	0.387146	13069.50	0.404112	14181.25
0.400000	12380.75	0.400000	-	0.400000	12331.75	0.400000	-
0.450000	9618.50	0.450000	10275.50	0.450000	9859.00	0.450000	11185.25
0.500000	8811.00	0.500000	9115.50	0.500000	8940.75	0.500000	9822.25
0.550000	8621.50	0.550000	8935.50	0.550000	8820.00	0.550000	9272.25
0.600000	8610.50	0.600000	8828.75	0.600000	8787.25	0.600000	9134.50
0.650000	8610.50	0.650000	8828.75	0.650000	8787.25	0.650000	9108.00
0.700000	8610.50	0.700000	8828.75	0.700000	8787.25	0.700000	9105.50
0.750000	8610.50	0.750000	8828.75	0.750000	8787.25	0.750000	9104.00

PRINCE+v2				**Midori64**			
ENC		**ENC/DEC**		**ENC**		**ENC/DEC**	
Lat. [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]
0.415065	14422.75	0.426661	16016.00	0.481522	13775.00	0.657338	30563.50
0.450000	11701.25	0.450000	14280.50	0.450000	-	0.450000	-
0.500000	9698.00	0.500000	11253.50	0.500000	11581.25	0.500000	-
0.550000	9234.00	0.550000	10193.00	0.550000	10427.00	0.550000	-
0.600000	9147.75	0.600000	9991.75	0.600000	9896.25	0.600000	-
0.650000	9163.00	0.650000	9967.25	0.650000	9831.00	0.650000	-
0.700000	9144.00	0.700000	9931.75	0.700000	9806.50	0.700000	28135.25
0.750000	9143.50	0.750000	9931.25	0.750000	9806.50	0.750000	22886.75
0.800000	9143.50	0.800000	9929.75	0.800000	9806.50	0.800000	20793.75
0.850000	9143.50	0.850000	9929.00	0.850000	9806.50	0.850000	18871.75
0.900000	9143.50	0.900000	9929.00	0.900000	9806.50	0.900000	17772.00
1.000000	9143.50	1.000000	9929.00	1.000000	9806.50	1.000000	16423.00
1.100000	9143.50	1.100000	9929.00	1.100000	9806.50	1.100000	15420.75
1.200000	9143.50	1.200000	9929.00	1.200000	9806.50	1.200000	15380.00
1.300000	9143.50	1.300000	9929.00	1.300000	9806.50	1.300000	15318.00

MANTIS$_7$				QARMA$_7$**-64-**σ_1			
ENC		**ENC/DEC**		**ENC**		**ENC/DEC**	
Lat. [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]	**Lat.** [ns]	**Area** [GE]
0.492660	17542.75	0.504465	18193.75	0.542777	20736.25	0.552887	22130.75
0.500000	17142.75	0.500000	-	0.500000	-	0.500000	-
0.550000	14404.00	0.550000	15159.25	0.550000	20736.25	0.550000	-
0.600000	13650.00	0.600000	14464.50	0.600000	16413.25	0.600000	18195.25
0.650000	12469.00	0.650000	12804.75	0.650000	14864.25	0.650000	16299.75
0.700000	12378.25	0.700000	12681.50	0.700000	13862.00	0.700000	15111.25
0.750000	12285.75	0.750000	12626.75	0.750000	13794.00	0.750000	14542.25
0.800000	12275.00	0.800000	12580.00	0.800000	13531.25	0.800000	14292.25
0.850000	12275.00	0.850000	12565.00	0.850000	13359.00	0.850000	14130.25
0.900000	12275.00	0.900000	12561.25	0.900000	13304.00	0.900000	14009.75
0.950000	12275.00	0.950000	12561.25	0.950000	13304.00	0.950000	13988.75

for the full variant and 7620.00 GE for the encryption-only version. That corresponds to a 5.0% or 8.4% higher area efficiency respectively. We conclude that

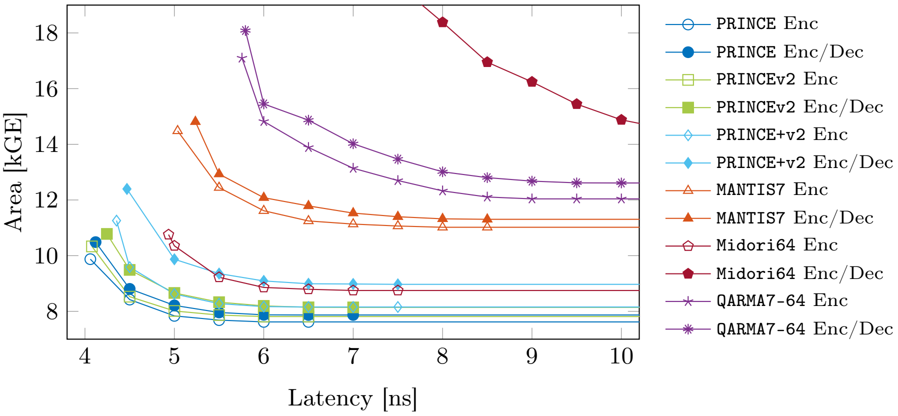


Fig. 5. Comparison of unrolled block ciphers in NanGate 45 nm Open Cell Library.

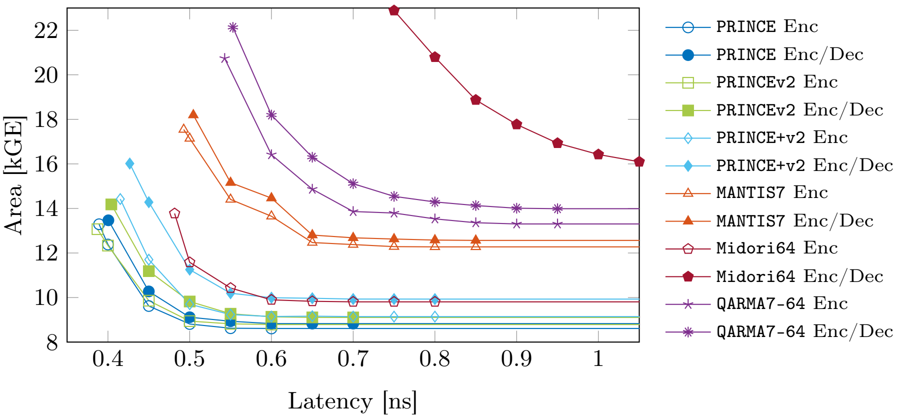


Fig. 6. Comparison of unrolled block ciphers in NanGate 15 nm Open Cell Library.

our base-implementation used for the construction of `PRINCEv2` and `PRINCE+v2` (and partially for $\texttt{MANTIS}_7$) is well optimized.

Finally, we compare the energy consumption of the 6 ciphers. As detailed before, open-source libraries are not suitable for power and energy estimation. Thus, we have performed the energy comparison in the commercial 40 nm Low Power CMOS technology. This particular technology proved to be the most energy efficient, as apparent from Tables 3 and 4. The results have been estimated at 50 MHz and are depicted in Figure 7. Please note the y-axis limits on the bar graph. The differences between the circuits are not as large as they may appear at first sight. Yet, the results confirm once again that `PRINCE` and `PRINCEv2` are the most cost-efficient unrolled circuits.

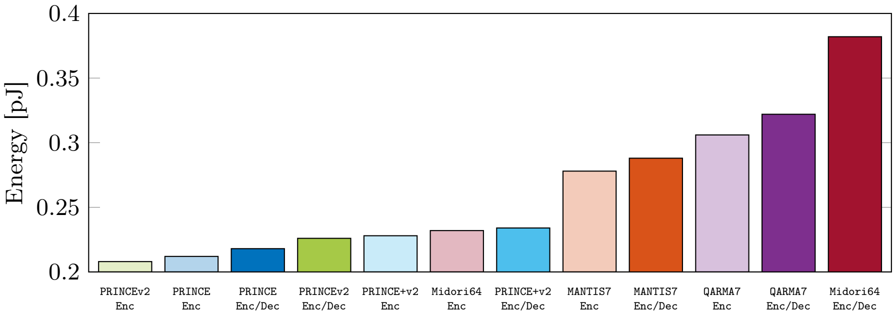


Fig. 7. Average energy consumption of unrolled block ciphers clocked at 50 MHz in a commercial 40 nm Low Power CMOS technology.

5 Security Analysis

We analyzed the security of `PRINCEv2` building on the previously published analysis of `PRINCE`. Most dedicated attacks on `PRINCEv2` are comparable to `PRINCE`, while `PRINCEv2` offers significantly better resistance to generic attacks. We show that several attacks successful against `PRINCE`, such as certain accelerated exhaustive search and meet-in-the-middle attacks, do not apply to `PRINCEv2`. Note that we do not consider analyses of variants with modified operations or related-key attacks, but we provide a discussion of the latter at the end of this section.

Since `PRINCEv2` is designed to provide a higher security level, we also need to consider attacks with higher complexity than for `PRINCE`. For several dedicated attack strategies, we find attacks that cover 1 or 2 more rounds with significantly higher time complexity T or data complexity D. This higher complexity is above the bound $D \cdot T < 2^{126}$ claimed for `PRINCE`, but below the generic bounds of $D < 2^{64}$, $T < 2^{128}$ for `PRINCEv2`, and thus relevant to judge the security margin of `PRINCEv2`. We note that the security claim for `PRINCEv2` limits the attacker to $D < 2^{47}$ and $T < 2^{112}$, while most of the results we propose for round-reduced `PRINCEv2` require more data than permitted by this bound.

Additionally, we provide several new results, including a linear attack, a 6-round integral distinguisher based on the division property, a more precise evaluation of boomerang attacks using recently published techniques, and a new 10-round Demirci-Selçuk meet-in-the-middle attack.

Table 7 provides an overview of the highlights of this section, including the best attacks on `PRINCEv2` and noteworthy new results. In summary, `PRINCEv2` is at least as secure as `PRINCE` against various dedicated attacks and provides better generic security.

Differential [ALL12,CFG$^+$*14a,CFG*$^+$*14b,DP15a,DP15b,GR16a,GR16b]:* The truncated differential used in [GR16a,GR16b] applies the subspace trail technique and attacks at most 6 rounds of `PRINCE`, which is the same for `PRINCEv2`.

Table 7. Overview of the main analysis results on round-reduced `PRINCEv2`, where 12 is the full number of rounds. Time complexity is given in computation equivalents, data complexity in known plaintexts (KP) or chosen plaintexts (CP). For most attacks in this table except meet-in-the-middle (†), the attacks on `PRINCE` apply to `PRINCEv2` with similar complexity and vice-versa; however, the complexity of the attacks listed for `PRINCEv2` is higher than permitted by the `PRINCE` security claim. For other results, refer to details in Section 5.

Attack	Target		Complexity		Reference
	Version	Rounds	Time	Data	
Differential	`PRINCE`	10	2^{61}	2^{58} CP	[CFG$^+$14a]
	`PRINCEv2`	11	2^{92}	2^{59} CP	New
Impossible differential	`PRINCE`	7	2^{54}	2^{56} CP	[DZLY17]
	`PRINCEv2`	9	$e^{-\alpha} \cdot 2^{128}$	$\alpha \cdot 2^{65}$ CP	New
Integral	`PRINCE`	7	2^{57}	2^{60} CP	[Mor17]
	`PRINCEv2`	8	$2^{107.4}$	2^{36} CP	New
Meet-in-the-middle	`PRINCE` †	10	2^{122}	2 KP	[RR16b]
	`PRINCEv2` †	10	2^{112}	2^{48} CP	New

The inside-out differential in [ALL12] took advantage of the key-less middle round to attack at most 6 rounds of `PRINCE` and is thus not applicable to `PRINCEv2`.

The most powerful differential attack against `PRINCE` was the one introduced in [CFG$^+$14a,CFG$^+$14b] that covers 10 rounds using $2^{57.94}$ chosen plaintexts, $2^{60.62}$ computations and $2^{61.52}$ blocks of memory. The distinguisher of this attack covers 6-round and it appends 2 rounds before and 2 rounds after which needs to guess 66 key bits. Using the same distinguisher and attack for `PRINCEv2`, we need to guess 64 key bits. The complexities of both attacks are roughly the same. The probability of the corresponding differentials are summarized in Table 8.

This attack can be extended by one round with the new key schedule at the cost of significantly higher time complexity, as illustrated in Figure 8.

1. Query N_s structures of 2^{32} chosen plaintexts P with columns P_1, P_3 fixed within each structure. This yields $N_s \cdot 2^{31} \cdot (2^{32} - 1) \approx N_s \cdot 2^{63}$ candidate pairs as in Figure 8.
2. For each of the $N_s \cdot 2^{63}$ candidate pairs (P, P'), we expect 1 candidate for the 96-bit key $(K_1, K_{0,0}, K_{0,2})$, which can be determined by a small number of table lookups:
   (a) We expect 1 key candidate for the key columns $K_{1,0}, K_{1,2}$ $(\star)$ that satisfies the pattern through `SB`, `MC` in rounds 1 and 11. This costs one lookup in a precomputed table $(\Delta X_0, \Delta Y_0, X_0 \oplus Y_0) \rightarrow X_0$ per pair.
   (b) For any fixed difference ΔU, we also get 1 key candidate for the 4 nibbles of K_0 involved in round 2, which determines 4 nibbles of W in round 10 $(\bullet)$.

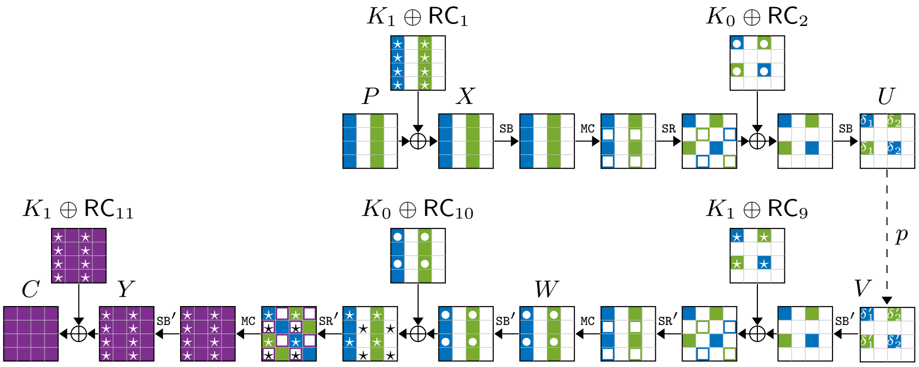

$K_1 \oplus \mathsf{RC}_1$
$K_0 \oplus \mathsf{RC}_2$
P
X
U
SB
MC
SR
SB
δ_1
δ_2
p
$K_1 \oplus \mathsf{RC}_{11}$
$K_0 \oplus \mathsf{RC}_{10}$
$K_1 \oplus \mathsf{RC}_9$
C
Y
W
V
SB′
MC
SR′
SB′
MC
SR′
SB′
δ'_1
δ'_2

Fig. 8. Differential attack on 11-round `PRINCEv2` extending [CFG$^+$14a] by one round.

(c) Due to the pattern for `MC` in round 10, there are only $2^8 \times 2^8$ possible differences ΔW. The key bits found so far determine 16 bits of this difference and thus on average fully determine ΔW.

(d) For any fixed difference ΔV, knowing $K_{1,0}, K_{1,2}$, we get on average 1 candidate for the value of the two active columns W_0, W_2. This determines the difference ΔZ and thus the key columns $K_{1,1}, K_{1,3}$ as well as the rest of $K_{0,0}, K_{0,2}$.

3. Rank the obtained $N_s \cdot 2^{63}$ candidates for the 96-bit key $(K_1, K_{0,0}, K_{0,2})$.

Using $N_s = 5/p/2^{31} = 2^{29}$ structures for the best differential $(\delta_0, \delta_1, \delta'_0, \delta'_1) = (\mathtt{1}, \mathtt{2}, \mathtt{1}, \mathtt{2})$ from Table 8, we expect about 5 right pairs, which should be easily sufficient for distinguishing. The remaining 32 key bits $(K_{0,1}, K_{0,3})$ can be recovered by brute-force search. The overall complexity is $N_s \cdot 2^{32} = 2^{61}$ chosen plaintexts and the time corresponding to $N_s \cdot 2^{63} = 2^{92}$ repetitions of a few table lookups and arithmetic operations, which can be roughly approximated by one encryption equivalent.

The attack can be slightly improved using multiple differentials from Table 8, but fewer structures. For example, we can use the 2^2 permutations of the best differential with the same p and decrease N_s by a factor of 2^2, obtaining an attack with the same expected number of valid pairs and key candidates, while the data complexity is reduced to 2^{59} and the time complexity is slightly lower than before.

Linear: Even though there is no published linear analysis for `PRINCE`, we find out that the activity patterns used in [CFG$^+$14a] for differential analysis are also useful for the linear one. This is because of the `MC` operation that uses an involutive and self-transpose matrix, i.e. $M'^T = M'^{-1} = M'$. We compute the average square correlation of all 6-round linear hulls which follow the given activity patterns similarly as in [CFG$^+$14a] for both `PRINCE` and `PRINCEv2`. These values are summarized in Table 8.

Table 8. Differentials and linear hulls fitting to the 6-round activity patterns given in [CFG+14a] for both `PRINCE` and `PRINCEv2`.

differential probability (divided by $2^{-72} \times 2^3$)			
(δ_0, δ_1)	(δ'_0, δ'_1)	`PRINCE`	`PRINCEv2`
(1,2)	(1,2)	6144	2560
(1,2)	(1,8)	3328	1344
(1,2)	(1,a)	1664	672
(1,2)	(4,2)	1536	640
(1,2)	(4,8)	1664	928
(1,2)	(4,a)	832	336
(1,8)	(1,8)	2112	784
(1,8)	(1,a)	1056	392
(1,8)	(4,2)	832	336
(1,8)	(4,8)	1056	520
(1,8)	(4,a)	528	196
(1,a)	(1,a)	528	196
(1,a)	(4,2)	416	168
(1,a)	(4,8)	528	260
(1,a)	(4,a)	264	98
(4,2)	(4,2)	384	160
(4,2)	(4,8)	416	232
(4,2)	(4,a)	208	84
(4,8)	(4,8)	656	388
(4,8)	(4,a)	264	130
(4,a)	(4,a)	132	49

average square linear correlation (divided by $2^{-91} \times 2^3$)			
(δ_0, δ_1)	(δ'_0, δ'_1)	`PRINCE`	`PRINCEv2`
(4,2)	(4,2)	3563313280	2701826048
(4,2)	(4,8)	243931552	177559040
(4,2)	(4,a)	215632256	165965824
(4,2)	(5,2)	44716840	22956032
(4,2)	(5,8)	23525008	14792960
(4,2)	(5,a)	3662080	2491520
(4,8)	(4,8)	16864144	11669888
(4,8)	(4,a)	14706248	10906624
(4,8)	(5,2)	3338620	1510400
(4,8)	(5,8)	1723948	972992
(4,8)	(5,a)	256192	163616
(4,a)	(4,a)	13067272	10194944
(4,a)	(5,2)	2613530	1409536
(4,a)	(5,8)	1385768	908416
(4,a)	(5,a)	219776	153088
(5,2)	(5,2)	1221068	203976
(5,2)	(5,8)	698555	133008
(5,2)	(5,a)	54472	20960
(5,8)	(5,8)	466562	87600
(5,8)	(5,a)	27160	13544
(5,a)	(5,a)	4024	2312

One can use these linear hulls to analyze 10-round `PRINCE` by guessing 66 key bits and on `PRINCEv2` with 64 key bits guesses (similar to differential attack). Data complexity for both of these attacks are factor of 2^{57} known plaintexts.

Impossible Differential [DZLY17]: The best previously known impossible differential attack was discovered by Ding *et al.* [DZLY17], based on a 4-round distinguisher and extended to an attack up to 7 rounds with 2^{56} data, $2^{53.8}$ time and 2^{43} bytes of memory. At Eurocrypt'17, Sasaki and Todo proposed a new way to search for impossible differentials [ST17] based on MILP, leading to much more sophisticated distinguishers than previously known. We implemented this algorithm and were able to find new impossible distinguishers over 5 rounds which are given in Table 9. Note that there are two different configurations for our distinguishers: either $1 + 2 + 2$ which means one forward round, the two middles rounds and 2 backward rounds, or $2 + 2 + 1$, *i.e.*, two forward rounds, two middle rounds and one backward round.

Table 9. Impossible differential distinguishers for 5 rounds.

$2+2+1$ Rounds	$1+2+2$ Rounds
0010000000000000 $\not\rightarrow$ 0000000040000000	0400000000000000 $\not\rightarrow$ 0000000000004000
0000000000100000 $\not\rightarrow$ 0000000001000000	0010000000000000 $\not\rightarrow$ 0000000000000010
0000000000004000 $\not\rightarrow$ 0400000000000000	0000000040000000 $\not\rightarrow$ 0010000000000000
0000000000000010 $\not\rightarrow$ 0010000000000000	0000000001000000 $\not\rightarrow$ 0000000000100000

Due to the specific shape of these impossible differentials (only one active bit in the input and output), we can take any one of them and use it to mount an attack up to 9 rounds. Using [BNPS14], we were able to estimate that the resulting attack would need $\alpha \cdot 2^{65}$ data and memory, and $2^{128} \cdot e^{-\alpha}$ time, where α is a parameter allowing for a trade-off between data/memory and time complexities (the higher is α, the higher is the data/memory complexity and the lower is the time complexity).

Integral and Higher-Order Differential [JNP^{+}13,Mor17,PN15,RR16c]: The longest known distinguisher of these types is a higher-order differential that is introduced in [Mor17]. This distinguisher includes 5 nonlinear layers and needs a data set of size 2^{32}. For key recovery, it is possible to append at most 3 rounds to the end of distinguisher and attack 8 round `PRINCEv2` by guessing 80 key bits. The complexity of this attack is 2^{112} computations (equivalent to $2^{107.4}$ 8-round `PRINCEv2` encryption), 2^{36} chosen plaintexts and 2^{36} blocks of memory.

A more recent technique to build integral distinguisher is to use the so-called *division property* introduced by Todo at Eurocrypt'15 [Tod15]. This technique was later refined into *bit-based division property* at FSE'16 by Todo and Morii [TM16] and some work was done to efficiently search for division property using *e.g.* Mixed Integer Linear Programming (MILP) [XZBL16,ZR19]. We implemented this algorithm to search for division property based distinguishers and we ended up finding such a distinguisher over 6 rounds. This distinguisher requires 2^{62} chosen plaintexts and due to this high data complexity, we do not expect it to be used to mount an attack over more than 8 rounds while also having better complexities than the above-mentioned attack.

Boomerang [PDN15]: The boomerang attack is applied to `PRINCE` in [PDN15], but there are some flaws on the estimation of the probability, e.g., the effect of the boomerang switching [BK09] is not taken in consideration. The so-called sandwich attack [DKS10,DKS14] is an experimental approach to estimate more rigorously this probability. We estimated the probability for a boomerang distinguisher with 4-round plus the middle layer. The probability of this 6-round distinguisher is about 2^{-34}, but the 7-round distinguisher is clearly worse than the classical differential attack because it involves 9 additional active S-boxes.

Accelerated Exhaustive Search [JNP^{+}13,PDN15,RR16a,RR16b]: These attacks on `PRINCE` either used the α-reflection and FX-construction property or the key-less middle rounds of the cipher to accelerate the exhaustive search. For `PRINCEv2`, these properties do not hold anymore and the only possible attack of this type is the one used in [RR16b]. Using the technique of [RR16b] and by starting from the middle of the cipher, attacking 4 round or more requires to guess all the key bits. By starting from the plaintext/ciphertext side, attacking 6 rounds or more requires to guess all the key bits.

Meet-in-the-Middle [CNPV13a,CNPV13b,DP15a,DP15b,LJW13,RR16b]: Meet-in-the-Middle attacks used in [LJW13,RR16b] took advantage of key-less middle rounds and use super-sboxes in the middle of the cipher to attack at most 10 rounds. These attacks do not work on `PRINCEv2` as effective as on `PRINCE`.

The sieve-in-the-middle attack, introduced in [CNPV13a,CNPV13b], is applicable to 8 rounds of `PRINCE` which includes 6 rounds for the sieve-in-the-middle and 2 rounds for the biclique part. Applying it to `PRINCEv2`, the sieve-in-the-middle part will be more complicated. The super-sbox used there will be key-dependent, which increases the time and memory complexity.

The meet-in-the-middle attack used in [DP15a,DP15b] reaches 10 rounds of `PRINCE`. Applying the tool given in [Der19] by Patrick Derbez which uses the same technique, we find out that it is possible to attack at most 6 rounds of `PRINCEv2` with the complexity of either 2^{96} computations and 2^{26} memory blocks or 2^{112} computations and 2^{6} memory blocks.

We also analyzed the security of `PRINCEv2` against Demirci-Selçuk meet-in-the-middle attacks using the same tool by Derbez. This attack can reach at most 10 rounds using 2^{48} chosen plaintexts, 2^{112} computations and 2^{70} memory blocks.

Time-Data-Memory Trade-Offs [Din15,JNP⁺13]: Excluding trivial Diffie-Hellman time-data-memory trade-offs, all of these attacks used FX-construction property of the `PRINCE` and do not work on the `PRINCEv2`.

Biclique [ALL12,YPO15]: These attacks could accelerate an exhaustive search maximally by a factor of 2, exploiting the FX-construction in `PRINCE`. Since `PRINCEv2` is not an FX-construction anymore and this attack does not improve the exhaustive search generally, we expect this attack to be not applicable.

Collisions [FJM14]: The FX-construction of `PRINCE` allows a collision-based attack, using 2^{32} data, 2^{96} off-line and 2^{32} on-line computations, to recover the key. But again, it does not apply to `PRINCEv2`.

Remarks about Related-Key Attacks: We emphasize that we never claim any security under related-key attacks, but it is also important to understand the impact on `PRINCEv2` when attackers can use related-key attacks.

First of all, when both rotational and XOR-difference relations are allowed, the trivial related-key distinguishing attack is possible by exploiting the convertible property from encryption to decryption. Even if the relationship is restricted to XOR-difference, attackers can still attack `PRINCEv2` by using related-key boomerang attacks. The related-key boomerang attack is applied to `PRINCE` without whitening keys in [JNP+13]. Inducing differences with the key allows attackers to construct iterative related-key differential characteristics whose number of active S-boxes is only one in each round. This iterative property is lost in the middle round R', but attackers can overcome R' by using related-key boomerang attack, where two iterative related-key differential characteristics are connected. The new key schedule for `PRINCEv2` is not designed to avoid this related-key boomerang attack, and there are related-key boomerang characteristics with 12 active S-boxes. Similarly to the single-key boomerang attack, evaluating the probability in detail requires to analyze the effect of the boomerang switching. However, we can estimate the probability is roughly $2^{-12 \times 2 \times 2} = 2^{-48}$ from the number of active S-boxes, and it implies that `PRINCEv2` is not secure against the related-key attack.

Again, we never claim any security under related-key attacks, and we believe that a related-key attack never happens in the environment that `PRINCEv2` is demanded.

6 Conclusion

The need for a secure and efficient low-latency block cipher which also has low-power and low-energy requirements is ever increasing with the widespread of multiple technologies using microcontrollers. While `PRINCE` family of block ciphers were specifically designed to tackle this problem, the recent lightweight crypto competition for AEAD from the NIST set a specific security level that `PRINCE` cannot reach. As a low-latency cipher would probably be deployed in a larger environment using such an AEAD primitive, it makes sense to want this low-latency cipher to reach the same security level. We show how to modify `PRINCE` to reach the required security level set by the NIST while minimizing the induced overhead, especially in a situation where `PRINCE` is already deployed.

We solve this problem by showing that a carefully built key-schedule is sufficient to provide the required security goal while keeping (almost) all of the remaining design untouched and propose `PRINCEv2` family of block ciphers. As proven by our various experiments, `PRINCEv2` only has a very small overhead compared to `PRINCE`, while still reaching the required higher security level. Moreover, the fact that the `PRINCE` and `PRINCEv2` designs are very similar allows one to implement both `PRINCE` and `PRINCEv2` in the same environment (e.g., for backward compatibility) with a very small overhead.

Finally, the similarities between `PRINCE` and `PRINCEv2` allow us to reuse a majority of the security analysis done by the community over the last 8 years since `PRINCE`'s publication. By doing so and carefully analyzing how the modifications made influenced the previously known attacks on `PRINCE`, as well as providing new cryptanalysis insights for both versions, we showed that `PRINCEv2` meets its security requirements.

We thus believe that `PRINCEv2` is a major improvement over `PRINCE` and we expect it to be widely adopted in the near future. Moreover, our work shows that one can improve the security level of some lightweight primitives with minimal downsides. An open question is thus to see if similar improvements could be made for other microcontroller-targeted ciphers such as `Midori`, `MANTIS` and `QARMA`, which could lead to interesting future work.

We made the reference implementation publicly available on GitHub in:

`https://github.com/rub-hgi/princev2`

Acknowledgments

The work described in this paper has been partially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and through the project 271752544 and partially by the German Federal Ministry of Education and Research (BMBF, project iBlockchain – 16KIS0901K).

References

ALL12. Farzaneh Abed, Eik List, and Stefan Lucks. On the security of the core of PRINCE against biclique and differential cryptanalysis. *IACR Cryptology ePrint Archive*, 2012:712, 2012.

Ava17. Roberto Avanzi. The QARMA block cipher family. *IACR Transactions on Symmetric Cryptology*, 2017(1):4–44, 2017.

BBI+15. Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015*, volume 9453 of *LNCS*, pages 411–436. Springer, 2015.

BCG+12. Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knežević, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE – A low-latency block cipher for pervasive computing applications – extended abstract. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 208–225. Springer, 2012.

BEK+20. Dušan Božilov, Maria Eichlseder, Miroslav Knežević, Baptiste Lambin, Gregor Leander, Thorben Moos, Ventzislav Nikov, Shahram Rasoolzadeh, Yosuke Todo, and Friedrich Wiemer. PRINCEv2. In *SAC 2020*, LNCS. Springer, Heidelberg, 2020.

BJK+16. Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, volume 9815 of *LNCS*, pages 123–153. Springer, 2016.

BK09. Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 1–18. Springer, 2009.

BNPS14. Christina Boura, María Naya-Plasencia, and Valentin Suder. Scrutinizing and improving impossible differential attacks: Applications to CLEFIA, Camellia, LBlock and Simon. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014*, volume 8873 of *LNCS*, pages 179–199. Springer, 2014.

CFG$^+$14a. Anne Canteaut, Thomas Fuhr, Henri Gilbert, María Naya-Plasencia, and Jean-René Reinhard. Multiple differential cryptanalysis of round-reduced PRINCE. In Carlos Cid and Christian Rechberger, editors, *Fast Software Encryption – FSE 2014*, volume 8540 of *LNCS*, pages 591–610. Springer, 2014.

CFG$^+$14b. Anne Canteaut, Thomas Fuhr, Henri Gilbert, María Naya-Plasencia, and Jean-René Reinhard. Multiple differential cryptanalysis of round-reduced PRINCE (full version). *IACR Cryptology ePrint Archive*, 2014:89, 2014.

CNPV13a. Anne Canteaut, María Naya-Plasencia, and Bastien Vayssière. Sieve-in-the-middle: Improved MITM attacks. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, volume 8042 of *LNCS*, pages 222–240. Springer, 2013.

CNPV13b. Anne Canteaut, María Naya-Plasencia, and Bastien Vayssière. Sieve-in-the-middle: Improved MITM attacks (full version). *IACR Cryptology ePrint Archive*, 2013:324, 2013.

Der19. Patrick Derbez. AES automatic tool. *https://seafile.cifex-dedibox.ovh/f/72be1bc96bf740d3a854/*, 2019.

Din15. Itai Dinur. Cryptanalytic time-memory-data tradeoffs for FX-constructions with applications to PRINCE and PRIDE. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 231–253. Springer, 2015.

DKS10. Orr Dunkelman, Nathan Keller, and Adi Shamir. A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3G telephony. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *LNCS*, pages 393–410. Springer, 2010.

DKS14. Orr Dunkelman, Nathan Keller, and Adi Shamir. A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3G telephony. *J. Cryptology*, 27(4):824–849, 2014.

DP15a. Patrick Derbez and Léo Perrin. Meet-in-the-middle attacks and structural analysis of round-reduced PRINCE. In Gregor Leander, editor, *Fast Software Encryption – FSE 2015*, volume 9054 of *LNCS*, pages 190–216. Springer, 2015.

DP15b. Patrick Derbez and Léo Perrin. Meet-in-the-middle attacks and structural analysis of round-reduced PRINCE. *IACR Cryptology ePrint Archive*, 2015:239, 2015.

DZLY17. Yao-Ling Ding, Jing-Yuan Zhao, Lei-Bo Li, and Hong-Bo Yu. Impossible differential analysis on round-reduced PRINCE. *J. Inf. Sci. Eng.*, 33(4):1041–1053, 2017.

FJM14. Pierre-Alain Fouque, Antoine Joux, and Chrysanthi Mavromati. Multi-user collisions: Applications to discrete logarithm, Even-Mansour and PRINCE. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014*, volume 8873 of *LNCS*, pages 420–438. Springer, 2014.

GR16a. Lorenzo Grassi and Christian Rechberger. Practical low data-complexity subspace-trail cryptanalysis of round-reduced PRINCE. In Orr Dunkelman and Somitra Kumar Sanadhya, editors, *Progress in Cryptology – INDOCRYPT 2016*, volume 10095 of *LNCS*, pages 322–342, 2016.

GR16b. Lorenzo Grassi and Christian Rechberger. Practical low data-complexity subspace-trail cryptanalysis of round-reduced PRINCE. *IACR Cryptology ePrint Archive*, 2016:964, 2016.

JNP$^+$13. Jérémy Jean, Ivica Nikolić, Thomas Peyrin, Lei Wang, and Shuang Wu. Security analysis of PRINCE. In Shiho Moriai, editor, *Fast Software Encryption – FSE 2013*, volume 8424 of *LNCS*, pages 92–111. Springer, 2013.

KNR12. Miroslav Knežević, Ventzislav Nikov, and Peter Rombouts. Low-latency encryption – is "lightweight = light + wait"? In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems – CHES 2012*, volume 7428 of *LNCS*, pages 426–446. Springer, 2012.

LJW13. Leibo Li, Keting Jia, and Xiaoyun Wang. Improved meet-in-the-middle attacks on AES-192 and PRINCE. *IACR Cryptology ePrint Archive*, 2013:573, 2013.

Mor17. Paweł Morawiecki. Practical attacks on the round-reduced PRINCE. *IET Information Security*, 11(3):146–151, 2017.

NIS. NIST. Lightweight cryptography. `https://csrc.nist.gov/projects/lightweight-cryptography`.

NIS18. NIST. Submission requirements and evaluation criteria for the lightweight cryptography standardization process. `https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf`, 2018.

NXP20. NXP. AN12278 LPC55S00 Security Solutions for IoT. `https://www.nxp.com/docs/en/application-note/AN12278.pdf`, 2020.

PDN15. Raluca Posteuca, Cristina-Loredana Duta, and Gabriel Negara. New approaches for round-reduced PRINCE cipher cryptanalysis. *Proceedings of the Romanian Academy, Series A*, 16:253–264, 2015.

PN15. Raluca Posteuca and Gabriel Negara. Integral cryptanalysis of round-reduced PRINCE cipher. *Proceedings of the Romanian Academy, Series A*, 16:265–270, 2015.

RR16a. Shahram Rasoolzadeh and Håvard Raddum. Cryptanalysis of 6-round PRINCE using 2 known plaintexts. *IACR Cryptology ePrint Archive*, 2016:132, 2016.

RR16b. Shahram Rasoolzadeh and Håvard Raddum. Cryptanalysis of PRINCE with minimal data. In David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *Progress in Cryptology – AFRICACRYPT 2016*, volume 9646 of *LNCS*, pages 109–126. Springer, 2016.

RR16c. Shahram Rasoolzadeh and Håvard Raddum. Faster key recovery attack on round-reduced PRINCE. In Andrey Bogdanov, editor, *Lightweight Cryptography for Security and Privacy – LightSec 2016*, volume 10098 of *LNCS*, pages 3–17. Springer, 2016.

ST17. Yu Sasaki and Yosuke Todo. New impossible differential search tool from design and cryptanalysis aspects – revealing structural properties of several ciphers. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, volume 10212 of *LNCS*, pages 185–215, 2017.

TM16. Yosuke Todo and Masakatu Morii. Bit-based division property and application to Simon family. In Thomas Peyrin, editor, *Fast Software Encryption – FSE 2016*, volume 9783 of *LNCS*, pages 357–377. Springer, 2016.

Tod15. Yosuke Todo. Structural evaluation by generalized integral property. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 287–314. Springer, 2015.

XZBL16. Zejun Xiang, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin. Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016*, volume 10031 of *LNCS*, pages 648–678, 2016.

YPO15. Zheng Yuan, Zhen Peng, and Haiwen Ou. Two kinds of biclique attacks on lightweight block cipher PRINCE. *IACR Cryptology ePrint Archive*, 2015:1208, 2015.

ZR19. Wenying Zhang and Vincent Rijmen. Division cryptanalysis of block ciphers with a binary diffusion layer. *IET Information Security*, 13(2):87–95, 2019.

A Code

SageMath code to generate the round constants:

```
a = RealField(prec=2000)(pi)-3
for i in range(1, 9):
b = (floor(a*2^(64*i)) + 2^64) % 2^64
print("0x%016x" % (b))
```

The output is:

```
0 0x243f6a8885a308d3
1 0x13198a2e03707344
2 0xa4093822299f31d0
3 0x082efa98ec4e6c89
4 0x452821e638d01377
5 0xbe5466cf34e90c6c
6 0xc0ac29b7c97c50dd
7 0x3f84d5b5b5470917
```

The 0th constant is not used in `PRINCE`, so we skip it, too. The second last constant (line 6) is α and thus we use the last one (line 7) as β.

B Test Vectors

Plaintext	k_0	k_1	Ciphertext
0000000000000000	0000000000000000	0000000000000000	0125fc7359441690
ffffffffffffffff	0000000000000000	0000000000000000	832bd46f108e7857
0000000000000000	ffffffffffffffff	0000000000000000	ee873b2ec447944d
0000000000000000	0000000000000000	ffffffffffffffff	0ac6f9cd6e6f275d
0123456789abcdef	0123456789abcdef	fedcba9876543210	603cd95fa72a8704

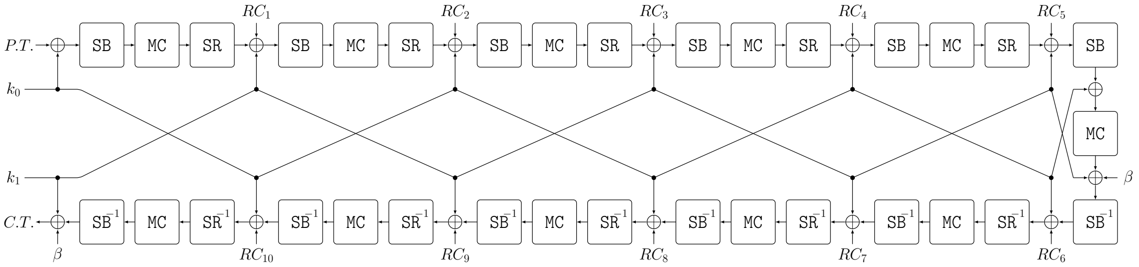


Fig. 9. `PRINCEv2` structure for encryption.

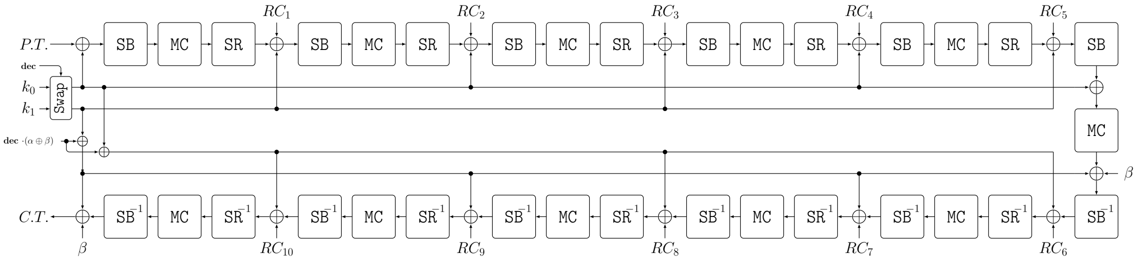


Fig. 10. `PRINCEv2` structure for encryption and decryption.

5.2 The SPEEDY Family of Block Ciphers

Publication Data

> Gregor Leander, Thorben Moos, Amir Moradi, and Shahram Rasoolzadeh. The SPEEDY family of block ciphers engineering an ultra low-latency cipher from gate level for secure processor architectures. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(4):510–545, 2021

The acceptance rate for Volume 2021 of the IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES) was **31,2%** [Acca].

Content This work introduces a family of ultra low-latency block ciphers called `SPEEDY` which is dedicated to standard-cell-based integrated circuit design. The cipher's construction is tailored to the latency properties of logic gates and logic circuits in advanced CMOS technology generations and thereby achieves a lower execution time in hardware than any other known secure encryption primitive. `SPEEDY` is designed for high-speed encryption inside of modern CPUs, in particular for advanced security features like memory encryption and secure cache architectures. Yet, it may be used in any application where high performance and security are the primary design goals.

Contribution The author of this thesis is a principal author of this publication. In detail, the general idea of this work together with all hardware-based considerations and implementations are contributed by the author of this thesis. He also contributed substantially to the general concept for low-latency design, the construction of the S-box and the writing of all sections. The author would like to thank all co-authors for their significant contributions to the design of the cipher and its security analysis.

The SPEEDY Family of Block Ciphers

Engineering an Ultra Low-Latency Cipher from Gate Level for Secure Processor Architectures

Gregor Leander[1], Thorben Moos[1], Amir Moradi[1] and
Shahram Rasoolzadeh*[2]

[1] Ruhr University Bochum, Horst Görtz Institute for IT Security, Bochum, Germany
firstname.lastname@rub.de
[2] Radboud University, Nijmegen, The Netherlands
firstname.lastname@ru.nl

Abstract. We introduce `SPEEDY`, a family of ultra low-latency block ciphers. We mix engineering expertise into each step of the cipher's design process in order to create a secure encryption primitive with an extremely low latency in CMOS hardware. The centerpiece of our constructions is a high-speed 6-bit substitution box whose coordinate functions are realized as two-level NAND trees. In contrast to other low-latency block ciphers such as `PRINCE`, `PRINCEv2`, `MANTIS` and `QARMA`, we neither constrain ourselves by demanding decryption at low overhead, nor by requiring a super low area or energy. This freedom together with our gate- and transistor-level considerations allows us to create an ultra low-latency cipher which outperforms all known solutions in single-cycle encryption speed. Our main result, `SPEEDY-6-192`, is a 6-round 192-bit block and 192-bit key cipher which can be executed faster in hardware than any other known encryption primitive (including `Gimli` in Even-Mansour scheme and the `Orthros` pseudorandom function) and offers 128-bit security. One round more, i.e., `SPEEDY-7-192`, provides full 192-bit security. `SPEEDY` primarily targets hardware security solutions embedded in high-end CPUs, where area and energy restrictions are secondary while high performance is the number one priority.

Keywords: Low-Latency Cryptography, High-Speed Encryption, Block Cipher

1 Introduction

In this paper we revisit the following fundamental problem: How do we design a secure encryption algorithm whose hardware implementation is fast? Specifically, we care about the entire latency of the hardware circuit from the point where the inputs are provided to the point where the final outputs are ready and stable, i.e., the latency of a fully-unrolled hardware implementation entirely made from combinatorial logic. Previous approaches, which led to the design of established low-latency constructions like `PRINCE` [BCG+12], `PRINCEv2` [BEK+20], `MANTIS` [BJK+16] and `QARMA` [Ava17], considered a low number of rounds and, to some extent, a small gate depth as design criteria. While both are obviously important factors to achieve a low latency, there are further aspects which have been ignored at the design level in the past – first and foremost the latency characteristics of the underlying hardware. At first sight it may appear to be of limited interest to tailor a cryptographic primitive towards one specific device technology due to the potential loss of generality. However, in the hardware world there has been only one de-facto standard for integrated circuit fabrication since the 1980s, namely Complementary Metal–Oxide–Semiconductor

---

*Part of this work was accomplished when S. Rasoolzadeh was at Ruhr University Bochum.

(CMOS) technology. The construction of CMOS logic gates, i.e., the arrangement of p- and n-channel MOSFETs (Metal–Oxide–Semiconductor Field-Effect Transistors) to create a certain functionality, has remained largely unchanged since its original proposal in 1963. In other words, CMOS logic gates – the essential building blocks for the vast majority of our computing technology today – have not experienced any fundamental redesign in almost 6 decades. Merely their size has seen a progressive decrease according to Moore's famous law [Moo65].

Notably, there are some operations which can be constructed more naturally from complementary logic. In particular, complementary gates in silicon hardware are naturally inverting and non-inverting Boolean functions cannot be realized in a single stage (i.e., they require more than one pull-up and pull-down network) [RCN04]. Among the naturally inverting logic gates some can be realized using only the minimum (lower bound) of $2n$ transistors, where n is the number of inputs the gate receives. These $2n$ transistors are then arranged in the classical layout of one pull-up network, built from p-channel MOSFETs (PMOS), and one pull-down network, built from n-channel MOSFETs (NMOS). The simple Boolean functions NAND, NOR and INV/NOT are constructed this way, but also the compound or complex logic gates AND-OR-INV (AOI) and OR-AND-INV (OAI). We argue that logic cells with these properties are immensely beneficial for low-latency constructions as they produce outputs much faster than their counterparts, independent of the particular specifications or the minimum feature size of the fabrication process.

When diving deeper into the physical characteristics of hardware circuits built from silicon, it is possible to make even further distinctions. In particular, we point out that cell layouts which require PMOS transistors to be connected in series (stacked) suffer from the lower mobility of PMOS compared to NMOS transistors more significantly. In consequence, a noticeable negative impact on the latency of such gates can be observed and larger transistor widths are required to partially offset this performance loss at the price of an increased area [RCN04]. Among the previously listed cells, only NAND and INV/NOT gates do not classically require PMOS transistors to be stacked. NOR gates with more than two inputs suffer most severely from the mobility mismatch due to the larger PMOS stacks. To clarify the impact of such observations on the performance of gates in common standard cell libraries, we present latency figures for individual logic gates exemplarily for NanGate 45 nm and 15 nm Open Cell Libraries (OCLs) in Section 2.

All gate- and transistor-level considerations described above are universally applicable to CMOS standard cells, independent of the particular foundry, manufacturing process and minimum feature size. Hence, it makes sense to take such characteristics into account when attempting to implement a certain function, like an encryption algorithm, as a hardware circuit with minimum latency. When revisiting previous latency-driven constructions in cryptography, it is clear that such low-level observations have not been considered in the past. We provide first contributions towards hardware-aware low-latency design and construct a family of ultra low-latency block ciphers based on the underlying principles.

1.1 Motivation

Approaches to secure the internals of modern Central Processing Units (CPUs) have received significant attention in the last few years as microarchitectural attacks, notably Meltdown [LSG+18] and Spectre [KHF+19], revealed serious shortcomings in the security architectures of widely deployed high-end processors. Hardware-based mitigations for such attacks are proposed "en masse". Many of them call for a higher level of encrypted communication *inside* of CPUs as well as between CPUs and their surrounding hardware components. Among the former are proposals for secure caches such as ScatterCache [WUG+19] and CEASER [Qur18]. Both of them are compared to a number of further cache architectures in [DXS19]. To implement new features of this kind in the next generations of mainstream processors without causing a large performance penalty, high-speed encryption primitives are among the most important building blocks.

Secure caches are only one example of security applications in CPU environments that require high-speed encryption. Dedicated hardware instructions, memory encryption, pointer authentication (as renownedly implemented using `QARMA` in ARM processors) and similar hardware-assisted mechanisms against software exploitation fall into this category as well. We expect to see a lot more of such features implemented in future generations of secure processor architectures, especially when more highly-optimized cryptographic primitives become available. `SPEEDY` is meant as a general purpose high-speed encryption primitive for all these applications and not limited or tailored to a subset of them.

Most low-latency ciphers published in the literature so far, such as `PRINCE` [BCG+12], `PRINCEv2` [BEK+20], `MANTIS` [BJK+16] and `QARMA` [Ava17], try to meet tight area and energy requirements in addition to low latency. These properties make them particularly suitable for highly-constrained microcontrollers in the Internet of Things (IoT). However, keeping the primitives suited for battery-powered devices requires sacrifices with respect to maximum performance. High-end CPUs do not impose the same kind of restrictions on area and energy, yet they require even higher performance in terms of latency and throughput. `SPEEDY` is able to outperform the state of the art by focusing on maximum encryption speed and high security only.

1.2 Related Work

Designing cryptographic primitives with minimum execution time in hardware is still a young and emergent research discipline. At CHES 2012 the authors of [KNR12] delivered first results in that area by comparing the latency properties of multiple (lightweight) block ciphers. It was concluded that, among other factors, the use of cryptographically-strong 4-bit (or even 3-bit) S-boxes should be favored over larger substitutions and that a low number of rounds should be maintained even at the price of a heavier linear layer when designing a low-latency primitive. These demands were immediately met by the first dedicated low-latency block cipher called `PRINCE` which has been presented at ASIACRYPT 2012. `PRINCE` is a 64-bit block cipher with a 128-bit key and 12 cipher rounds which features an innovative reflection property that allows to encrypt and decrypt data with essentially the same circuit. Recently, an updated version called `PRINCEv2` has been proposed which claims to increase the security level of `PRINCE` by making small modifications to the key schedule and the middle rounds [BEK+20]. This work also provides a comparison of multiple low-latency block ciphers which confirms that `PRINCE` and `PRINCEv2` are still the fastest such primitives in public literature [BEK+20]. The comparison also includes the tweakable block ciphers `MANTIS` [BJK+16] and `QARMA` [Ava17] as well as the low-energy block cipher `Midori` [BBI+15] and demonstrates that all three of them come at a latency overhead between 22 % and 42 % (considering the encryption-only variants) compared to `PRINCE` in open-source NanGate libraries. This result may not come as a surprise, since tweakable block ciphers such as `MANTIS` and `QARMA` are expected to require a larger circuit depth due to the additional tweak input and since `Midori` has not been designed with low latency being the primary design goal, although its substitution layer has been chosen particularly to offer a small delay. However, two recent works claim that cryptographic primitives aside from traditional block ciphers are able to outperform `PRINCE` in terms of latency. First, the high performance cross-platform permutation `Gimli` introduced in [BKL+17] is claimed to enable encryption with a 1.7 times smaller latency than `PRINCE` in [GKD20], while the low-latency pseudorandom function (PRF) `Orthros` introduced in [BIL+21] claims to achieve a latency about 7 % below `PRINCE`'s. We analyze both claims in our comparison in Section 7 and conclude that the latter is consistent with our results, while the former is clearly not. `Orthros` is able to achieve a lower latency than `PRINCE` by computing the sum of two keyed permutations [BIL+21] which makes the resulting primitive non-invertible (in contrast to block ciphers like `SPEEDY`).

Apart from the full cryptographic primitives discussed above, there are also some works focusing on particular cryptographic building blocks only. For instance, in [LSL+19] it is shown how to construct involutory low-latency Maximal Distance Separable (MDS) matrices. The authors of [BFP19] present techniques for finding small low-depth circuits for cryptographic functions. In [BMD+20] the main goal is to construct S-boxes whose masked variants (i.e., their side-channel protected versions) have a low latency in hardware which conceptually requires a low AND depth and AND gate complexity. Low-latency hardware masking in general, used to protect cryptographic primitives against side-channel attacks, has received significant attention in the last few years, as demonstrated in [MS16, GIB18, ABP+18, BKN19, SBHM20]. However, this field is not directly related to the development of low-latency symmetric primitives in general, as the requirements are vastly different and sometimes even direct opposites.[1]

1.3 Our Contribution

We introduce `SPEEDY`, a family of ultra low-latency block ciphers dedicated to semi-custom, i.e., standard-cell-based, integrated circuit design. In order to tailor this cryptographic primitive towards maximum execution speed in hardware we first analyze which type of logic gates and circuit topologies are particularly suited for ultra low-latency encryption. Our considerations in this regard are novel and have, to the best of our knowledge, not been applied in previous designs of symmetric cryptographic primitives.

`SPEEDY` can be instantiated with different block and key sizes and varying numbers of rounds. However, due to our S-box width of 6 bits and our main target application of 64-bit high-end CPUs we decided to use the least common multiple of 6 and 64, namely 192 as the default block and key size and call this instance `SPEEDY-r-192`. We claim that `SPEEDY-r-192` achieves 128-bit security when iterated over $\texttt{r} = 6$ rounds and full 192-bit security when iterated over $\texttt{r} = 7$ rounds, while the $\texttt{r} = 5$ round variant already provides a decent security level that is sufficient for many practical applications. Our extensive evaluation of hardware implementations in 6 different standard cell libraries shows that both `SPEEDY-5-192` and `SPEEDY-6-192` achieve a lower latency in hardware than any other known encryption primitive, while `SPEEDY-7-192` is only marginally slower than `PRINCE`. Considering the provided security levels this is a significant improvement over the state of the art in the area of (ultra) low-latency cryptography.

2 Background

In this section we revisit the necessary concepts which build the foundation for `SPEEDY` and analyze the primary traits that make certain CMOS standard cells and circuit topologies particularly useful for high-speed cryptography.

2.1 Natural CMOS Gates (NCGs)

A static CMOS gate is constructed by combining a pull-up with a pull-down network. The pull-up network, as the name suggests, is responsible for pulling the output of the gate up to `VDD` whenever the Boolean function should result in a logical '1'. The pull-down network, analogously, is responsible for pulling the output down to `GND` whenever the Boolean function should output a logical '0'. The networks are built in a mutually exclusive manner such that only one of them is conductive for each combination of input signals [RCN04]. While the pull-up networks are exclusively built from PMOS devices, the pull-down networks are built from NMOS devices. PMOS devices can be understood

---

[1] In regular cryptographic S-boxes, non-linear gates such as AND and NAND are beneficial for area and latency over linear gates like XOR and XNOR for instance. In masked S-boxes on the other hand, linear operations are optimal and non-linear gates are the primary cost factor [BMD+20].

as switches that conduct current between their drain and source terminals whenever their gate voltage is low, NMOS devices conduct current between the terminals whenever their gate voltage is high. For the opposite gate voltages the transistors are in a high-resistance state. The assignment of PMOS transistors to pull-up networks and NMOS to pull-down networks originates from the fact that PMOS devices cannot produce so-called *strong zeros*, while NMOS devices cannot produce *strong ones* [RCN04]. In consequence, static CMOS gates with a single stage are naturally inverting by design. Non-inverting Boolean functions require at least two stages of pull-up and pull-down networks. Thus, as already discussed in Section 1, certain logic functions are a more natural fit for technologies that are based on complementary metal–oxide–semiconductor logic. Inverting Boolean functions include for instance the common logic gates INV/NOT, NAND, NOR, XNOR, AOI and OAI. Most of them (all except XNOR) can be realized as static gates by using only the lower bound of $2n$ devices, namely n PMOS and n NMOS transistors. We call all inverting logic gates which require only one stage and $2n$ transistors for their implementation *Natural CMOS Gates (NCGs)*. All NCGs commonly found in standard cell libraries with $1 \leq n \leq 4$ inputs are depicted in Appendix A, Figure 4. Such logic cells are not only interesting from a hardware design perspective because they require a lower number of transistors and therefore have a smaller area footprint, they are also faster than their opposition and therefore beneficial for low-latency constructions.

2.2 Latency of CMOS Logic Gates

The time that a physical instance of a logic gate requires to respond to a change in its input signals by updating its output signal is called the delay or the latency of a cell. Considering CMOS hardware, the latency of a physical instance of a logic cell depends on a number of factors. Besides environmental influences like the temperature and the supply voltage, also the transition time of the input signals and the capacitance that needs to be driven at its output play a significant role. In this subsection, however, we want to compare the base latencies of static CMOS gates when all outside factors are equal. Tables 1 and 2 list the latencies of common logic gates in two open-source standard cell libraries, namely NanGate 45 nm and 15 nm Open Cell Libraries (OCLs), respectively. The latency values are given in picoseconds and have been obtained by analyzing a netlist containing only the individual logic gate enclosed between standard D-flip-flop cells for typical operating conditions (25 °C, nominal voltage) with the Electronic Design Automation (EDA) software *Synopsys Design Compiler Version O-2018.06-SP4* using Composite Current Source (CCS) models of the standard cells. Please note that for simplicity only the logic gates with the minimum drive strength (denoted by the suffix `"_X1"` in NanGate libraries) are shown here. However, the following arguments and considerations also apply to the higher drive strength variants. As expected, the natural CMOS gates, defined in the previous subsection, produce their outputs significantly faster than the competition. Interestingly, though, some significant differences between analogous natural gates such as NAND and NOR can be observed. In NanGate 45 nm technology for example, the `NAND4_X1` cell is more than twice as fast as the `NOR4_X1` cell. This is due to the different physical behavior of p-type and n-type MOSFETs realized in silicon as semiconductor material. In n-type MOSFETs the majority carriers are electrons which are negatively charged. In p-type MOSFETs on the other hand, the majority carriers are positively charged holes [RCN04]. Holes are less mobile than electrons, which means they move slower. Therefore, simply speaking, PMOS transistors operate slower than NMOS transistors of the same size. This situation is even amplified when connecting PMOS devices in series (stacking) and leads to a significant performance degradation and an increased area demand due to the larger widths required to partially offset the performance penalty and achieve balanced rise and fall times. Classic CMOS NOR gates require stacks of n PMOS transistors and are therefore among the logic functions which suffer the most from the lower mobility of holes as majority carriers. Since

Table 1: Fan-In, Latency, Fan-In-to-Latency-Ratio and Linearity of logic gates in NanGate 45nm Open Cell Library (OCL) for typical operating conditions.

Cell Name	Fan-In	Latency [ps]	FLR	Linearity
`INV_X1`	1	22.047900	0.045356	2
`BUF_X1`	1	33.556521	0.029800	2
`AND2_X1`	2	40.170699	0.049788	2
`NAND2_X1`	2	27.885556	0.071722	2
`NOR2_X1`	2	40.649809	0.049201	2
`OR2_X1`	2	56.413554	0.035452	2
`XNOR2_X1`	2	57.604454	0.034720	4
`XOR2_X1`	2	73.018849	0.027390	4
`AND3_X1`	3	51.869132	0.057838	6
`AOI21_X1`	3	51.618919	0.058118	6
`MUX2_X1`	3	75.174913	0.039907	4
`NAND3_X1`	3	34.766912	0.086289	6
`NOR3_X1`	3	61.542571	0.048747	6
`OAI21_X1`	3	32.650799	0.091881	6
`OR3_X1`	3	85.839920	0.034949	6
`AND4_X1`	4	65.491892	0.061076	14
`AOI22_X1`	4	57.255469	0.069862	6
`NAND4_X1`	4	44.487149	0.089914	14
`NOR4_X1`	4	91.312885	0.043805	14
`OAI22_X1`	4	54.596245	0.073265	6
`OR4_X1`	4	118.592046	0.033729	14

both types of complex gates, AOI and OAI, require stacked PMOS transistors in their layouts as well, we can make similar arguments here, although the effect is less striking since the stacks are smaller. OAI gates are typically faster than AOI gates in common standard cell libraries since the internal capacitances in the pull-up networks of AOI gates are larger. NAND and INV/NOT gates are the only NCGs that do not require PMOS stacks in their classical layout. As a result, INV/NOT and NAND2 gates are almost exclusively the fastest CMOS gates for $n = 1$ and $n = 2$ in any CMOS gate library. For $n = 3$ and $n = 4$ the situation depends on the exact sizing of the transistors chosen by the cell designer for each particular gate. This choice determines the trade-off between area and latency of the logic cells. Typically, either NAND3 and NAND4 or OAI21 and OAI22 are the fastest gates for $n = 3$ and $n = 4$, respectively. In NanGate 45 nm technology OAI21 ($n = 3$) and NAND4 ($n = 4$) are the fastest cells for their respective number of inputs while in 15 nm technology NAND3 ($n = 3$) and OAI22 ($n = 4$) cells are the fastest, as apparent in Tables 1 and 2.

2.2.1 Suitability for High-Speed Encryption

There are several factors to be considered when determining which cells in a standard gate library are most suitable for low-latency encryption. Building a low-latency encryption primitive in hardware is essentially the task of creating a circuit that, as quickly as possible, establishes an, as highly as possible, non-linear relationship between the plaintext and, as many as possible, independent key bits. Of course, this is an extreme oversimplification of the large number of requirements that symmetric cryptographic primitives need to fulfill in order parry all known attacks. Yet, when following this simplified idea, the design process for an ultra low-latency cipher should start at the gate level. In particular, we are interested in logic gates that are capable of establishing a Boolean relationship between

Table 2: Fan-In, Latency, Fan-In-to-Latency-Ratio and Linearity of logic gates in NanGate 15nm Open Cell Library (OCL) for typical operating conditions.

Cell Name	Fan-In	Latency [ps]	FLR	Linearity
`INV_X1`	1	1.580082	0.632879	2
`BUF_X1`	1	3.068201	0.325924	2
`AND2_X1`	2	3.579786	0.558692	2
`NAND2_X1`	2	2.030621	0.984920	2
`NOR2_X1`	2	2.554366	0.782973	2
`OR2_X1`	2	3.643867	0.548867	2
`XNOR2_X1`	2	6.788322	0.294624	4
`XOR2_X1`	2	5.268465	0.379617	4
`AND3_X1`	3	5.496015	0.545850	6
`AOI21_X1`	3	3.394032	0.883904	6
`MUX2_X1`	3	6.133133	0.489146	4
`NAND3_X1`	3	2.360978	1.270660	6
`NOR3_X1`	3	3.787567	0.792065	6
`OAI21_X1`	3	2.830147	1.060016	6
`OR3_X1`	3	5.862194	0.511754	6
`AND4_X1`	4	7.125210	0.561387	14
`AOI22_X1`	4	4.070343	0.982718	6
`NAND4_X1`	4	4.659015	0.858551	14
`NOR4_X1`	4	5.250172	0.761880	14
`OAI22_X1`	4	3.775570	1.059443	6
`OR4_X1`	4	7.682688	0.520651	14

as many inputs as possible in a short period of time. In that regard, we introduce a new metric, which we call the Fan-in-to-Latency Ratio (FLR). Essentially, we divide the fan-in n of each gate (i.e., the number of inputs it receives) by its latency. Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be the Boolean function of a logic gate and n the number of inputs it receives (i.e., the fan-in), then the *Fan-in-to-Latency Ratio (FLR)* of f can be expressed as Equation 1.

$$\mathrm{FLR}(f) = \frac{n}{\mathrm{latency}(f)} \tag{1}$$

By calculating the FLR for each logic gate in a standard cell library one can rank the gates by their suitability for ultra low-latency encryption. Tables 1 and 2 list the FLR scores for all standard logic gates with n inputs for $1 \leq n \leq 4$. The FLR score reflects the ability of a logic gate to rapidly evaluate a Boolean function on multiple inputs. Hence, the higher the value in the FLR-column for a logic gate, the higher is its potential to be suitable for ultra low-latency encryption. NAND and OAI gates are among the logic cells with the highest FLR scores, while XOR and XNOR gates are among the worst performers. Thus, despite the importance of XOR (and XNOR) gates in symmetric cryptography (mostly for key addition and strong linear layers) it is prudent to limit their occurrence to a minimum. Obviously, the kind of Boolean logic function that is evaluated plays a significant role in determining its suitability for high-speed encryption as well. In that regard, a further important aspect is the linearity of a function. $\mathrm{Lin}(f)$ denotes the linearity of the Boolean function f, defined by Equation 2, where $\widehat{f} : \mathbb{F}_2^n \to \mathbb{Z}$ is the Fourier transform of f given by Equation 3.

$$\mathrm{Lin}(f) := \max_{\alpha \in \mathbb{F}_2^n} \left| \widehat{f}(\alpha) \right| \tag{2}$$

$$\widehat{f}(\alpha) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) + \langle \alpha, x \rangle} \tag{3}$$

Tables 1 and 2 provide the linearity of all listed logic gates. The linearity of a Boolean function $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ is lower bounded by $2^{\frac{n}{2}}$ and upper bounded by 2^n. Whenever $\mathrm{Lin}(f) = 2^n$, f is an affine function, i.e., Equation 4 holds with $\alpha \in \mathbb{F}_2^n, c \in \mathbb{F}_2$.

$$f(x) = \langle \alpha, x \rangle + c \tag{4}$$

In our tables, the logic functions INV/NOT, BUF, XOR, XNOR have maximum linearity (2^n) and can be expressed as constant or affine functions, while the logic gates AND2, NAND2, NOR2 and OR2 reach the lower bound for the linearity of $2^{\frac{n}{2}}$.

While both, linear and non-linear functions, are useful for the construction of secure encryption algorithms, they are typically used in different layers or round operations. The non-linear layer in block cipher design is typically the substitution layer while all other operations tend to be linear. Often the substitution boxes, in short S-boxes, are among the most resource consuming elements in terms of area, energy and latency. Therefore, it is particularly interesting to optimize this building block towards the desired design goal when developing and implementing a cipher. In that regard, non-linear gates with a high FLR score, like NAND and OAI, are the prime candidates for building strong and fast S-boxes.

2.3 Latency of Logic Circuits

It is insufficient to consider only the latencies of individual logic elements in order to determine the resulting total latency of a combinatorial circuit or path. When connecting logic gates to logic circuits, the individual propagation delays of the gates depend significantly on their direct electrical environment. Merely summing up the base latencies of the gates in a path (e.g., the values given in Tables 1 and 2) may give a *very* incorrect idea about the path's total latency. Despite the fact that some obvious correlation between these quantities can be observed, the gate depth of a path is not always directly proportional to its latency. Therefore, it is important to also consider adequate circuit topologies which minimize the latency of combinatorial circuits when designing a low-latency cipher. In this regard, we first want to dispel two common myths about the latency of CMOS circuits:

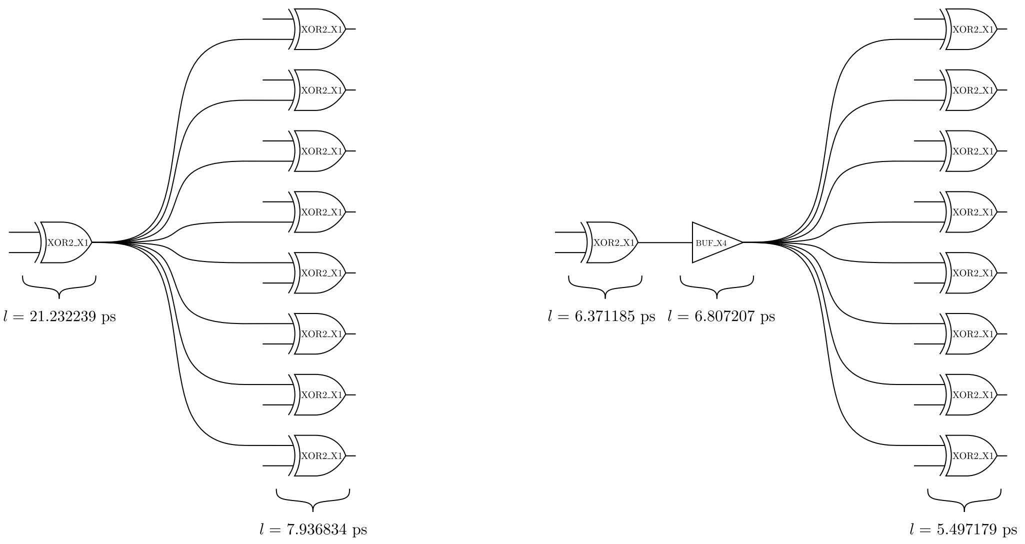
- *Myth 1:* Each CMOS standard cell has a fixed delay and each instantiation of the same exact standard cell adds (approximately) the same latency to a path.

  Truth: This is false. The propagation delay of a CMOS cell is always a function of the transition time of its input signals, which is influenced by the drive strength of preceding cells and the capacitance of the nets they need to drive, as well as the capacitive load that the CMOS cell itself needs to drive at its output. The variations of the delay of a cell instance depending on its electrical environment can easily be in the range of 200-300%. Therefore, it is not uncommon that two instances of the same cell in different positions of a logic circuit have delays associated with them (e.g., in a timing report) that differ by a factor of 3 or 4.

- *Myth 2:* Adding a gate to a path of a circuit and not making any other changes to the path will always increase the path's latency.

  Truth: This is also false. Often, adding a well-placed buffer or inverter (where logically applicable) to a path in order to charge a significant capacitive load faster can decrease the overall latency of the path. Hence, the mere gate depth is not always indicative of the latency of a circuit. Generally, the topology of a circuit, primarily the fan-out of the logic cells, is similarly important as the number and type of gates in its critical path when determining the maximum latency.

In the following we provide an example which demonstrates the incorrectness of the two myths. We consider a simple circuit in Figure 1(a) where the output signal of a single



(a) without buffering

(b) with buffering

Figure 1: Impact on the latency of the circuit in NanGate 15 nm technology when buffering the high fan-out net. Total latency is 29.169073 ps without the buffer (left) and 18.675571 ps with the buffer (right), despite the larger gate depth on the right.

XOR logic gate in NanGate 15 nm technology (`XOR2_X1`) is the input to 8 further XOR cells. The respective maximum latencies for each of the two circuit stages are denoted below the gates in Figure 1. While the base latency of a simple XOR logic gate in this technology is 5.268465 ps according to Table 2, it is obvious that the actual latencies of the gates in this circuit are significantly larger. The first XOR gate in particular which feeds the other 8 gates requires a latency which is more than 4 times as large as its base latency due to the significant capacitive load it needs to drive. The XOR gates in the second stage do not drive any large loads but their latency is increased because their input signals have a large transition time. It is noteworthy that this is a synthesis result, which means that the actual capacitances and resistances of the routing (i.e., wiring) are not even considered yet. After placing and routing this circuit in a chip design the latencies would likely be even larger. Figure 1(b) shows a circuit with the same logic functionality and the same 9 total XOR gates, but here the output of the first stage XOR is buffered by a drive strength buffer (`BUF_X4`). Although this change increases the gate depth of the circuit, it decreases its overall latency. The first stage XOR now only needs to drive a small load and the last stage XORs are driven by input signals with a short transition time. As a result, the buffered circuit has a total latency of 18.675571 ps (Fig. 1(b)) while the circuit without a buffer has a total latency of 29.169073 ps (Fig. 1(a)). Hence, the buffered circuit is more than 35% faster. Please note that the NanGate 15 nm library does not provide XOR gates with a higher drive strength, thus up-sizing the first stage XOR itself is not an option here and buffering the high fan-out net is inevitable when the latency should be reduced. Of course, this is done automatically by the synthesis tool. Our point here is simply that, regardless of how the large fan-out is addressed by the tool or the designer, e.g., up-sizing the gate or inserting a buffer, it assuredly causes an increased latency compared to a circuit with the same depth and the same gates in both levels, but with smaller fan-outs. Thus, we conclude that dedicated low-latency circuits should use topologies where the fan-outs of the logic gates are as small as possible (e.g. tree-based).

2.3.1 Finding Circuits with Minimum Latency

We would like to caution against the common perception that professional synthesis tools can readily be used to find and generate a netlist with minimum achievable latency for a simple Boolean function like an S-box coordinate function. First of all, the complexity of checking any possible circuit representation composed of a finite (but usually large) set of standard cells for a Boolean function is often remarkably high and market-leading EDA tools are built for time efficiency (especially the synthesis routines). Furthermore, the proprietary synthesis algorithms may not be sufficiently configurable to consider latency as the only or primary design goal. The tools may rather take area and energy into account as well and not consider latency optimizations that come at a harsh penalty for the other two optimization goals. In our own experience, the thresholds for such decisions cannot be adjusted sufficiently by the designer. Thus, we have found that constructing optimal building blocks for ultra low-latency cryptography needs to be done from scratch (by hand or via heuristics) instead of analyzing many different variants with a synthesis tool and selecting the ones that delivered the best performance. In our evaluations, the synthesis algorithms usually produced the best results with respect to low latency, when the underlying gate structure was already given and only incremental performance optimizations were required.

3 Ultra Low-Latency 6-bit S-box

In this section, we describe the technique we have used to build an ultra low-latency S-box from gate level. In order to design an S-box which is extremely fast in CMOS hardware while at the same time provides good cryptographic properties, we used the following criteria:

- Ultra low-latency: As explained in Subsection 2.2, NAND and OAI gates are among the best-suited logic gates for low-latency S-box design. Thus, we search for S-boxes that can be realized with as few as possible levels of only NAND and OAI gates. Furthermore, as discussed in Subsection 2.3, we try to make sure that in as many stages as possible the logic gates have a minimum fan-out.
- Bijective mapping with fully-dependent outputs: Since we aim for an SPN cipher, we need the S-box to be a bijective mapping. Moreover, we restrict the search to the S-boxes with fully-dependent outputs. In more detail, this means that all input bits are involved in the computation of each output bit.
- Small linearity and uniformity: To provide strong resistance against differential and linear attacks, we are only interested in S-boxes with small *uniformity* u and *linearity* l defined as

$$u = \mathrm{Uni}(S) := \max_{\substack{\alpha,\beta\in\mathbb{F}_2^n \\ \alpha\neq 0}} \left|\{x \in \mathbb{F}_2^n | S(x) \oplus S(x \oplus \alpha) = \beta\}\right|,$$

$$l = \mathrm{Lin}(S) := \max_{\substack{\alpha,\beta\in\mathbb{F}_2^n \\ \beta\neq 0}} \left|\sum_{x\in\mathbb{F}_2^n} (-1)^{\langle\alpha,x\rangle\oplus\langle\beta,S(x)\rangle}\right|.$$

By definition, the latency of a vectorial Boolean function, e.g., an S-box, is the maximum of the latencies of its coordinate Boolean functions. Besides, to have a bijective fully-dependent S-box with a small linearity, all of its coordinate functions must be balanced, fully-dependent and have a small linearity. Hence, our strategy was to first find low-latency Boolean functions and in a second step try to combine those into an S-box.

It is noteworthy that the S-boxes within the same class of extended bit-permutation equivalence have roughly the same latency cost (with a small margin of tolerance). Moreover, those functions will have the same uniformity and linearity. We recall from [LP07] that two n-bit to m-bit vectorial Boolean functions F and G of the form $\mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ are called extended bit-permutation equivalent, if there exist $a \in \mathbb{F}_2^n$, $b \in \mathbb{F}_2^m$, P_{in} a bit permutation function of n bits and P_{out} a bit permutation function of m bits such that

$$G(x) = P_{out} \circ F \circ P_{in}(x \oplus a) \oplus b \quad \forall x \in \mathbb{F}_2^n .$$

Therefore, it is sufficient to consider S-boxes only up to this equivalence.

3.1 Suitable Boolean Functions

To achieve a minimal latency, we searched for coordinate functions that can be realized in only two levels of NAND and OAI gates, or more specifically NAND2, NAND3, NAND4, OAI21 and OAI22 gates, while the larger and slower NAND4 and OAI22 gates should only be used in one of both levels. Additionally the first stage of NAND and OAI gates should have a fan-out of 1 for each gate. With this restriction, we are able to find Boolean functions with an extremely low latency in CMOS hardware.

We empirically found that Boolean functions based on NAND gates exclusively achieve the best cryptographic properties and latencies with only two levels at a higher quantity; therefore, in the following we limit ourselves to S-boxes which are possible to be built only from NAND gates. However, using the same process described in the following we have created S-boxes based on OAI gates exclusively (functions based on a mix between NAND and OAI have shown to be less promising) and compare them to the NAND-based boxes at the end of this section.

By considering all the possibilities for the inputs of the NAND gates at the first level, we aim at building all the n-bit Boolean functions $f(x_0, \ldots, x_{n-1})$; i.e., for each input for NAND gates we test $2n$ possible inputs: either x_i or its inverted value $\neg x_i$ with $0 \leq i < n$. We then filter the Boolean functions with respect to the aforementioned criteria, that is balancedness and low-linearity. Please note that selecting the inverted inputs requires additional inverter gates before the first stage of NAND gates. Yet, since each of the S-box inputs feeds multiple coordinate Boolean functions at the same time it is prudent to instantiate buffers to drive those nets anyway and an inverter can serve the same purpose. Following this argument, the inverted inputs do not cause any significant extra cost.

The first step is to find all the Boolean functions $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ which are: 1) possible to be built by using two levels of NAND gates as explained previously, 2) balanced, 3) fully-dependent on all the input bits, and 4) with linearity at most l. It is important to mention that the order of checking these features is quite important for reducing the computational cost.

We save all those Boolean functions in a set, named $\mathcal{F}$. Note that if there is a function $f \in \mathcal{F}$, then all of its extended bit-permutation equivalent functions such as $g(\cdot) = f \circ P(\cdot \oplus a) \oplus b$ with $a \in \mathbb{F}_2^n$, $b \in \mathbb{F}_2$ and P a bit permutation function of n bits, are included in $\mathcal{F}$. Next, we reduce the Boolean functions within $\mathcal{F}$ by the extended bit-permutation equivalence, and only keep one representative of each equivalence class in another set $\mathcal{F}^*$. Note that if there are N_f^* Boolean functions in $\mathcal{F}^*$, then there are about $N_f = N_f^* \cdot (n! \cdot 2^{n+1})$ functions in $\mathcal{F}$. This reduction corresponds to the $n!$ permutations of the input bits, the 2^n constants we can add to the input and the single bit we can add to the output.

3.2 Building Sboxes

To find all the bijective S-boxes $S = (f_0, \ldots, f_{n-1})$ such that each coordinate function is in $\mathcal{F}$, we can simply choose n of those N_f functions and then check for the necessary criteria, but this requires about $(N_f)^n$ steps of checking all the criteria which for $n > 4$

is a large computation cost. The two main options to reduce this cost is (i) considering permutation equivalence and (ii) to select the coordinate function step-by-step and filter after each additional choice.

Since it is sufficient to find the bijective S-boxes up-to the extended bit-permutation equivalence, we can restrict the first coordinate function f_0 to be chosen from $\mathcal{F}^*$ that is due to the freedom on choosing the constant and the bit-permutation in the input of the S-box. Besides, for all the other coordinate functions $f_1, \ldots, f_{n-1}$, we can fix an input's output to a constant, e.g., $f_i(0) = \mathtt{0}$ and this is because of the freedom in the output constant of the S-box. Note that since f_0 is chosen from $\mathcal{F}^*$ and it is a representative function, we already considered that $f_0(0) = \mathtt{0}$. Moreover, since we are still left with the freedom on the output bit-permutation of the S-box, we can fix the order of the coordinate functions of the S-box. In other words, if we consider that the elements of $\mathcal{F}$ are indexed, then we can fix the index of f_1 to be smaller than the index of f_2 and both are smaller than the index of f_3 and so on. This way, we reduce the number of choices to build an S-box to about $N_f^n/(n! \cdot 2^n)^2 \approx (N_f^*)^n \cdot (n!)^{n-2} \cdot 2^{n^2-n}$. In case of $n = 5$, this number is about $(N_f^*)^5 \cdot 2^{41}$ which is still not feasible to search.

The other main technique to reduce the computation cost of this search is that instead of choosing all the coordinate functions at once and then check for the criteria, we choose them one by one and in each step of choosing a coordinate function, we check for the probable possible criteria. In more details, in step one, we choose $f_0 \in \mathcal{F}^*$, then in step 2, we choose $f_1 \in \mathcal{F}$. Before, going to step 3, we can check for balancedness and linearity of the component function $f_0 \oplus f_1$. We go to the next step, if the criteria for $f_0 \oplus f_1$ have met, otherwise, we stay in step 2 and choose another function as f_1. In step 3, after choosing $f_2 \in \mathcal{F}$, we again can check for balancedness and linearity of the component functions $f_0 \oplus f_2$, $f_1 \oplus f_2$, $f_0 \oplus f_1 \oplus f_2$. We go to step 4, if all these criteria have met. In this way, we choose all the n coordinate functions to build the S-box, and then we can check for the uniformity criterion.

This technique, together with several other low-level techniques for speeding up the search, reduces the computation cost of this search significantly. Our search algorithm is written in `C++` code and we run it on an Intel Core i7 CPU with 8 threads for about 10 days to exhaustively search all the possible 6-bit S-boxes. Finding all 5-bit S-boxes only requires about two hours.

We also constructed 7- and 8-bit S-boxes, but due to the larger linearity or uniformity value, they would not have been beneficial over the 6-bit S-box.

3.3 Results

In case of 6-bit S-boxes, the minimum linearity and the minimum uniformity of all S-boxes possible to built, is 24 and 8, respectively. For these properties, up to the extended bit-permutation equivalence, there are only two class of such S-boxes. We choose the S-box class equivalent to the one shown in Figure 2 and given in Table 3, because of the higher algebraic degree.

For the chosen S-box class, we have the freedom to choose the input/output constants a and b and also P_{in} and P_{out} bit-permutation functions. We choose the output constant b in such a way that there is no need to insert an inverter in the output of the NAND gates of the second gate level. Even though it is a tiny improvement, the input constant a is chosen in a way to minimize the latency of the whole structure.

Finally, we choose the bit-permutations in such a way that it improves the cryptographic properties of the round function for `SPEEDY` which is explained in more detail in Section 6. Note that the optimum choice of these bit-permutations can be different for round functions of different primitives. Altogether, we end up with the S-box presented in Table 3. Its corresponding implementation is depicted in Figure 2. Furthermore, the disjunctive normal form (DNF) of the S-box is presented below, which is equivalent to the representation by

Table 3: The 6-bit S-box of `SPEEDY`.

x_0x_1	$x_2x_3x_4x_5$															
	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.a	.b	.c	.d	.e	.f
0.	08	00	09	03	38	10	29	13	0c	0d	04	07	30	01	20	23
1.	1a	12	18	32	3e	16	2c	36	1c	1d	14	37	34	05	24	27
2.	02	06	0b	0f	33	17	21	15	0a	1b	0e	1f	31	11	25	35
3.	22	26	2a	2e	3a	1e	28	3c	2b	3b	2f	3f	39	19	2d	3d

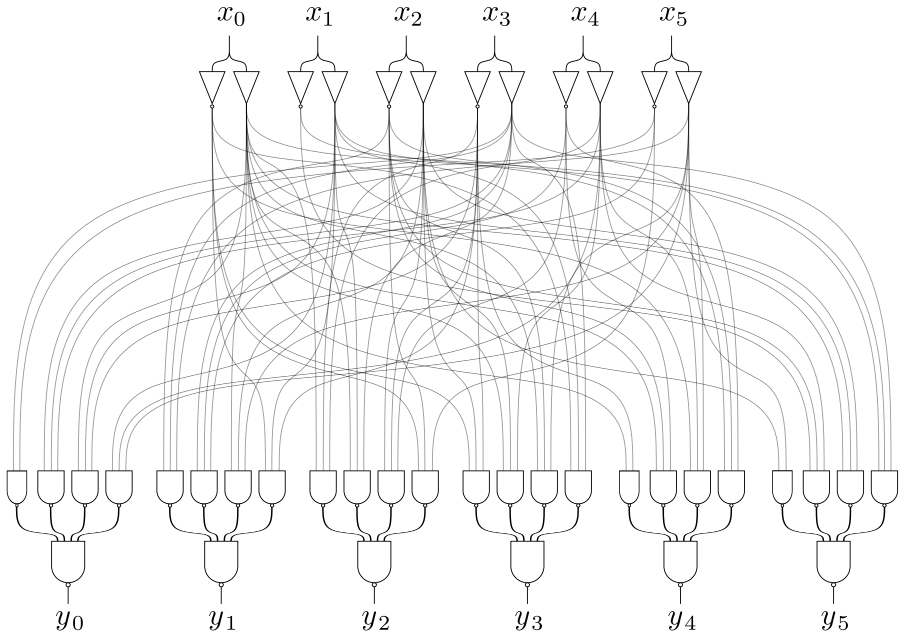


Figure 2: Implementation of the 6-bit S-box of `SPEEDY` based on two-level NAND trees.

the 2-level NAND gates.

$$
\begin{aligned}
y_0 &= (x_3 \wedge \neg x_5) \vee (x_3 \wedge x_4 \wedge x_2) \vee (\neg x_3 \wedge x_1 \wedge x_0) \vee (x_5 \wedge x_4 \wedge x_1),\\
y_1 &= (x_5 \wedge x_3 \wedge \neg x_2) \vee (\neg x_5 \wedge x_3 \wedge \neg x_4) \vee (x_5 \wedge x_2 \wedge x_0) \vee (\neg x_3 \wedge \neg x_0 \wedge x_1),\\
y_2 &= (\neg x_3 \wedge x_0 \wedge x_4) \vee (x_3 \wedge x_0 \wedge x_1) \vee (\neg x_3 \wedge \neg x_4 \wedge x_2) \vee (\neg x_0 \wedge \neg x_2 \wedge \neg x_5),\\
y_3 &= (\neg x_0 \wedge x_2 \wedge \neg x_3) \vee (x_0 \wedge x_2 \wedge x_4) \vee (x_0 \wedge \neg x_2 \wedge x_5) \vee (\neg x_0 \wedge x_3 \wedge x_1),\\
y_4 &= (x_0 \wedge \neg x_3) \vee (x_0 \wedge \neg x_4 \wedge \neg x_2) \vee (\neg x_0 \wedge x_4 \wedge x_5) \vee (\neg x_4 \wedge \neg x_2 \wedge x_1),\\
y_5 &= (x_2 \wedge x_5) \vee (\neg x_2 \wedge \neg x_1 \wedge x_4) \vee (x_2 \wedge x_1 \wedge x_0) \vee (\neg x_1 \wedge x_0 \wedge x_3).
\end{aligned}
$$

3.4 S-box Latency Comparison

We benchmark our chosen S-box with respect to minimum latency in hardware and compare it to a number of other S-boxes from literature in Table 4. Details about the synthesis tools and process are given in Section 7. Please note that up to now only 4-bit S-boxes have been proposed for low-latency constructions in literature, namely (in alphabetical order) the `Midori` S-boxes [BBI+15], the `Orthros` S-box [BIL+21], the `PRINCE` S-box [BCG+12] and the `QARMA` S-boxes [Ava17]. Yet, in order to compare the `SPEEDY` S-box also to larger substitution boxes we chose the `ASCON` 5-bit S-box [DEMS19], the Data Encryption Standard (DES) S_1 6-to-4-bit box (as a representative of the 8 different DES S-boxes) [oST79], the `Q2263` 6-bit S-box [BMD+20] and the Advanced Encryption

Table 4: Latency comparison of different S-boxes with varying numbers of input bits (#ib). If not stated otherwise, each S-box is implemented as a lookup table (using with/select in VHDL).

		Minimum Latency [ns]					
		Commercial Foundry				NanGate OCL	
#ib	S-box	90 nm LP	65 nm LP	40 nm LP	28 nm HPC	45 nm	15 nm
4	`Midori Sb`$_0$	0.089098	0.070579	0.055577	0.021051	0.111156	0.010619
4	`Midori Sb`$_1$	0.132489	0.095724	0.080657	0.026898	0.119637	0.009058
4	`Orthros`	0.075344	0.051435	0.055908	0.021003	0.133932	0.008821
4	`PRINCE`	0.087938	0.066545	0.052826	0.031010	0.126588	0.010176
4	`QARMA` σ_0	0.090568	0.057602	0.051993	0.022180	0.128350	0.009409
4	`QARMA` σ_1	0.144465	0.101487	0.077186	0.031306	0.156462	0.011272
4	`QARMA` σ_2	0.100530	0.075846	0.081528	0.036485	0.154379	0.013354
5	`ASCON`	0.197794	0.151025	0.123356	0.057595	0.210599	0.019854
6	`DES S`$_1$	0.260286	0.190725	0.153514	0.069299	0.309009	0.030846
6	`OAIU8L24`	0.138926	0.111734	0.088775	0.046295	0.215628	0.017971
6	`Q2263`	0.233256	0.171537	0.157194	0.068870	0.246198	0.028648
6	min(`RU8L24`)	0.220168	0.144777	0.126819	0.060535	0.240982	0.026696
6	`SPEEDY`	0.106872	0.081330	0.065966	0.029890	0.161653	0.016124
6	`SPEEDY` *	0.096468	0.073253	0.064215	0.029470	0.138825	0.012799
6	`SPEEDY_INV`	0.207746	0.152161	0.129433	0.071523	0.278395	0.025665
8	`AES`	0.407332	0.304098	0.248914	0.130490	0.491570	0.048258

* = Optimized HDL code with direct instantiation of library cells based on Figure 2.

Standard (AES) 8-bit S-box [oST01] for the comparison. Under the abbreviation `OAIU8L24` we have listed a 6-bit S-box built from two levels of OAI22 gates with uniformity 8 and linearity 24 (same properties as the SPEEDY S-box). By min(`RU8L24`) we denote the minimum latency achieved among 10 randomly generated 6-bit S-boxes with uniformity 8 and linearity 24 (without focusing on a particularly efficient implementation). Finally, the inverse of the `SPEEDY` S-box is included. However, this inverse is not required for the `SPEEDY` encryption and therefore only relevant for the latency of its decryption. Minimizing the decryption's latency is not a focus of this work.

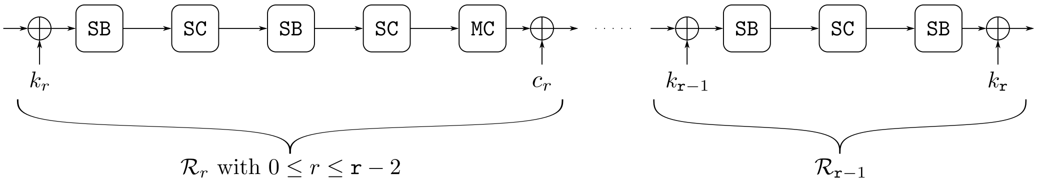
From the comparison it becomes clear that the `SPEEDY` S-box is impressively fast in hardware. It is much faster than any other S-box with more than 4 input bits (#ib), especially when considering the optimized version with direct instantiation of standard cells in the code based on Figure 2. Additionally, it even outperforms multiple of the 4-bit low-latency S-boxes (including `Midori Sb`$_1$, `QARMA` σ_1 and `QARMA` σ_2). This is a great result, since the `SPEEDY` S-box not only provides better diffusion in general but also offers stronger protection against linear and differential attacks than any 4-bit S-box possibly could. Thus, we are confident in our S-box choice as the centerpiece for an ultra low-latency cipher.

4 Specification of `SPEEDY`

`SPEEDY` is a family of ultra low-latency block ciphers with different block and key sizes, and varying numbers of rounds. Precisely, `SPEEDY-r-`6ℓ is an instance of this family with block and key size 6ℓ bits and it iterates over `r` rounds.

The internal state is viewed as an $\ell \times 6$ rectangle array of bits. We use the notation $x_{[i,j]}$ to denote the bit located at row i and column j of the state x with $0 \leq i < \ell$ and $0 \leq j < 6$. It is important to emphasize that in the remainder of this paper, all the indices start from zero and the zero-th bit or word is always considered the most significant one. Besides, note that if there is an addition or a subtraction in the indices of the state, it is always in modulo ℓ for the first (row) index and in modulo 6 for the second (column) index.

Initialization: The cipher receives a 6ℓ-bit plaintext and initializes the internal state with it using the same order used for indexing bits, i.e. it first fills $x_{[0,0]}$, then $x_{[0,1]}$ and so on. Then, $\mathtt{r}$ round functions, $\mathcal{R}_r$ (with $0 \le r < \mathtt{r}$), are applied on the internal state, the first $\mathtt{r}-1$ ones of which (up to the round keys and round constants) are identical. Each round function is composed of the following four different operations: (2×) `SubBox`, (2×) `ShiftColumns`, `MixColumns`, `AddRoundConstant` and `AddRoundKey`. Considering $x \in \mathbb{F}_2^{\ell\times 6}$ as the input, $y \in \mathbb{F}_2^{\ell\times 6}$ as the output of operations, $0 \le i < \ell$ and $0 \le j < 6$, the round operations are defined as follows:


SB
SC
SB
SC
MC
SB
SC
SB
k_r
c_r
$k_{\mathtt{r}-1}$
$k_{\mathtt{r}}$
$\mathcal{R}_r$ with $0 \le r \le \mathtt{r}-2$
$\mathcal{R}_{\mathtt{r}-1}$

- **`SubBox` (`SB`):** The 6-bit S-box S is applied to each row of the state.

  $$(y_{[i,0]}, y_{[i,1]}, y_{[i,2]}, y_{[i,3]}, y_{[i,4]}, y_{[i,5]}) = S(x_{[i,0]}, x_{[i,1]}, x_{[i,2]}, x_{[i,3]}, x_{[i,4]}, x_{[i,5]})\,, \quad \forall\, i\,.$$

  The table for the S-box (in hexadecimal notation) is given in Table 3 and its implementation based on two-level NAND trees is shown in Figure 2.

- **`ShiftColumns` (`SC`):** The j-th column of the state is rotated upside by j bits.

  $$y_{[i,j]} = x_{[i+j,j]}\,, \quad \forall\, i,j\,.$$

- **`MixColumns` (`MC`):** A cyclic binary matrix is multiplied to each column of the state.

  $$y_{[i,j]} = x_{i,j} \oplus x_{[i+\alpha_1,j]} \oplus x_{[i+\alpha_2,j]} \oplus x_{[i+\alpha_3,j]} \oplus x_{[i+\alpha_4,j]} \oplus x_{[i+\alpha_5,j]} \oplus x_{[i+\alpha_6,j]}\,, \quad \forall\, i,j\,.$$

  For simplicity, we identify the applied matrix with $\alpha = (\alpha_1, \ldots, \alpha_6)$ that is parameterized for each version of the cipher with different ℓ value.

- **`AddRoundKey` ($\mathtt{A}_{k_r}$):** The 6ℓ-bit round key k_r is XORed to the whole of the state.

  $$y_{[i,j]} = x_{[i,j]} \oplus k_{r\,[i,j]}\,, \quad \forall\, i,j\,.$$

- **`AddRoundConstant` ($\mathtt{A}_{c_r}$):** The 6ℓ-bit constant c_r is XORed to the whole of the state.

  $$y_{[i,j]} = x_{[i,j]} \oplus c_{r\,[i,j]}\,, \quad \forall\, i,j\,.$$

  Similar to `PRINCE`, the round constants are chosen as the binary digits of the number $\pi - 3 = 0.1415\ldots$. Table 5 presents the first 100×64 bits of this constant. We use the first 6ℓ bits as c_0, the second 6ℓ bits as c_1 and so on.

Round Function: Using the above mentioned round operations, the first $\mathtt{r}-1$ round functions (with $0 \le r \le \mathtt{r}-2$) are defined as

$$\mathcal{R}_r = \mathtt{A}_{c_r} \circ \mathtt{MC} \circ \mathtt{SC} \circ \mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB} \circ \mathtt{A}_{k_r}\,,$$

while in the last round, the linear layer and constant addition are omitted, and instead an extra key addition is applied, i.e.,

$$\mathcal{R}_{\mathtt{r}-1} = \mathtt{A}_{k_{\mathtt{r}}} \circ \mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB} \circ \mathtt{A}_{k_{\mathtt{r}-1}}\,.$$

Table 5: The first 100×64 bits of the constant used in the round constants of `SPEEDY`.

0	243f6a8885a308d3	13198a2e03707344	a4093822299f31d0	082efa98ec4e6c89
1	452821e638d01377	be5466cf34e90c6c	c0ac29b7c97c50dd	3f84d5b5b5470917
2	9216d5d98979fb1b	d1310ba698dfb5ac	2ffd72dbd01adfb7	b8e1afed6a267e96
3	ba7c9045f12c7f99	24a19947b3916cf7	0801f2e2858efc16	636920d871574e69
4	a458fea3f4933d7e	0d95748f728eb658	718bcd5882154aee	7b54a41dc25a59b5
5	9c30d5392af26013	c5d1b023286085f0	ca417918b8db38ef	8e79dcb0603a180e
6	6c9e0e8bb01e8a3e	d71577c1bd314b27	78af2fda55605c60	e65525f3aa55ab94
7	5748986263e81440	55ca396a2aab10b6	b4cc5c341141e8ce	a15486af7c72e993
8	b3ee1411636fbc2a	2ba9c55d741831f6	ce5c3e169b87931e	afd6ba336c24cf5c
9	7a32538128958677	3b8f48986b4bb9af	c4bfe81b66282193	61d809ccfb21a991
10	487cac605dec8032	ef845d5de98575b1	dc262302eb651b88	23893e81d396acc5
11	0f6d6ff383f44239	2e0b4482a4842004	69c8f04a9e1f9b5e	21c66842f6e96c9a
12	670c9c61abd388f0	6a51a0d2d8542f68	960fa728ab5133a3	6eef0b6c137a3be4
13	ba3bf0507efb2a98	a1f1651d39af0176	66ca593e82430e88	8cee8619456f9fb4
14	7d84a5c33b8b5ebe	e06f75d885c12073	401a449f56c16aa6	4ed3aa62363f7706
15	1bfedf72429b023d	37d0d724d00a1248	db0fead349f1c09b	075372c980991b7b
16	25d479d8f6e8def7	e3fe501ab6794c3b	976ce0bd04c006ba	c1a94fb6409f60c4
17	5e5c9ec2196a2463	68fb6faf3e6c53b5	1339b2eb3b52ec6f	6dfc511f9b30952c
18	cc814544af5ebd09	bee3d004de334afd	660f2807192e4bb3	c0cba85745c8740f
19	d20b5f39b9d3fbdb	5579c0bd1a60320a	d6a100c6402c7279	679f25fefb1fa3cc
20	8ea5e9f8db3222f8	3c7516dffd616b15	2f501ec8ad0552ab	323db5fafd238760
21	53317b483e00df82	9e5c57bbca6f8ca0	1a87562edf1769db	d542a8f6287effc3
22	ac6732c68c4f5573	695b27b0bbca58c8	e1ffa35db8f011a0	10fa3d98fd2183b8
23	4afcb56c2dd1d35b	9a53e479b6f84565	d28e49bc4bfb9790	e1ddf2daa4cb7e33
24	62fb1341cee4c6e8	ef20cada36774c01	d07e9efe2bf11fb4	95dbda4dae909198

Key Schedule: The cipher receives a 6ℓ-bit master key and initializes it to the state of the zero-th round key (k_0). Then, it applies the bit permutation `PB` to compute the next round key, i.e., using the following permutation P, the positions of the bits are changed. That is

$$k_{r+1} = \texttt{PB}(k_r) \quad \text{with} \quad k_{r+1\,[i',j']} = k_{r\,[i,j]}\,,$$

such that

$$(i',j') := P(i,j) \quad \text{with} \quad (6i'+j') \equiv \big(\beta \cdot (6i+j) + \gamma\big) \bmod 6\ell\,,$$

i.e., i' and j' are the quotient and remainder of dividing $\big(\beta \cdot (6i+j) + \gamma\big) \bmod 6\ell$ to 6, respectively. The parameters β and γ are dependent on the block length of the cipher with the condition of $\gcd(\beta, 6\ell) = 1$.

Instantiation: As already mentioned, `SPEEDY` is a family of block ciphers that allows instantiations of a wide range of block sizes and security levels. One may choose the block size of the encryption (6ℓ) by to the type of data blocks that need to be encrypted, and select the number of rounds (`r`) based on the necessary security level. By applying an appropriate $\alpha = (\alpha_1, \ldots, \alpha_6)$ value with regards to the rationale explained in Section 5, `SPEEDY-r-`6ℓ is ready to use.

To provide encryption of 64-bit blocks, which is the common instruction and data width in modern CPUs, we suggest to instantiate `SPEEDY-r-192` with $\alpha = (1, 5, 9, 15, 21, 26)$ as the linear layer's parameter. We leave the number of rounds to be chosen based on the required security level. That is, for 128- and 192-bit security levels, we recommend using $\texttt{r} \geq 6$ and $\texttt{r} \geq 7$ rounds, respectively. More details about our security claims are provided

Table 6: P bit-permutation for `SPEEDY-r-192` with $\ell = 32$, $\beta = 7$ and $\gamma = 1$.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
$P(i)$	1	8	15	22	29	36	43	50	57	64	71	78	85	92	99	106	113	120	127	134	141	148	155	162
i	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$P(i)$	169	176	183	190	5	12	19	26	33	40	47	54	61	68	75	82	89	96	103	110	117	124	131	138
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71
$P(i)$	145	152	159	166	173	180	187	2	9	16	23	30	37	44	51	58	65	72	79	86	93	100	107	114
i	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
$P(i)$	121	128	135	142	149	156	163	170	177	184	191	6	13	20	27	34	41	48	55	62	69	76	83	90
i	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
$P(i)$	97	104	111	118	125	132	139	146	153	160	167	174	181	188	3	10	17	24	31	38	45	52	59	66
i	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
$P(i)$	73	80	87	94	101	108	115	122	129	136	143	150	157	164	171	178	185	0	7	14	21	28	35	42
i	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167
$P(i)$	49	56	63	70	77	84	91	98	105	112	119	126	133	140	147	154	161	168	175	182	189	4	11	18
i	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
$P(i)$	25	32	39	46	53	60	67	74	81	88	95	102	109	116	123	130	137	144	151	158	165	172	179	186

below. The security analysis and the implementation of this instance are discussed in Section 6 and Section 7, respectively. Furthermore, for this instance we suggest to use $\beta = 7$ and $\gamma = 1$ for the key schedule parameters that the corresponding permutation P (given in Table 6) receives.
We provide several test vectors for `SPEEDY-r-192` encryption in Appendix G.

Security Claim While `SPEEDY` can be instantiated with different block and key sizes, the default is 192 bit as it constitutes the least common multiple of 6 (our S-box width) and 64 (the instruction width in high-end CPUs). We expect that `SPEEDY-r-192` achieves 128-bit security when iterated over $\mathtt{r} = 6$ rounds and full 192-bit security when iterated over $\mathtt{r} = 7$ rounds, while the $\mathtt{r} = 5$ round variant already provides a decent security level that is sufficient for many practical applications ($\geq 2^{128}$ time complexity when data complexity is limited to $\leq 2^{64}$). Compared to the security claims made for example for `PRINCE` ($\geq 2^{127-n}$ time complexity when data complexity is limited to $\leq 2^{n}$) or `PRINCEv2` ($\geq 2^{112}$ time complexity when data complexity is limited to $\leq 2^{50}$) the security level claimed by `SPEEDY-5-192` is already superior.

5 Design Rationale

The primary criterion for the design of `SPEEDY` is to use round operations with a low latency that still provide good enough cryptographic properties to provide a secure encryption with a small number of rounds. To achieve this goal, we applied the ultra low-latency S-box found in Section 3. While the design approach for the S-box is described in Section 3, all details regarding the design choices for the other round operations are explained in the following.

`MixColumns`: It is clear that the latency cost (in terms of XOR gate depth) of XORing n bits, i.e., $x_0 \oplus \ldots \oplus x_{n-1}$ is equal to $d = \lfloor \log_2 n \rfloor$. This means that XORing n bits with $2^{d-1} < n \leq 2^d$, has the same cost for all n values with respect to the latency of the circuit (considering identical topology). Therefore, to use the maximum capacity of the given latency, it is prudent to choose $n = 2^d$.

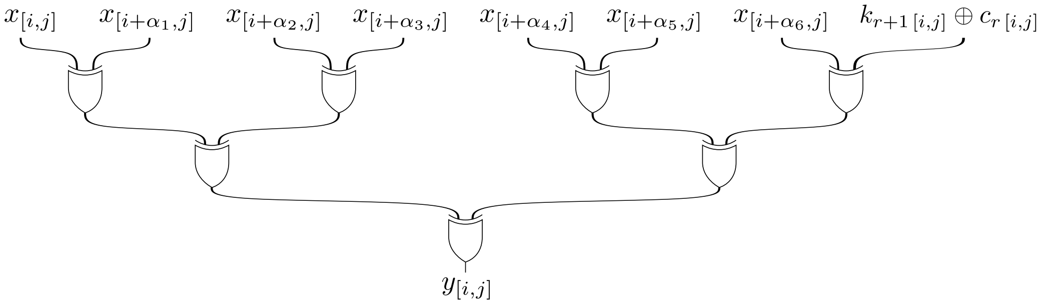


Figure 3: Implementation of each output bit of the merged function $\mathtt{A}_{k_{r+1}} \circ \mathtt{A}_{c_r} \circ \mathtt{MC}$ of the `SPEEDY` design.

In the design of `SPEEDY`, since the $\mathtt{A}_{k_{r+1}}$ operation from round $r+1$ occurs right after the $\mathtt{A}_{c_r}$ and `MC` operations from the r-th round, it is possible to merge all three operations. Considering that x and y from $\mathbb{F}_2^{\ell \times 6}$ are the input and output of the merged $\mathtt{A}_{k_{r+1}} \circ \mathtt{A}_{c_r} \circ \mathtt{MC}$ operation, respectively, then each output bit can be calculated as

$$y_{[i,j]} = x_{[i,j]} \oplus x_{[i+\alpha_1,j]} \oplus x_{[i+\alpha_2,j]} \oplus x_{[i+\alpha_3,j]} \oplus x_{[i+\alpha_4,j]} \oplus x_{[i+\alpha_5,j]} \oplus x_{[i+\alpha_6,j]} \oplus \left(k_{r+1\,[i,j]} \oplus c_{r\,[i,j]}\right).$$

Hence, it is possible to implement the whole $\mathtt{A}_{k_{r+1}} \circ \mathtt{A}_{c_r} \circ \mathtt{MC}$ as a merged function within three XOR gate levels. Note that since the input $k_{r+1\,[i,j]}$ is not in the critical path of the circuit, $k_{r+1\,[i,j]}$ and $c_{r\,[i,j]}$ can be combined with each other beforehand. Depending on the value of the round constant bit, we actually only need to use $k_{r+1\,[i,j]}$ itself or its inverted value $\neg k_{r+1\,[i,j]}$. Figure 3 depicts the corresponding circuit to implement each output bit of the merged function. Please note that the fan-out of each XOR gate in this circuit is 1. It is important to consider that for CMOS technologies where the XNOR gate is significantly faster than the XOR gate (such as NanGate 45 nm), it is easily possible to implement this linear layer with only XNOR gates instead of XORs and simply exchange the buffers and inverters of the next S-box stage to revert its inverted output.

For the `MC` operation, we decided to use the same binary cyclic matrix with polynomial representation of $1 + z^{\alpha_1} + \ldots + z^{\alpha_{w-1}}$ and multiply it with each column of the state. Therefore, each output bit of the `MC` operation is the XOR of w input bits. As explained above, the optimal choices for w are 3, 7, 15 and so on, so that it is possible to implement the above mentioned merged function with 2, 3, 4 XOR gate levels, respectively. While in `PRINCE`, `MIDORI` and `QARMA` block ciphers, this technique of merging is used by applying cyclic matrices of $w = 3$ and repeated after each S-box layer, we found that it is a good trade-off to use cyclic matrices with $w = 7$, but only after each second S-box layer, which is effectively cheaper from a latency cost perspective.

For each `SPEEDY-r-6`ℓ version of the cipher, we need to find a bijective $\ell \times \ell$ binary cyclic matrix M with polynomial representation of $1 + z^{\alpha_1} + \ldots + z^{\alpha_6}$. Finding an appropriate bijective cyclic matrix with $w = 7$ being an odd integer, is quite possible for wide range of ℓ. But, since the value of $\alpha = (\alpha_1, \ldots, \alpha_6)$ is always dependent on the value of ℓ, we leave it as a parameter of the cipher's instantiation.

Since, the probability of M being a non-singular matrix is high, we can add extra criteria regarding the choice of the α parameter.

- All values for α_1, $\alpha_2 - \alpha_1$, $\alpha_3 - \alpha_2$, $\alpha_4 - \alpha_3$, $\alpha_5 - \alpha_4$, $\alpha_6 - \alpha_5$ and $\ell - \alpha_6$ need to be smaller or equal to 6. The reason for this criterion is explained later, in the corresponding paragraph for `ShiftColumns`. Note that this criterion is only possible for $\ell \leq 42$.

- Maximum branch number: Branch number of a matrix is defined as

  $$bn := \min_{x \in \mathbb{F}_2^\ell \setminus \{0\}} \mathrm{hw}(x) + \mathrm{hw}(M \times x^T)\,,$$

  where hw denotes the Hamming weight of a binary array. In case of a bijective $\ell \times \ell$ binary cyclic matrix M with polynomial representation of $1 + z^{\alpha_1} + \ldots + z^{\alpha_{w-1}}$, the branch number cannot be higher than $w+1$. In our case, we restrict the choice of the α parameter to the ones which provide maximum branch number, i.e., 8.

- For the corresponding matrix M of parameter $\alpha = (\alpha_1, \ldots, \alpha_6)$, we build a binary table H such that the element in the position (i, j) is $\mathtt{1}$, if and only if there is an $x \in \mathbb{F}_2^\ell \setminus \{0\}$ with $\mathrm{hw}(x) = i$ and $\mathrm{hw}(M \times x^T) = j$. Then, we compute the following three numbers:

  $$\begin{aligned} bn_3 &= \min_{\substack{i_1, i_2, i_3 \\ H[i_1][i_2] = H[i_2][i_3] = \mathtt{1}}} i_1 + i_2 + i_3\,, \\ bn_4 &= \min_{\substack{i_1, i_2, i_3, i_4 \\ H[i_1][i_2] = H[i_2][i_3] = H[i_3][i_4] = \mathtt{1}}} i_1 + i_2 + i_3 + i_4\,, \\ bn_5 &= \min_{\substack{i_1, i_2, i_3, i_4, i_5 \\ H[i_1][i_2] = H[i_2][i_3] = H[i_3][i_4] = H[i_4][i_5] = \mathtt{1}}} i_1 + i_2 + i_3 + i_4 + i_5\,. \end{aligned} \tag{5}$$

  As explained later in Section 6, larger values for bn_r lead to a stronger resistance of the r-round `SPEEDY` against differential and linear attacks. Therefore, for all the possible choices of α which are meeting the first two criteria, we compute the above bn_r numbers and choose one of the corresponding α values which leads to the maximum bn_r values.

  It is noteworthy that the branch number bn is the same as bn_2 defined as

  $$bn_2 = \min_{\substack{i_1, i_2 \\ H[i_1][i_2] = \mathtt{1}}} i_1 + i_2\,.$$

  Besides, bn_r with $r > 2$ can be considered as an extension for the definition of branch number, and hereafter, we will call it a *higher-order branch number*.

In the case of `SPEEDY-r-192`, with $\ell = 32$, we applied the above criteria and end up with 30 choices from which we choose the first one that is $\alpha = (1, 5, 9, 15, 21, 26)$ with $bn_3 = 13$, $bn_4 = 20$, and $bn_5 = 25$. It is important to mention that the corresponding matrix for inverse of the `MC` operation is a cyclic matrix with $w = 19$ and $\alpha^{-1} = (4, 5, 6, 7, 10, 12, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 28)$.

`ShiftColumns`: The existence of the first `SC` operation, right after the first `SB` makes it possible that input bits of each S-box in the second `SB` operation are all from the outputs of different S-boxes of the first `SB` operation. Therefore, since the applied S-box has the full diffusion property (in both straight and inverse direction), each output bit of $\mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB}$ is a function of 36 consecutive input bits. Namely, for $\mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB}$, the output bit in the position $[i, j]$ is a function of all input bits in the position of the form $[i + p, q]$ with $0 \leq p, q < 6$, while for $(\mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB})^{-1}$, the output bit $[i, j]$ is a function of all input bits of the form $[i - p, q]$.
By considering the first criterion for `MixColumns`, namely that α_1, $\alpha_2 - \alpha_1$, $\alpha_3 - \alpha_2$, $\alpha_4 - \alpha_3$, $\alpha_5 - \alpha_4$, $\alpha_6 - \alpha_5$ and $\ell - \alpha_6$ are all smaller or equal to 6, it means that the output bit of $\mathtt{MC} \circ \mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB}$ and equivalently, output of one *key-less round function* $\mathtt{MC} \circ \mathtt{SC} \circ \mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB}$ is dependent on the whole 6ℓ input bits. The same holds for $(\mathtt{MC} \circ \mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB})^{-1}$ in the decryption side, hence, the input of one key-less round function is dependent on the whole 6ℓ output bits.

Moreover, the same arguments hold for inserting the second `SC`, right after the second `SB` operation, which means that each output bit of `SB`∘`MC`∘`SC`∘`SB` depends on the whole 6ℓ input bits which equivalently holds for the *rotated key-less round function* `SC` ∘ `SB` ∘ `MC` ∘ `SC` ∘ `SB`. Altogether, one key-less round function or rotated round function, in both encryption and decryption directions, provides full diffusion. In other words, in a key recovery attack, to compute one output bit of those functions, the attacker needs to know the value of the whole input state. Note that knowing the value for the whole input state of these functions requires knowing the whole state of the round key. This means, if the attacker wants to extend a distinguisher by appending one complete round (or rotated round) function, to do a key recovery attack, he needs to guess the whole 6ℓ bits of the key.
It is important to mention that since existence of any key-independent linear operation right before the ciphertext does not add any security to the encryption, we exclude the `MC` and the second `SC` operations from the last round.

Key Schedule: Since the main target of our design is to provide a low-latency encryption routine, and since other cost factors of the implementation such as area or energy consumption of the circuits are only secondary priorities, one can apply a key schedule built from costly operations. Yet, since we do not aim for related-key security, and since the round function has a strong diffusion, we found that using a linear key schedule is sufficient for our purposes. Besides, updating round keys by a bit-permutation function in an unrolled implementation has no latency, area or energy costs, thus we decided to use such a key schedule. Furthermore, we wanted to use a bit-permutation such that it is easy to generalize for all `SPEEDY-r-`6ℓ members. To do so, we chose the general affine mapping in the finite integer field of $\{0, \ldots, 6\ell - 1\}$, that the permutation P maps x, an element of this field, to $P(x) = \beta x + \gamma \mod 6\ell$. The only requirement for P being a bijection is that β and 6ℓ need to be co-prime, i.e., $\gcd(\beta, 6\ell) = 1$.

6 Security Analysis

In this section, we provide details about the cryptographic properties of the `SPEEDY` family of block ciphers. We start with differential, linear and algebraic properties of the S-box S and expand them over a round function of the cipher. By applying properties for the round function, we discuss the security of an `r` round structure of `SPEEDY`.

Cryptographic Properties of the S-box: The S-box S, presented in Section 3, is the heart of the `SPEEDY` design and it needs to be studied in detail. As described before the uniformity and linearity of S is equal to 8 and 24, respectively. This means that the maximum probability of differentials over S is $8 \cdot 2^{-6} = 2^{-3}$ and the maximum absolute correlation of linear approximations is $24 \cdot 2^{-6} = 3 \cdot 2^{-3}$ (equally means that the maximum potential of linear approximations is $(3 \cdot 2^{-3})^2 = 9 \cdot 2^{-6} \approx 2^{-2.83}$). As one important part of the Differential Distribution Table (DDT) and Linear Approximation Table (LAT), we present the 1-bit to 1-bit differentials and linear approximations in Appendix B, Table 10. In more detail, entry (i, j) of the 1-bit to 1-bit DDT denotes the probability that having only one active bit in the position i of the S-box inputs leads to only one active bit in the position j of the S-box output. In case of 1-bit to 1-bit LAT, entry (i, j) of the table denotes the absolute correlation value for the $x_i = y_j$ linear approximation.
Even though, one of the criteria for building the low-latency S-box was to provide full dependency of the output bits on the input bits, this is not sufficient to provide all information about algebraic properties of the function. We provide the algebriac normal form (ANF) representation of both S and S^{-1} in Appendix C. As shown, not only all the input/output variables are non-linearly involved in all the output/input coordinates (i.e., the S-box provides full diffusion in both straight and inverse directions), each coordinate

function is quite dense with respect to the number of involved terms. Another interesting information is that the ANF degree for coordinates of S is 5, 3, 3, 3, 4 and 5, respectively, while in the case of S^{-1}, these numbers are 5, 4, 5, 4, 5 and 5, respectively (cf. Appendix C).

Cryptographic Properties of `SB` $\circ$ `SC` $\circ$ `SB`: Since in the round function of `SPEEDY`, two `SB` operations are connected through the `SC` operation which is a simple bit permutation, it is necessary to look at the properties of this combination. We first investigate the 1-bit to 1-bit differentials and linear approximations of `SB` $\circ$ `SC` $\circ$ `SB`. Since each input bit of the second `SB` operation comes from a different first-stage S-box, 1-bit to 1-bit transitions over `SB` $\circ$ `SC` $\circ$ `SB` are possible if and only if the transitions over the first and second `SB` operations, both are 1-bit to 1-bit transitions. Besides, without any extra assumption (such as independency between the state bits), it can be proven that probability or correlation of this 1-bit to 1-bit transitions over `SB` $\circ$ `SC` $\circ$ `SB` is the multiplication of probabilities or correlations over two active S-boxes (one from the first `SB` and another from the second `SB` operation).

Since `SC` does not change the column position of active bits, it is easily possible to compute these probabilities. Appendix B, Table 11 presents the 1-bit to 1-bit differential probabilities and linear correlations over `SB` $\circ$ `SC` $\circ$ `SB` such that entry $[i, j]$ denotes the maximum possible probabilities or linear correlations that an active input bit in the column i transits to an active output bit in the column j. To compute these values, we used the following equation which T_1 and T_2 denote the Table 10 and Table 11, respectively.

$$T_2[i, j] = \max_k \; T_1[i, k] \cdot T_1[k, j] \,.$$

Note that the maximum entry for differential transitions is 2^{-6} and for linear transitions it is $15 \cdot 2^{-7} \approx 2^{-3}$. We are only interested in 1-bit to 1-bit transitions, because the probability or the correlation of such transitions are among the highest ones and also because based on such transitions, we can build differential or linear characteristics with a high differential probability or linear correlation.

Again due to the fact that `SC` does not change the column position of the bits and each input bit of the second `SB` is the output of a different S-box, it is possible to compute the algebraic degree of `SB` $\circ$ `SC` $\circ$ `SB`. The degree of any output bit in the columns 0, 1, … and 5 is equal to 19, 15, 13, 13, 13 and 20, respectively.

It is important to mention that replacing the current S-box with another bit-permutation equivalent S-box will change differential, linear and algebraic properties of `SB` $\circ$ `SC` $\circ$ `SB`. While in Section 3, we ended up with a bit-permutation equivalency class of S-boxes, we tried all the S-boxes of this class to find an S-box such that the maximum entry in Table 11 and also the number of entries with maximum value are as small as possible. Moreover, we want the minimum algebraic degree over `SB` $\circ$ `SC` $\circ$ `SB` coordinates to be as large as possible. Note that due to the structure of the round function, since encryption with S-box $P_{out} \circ S \circ P_{in}$ is identical to encryption with S-box $P_{in} \circ P_{out} \circ S$ (up to a column permutation in the state of plaintext, ciphertext, round key and round constants), we can consider one of them to be the identity bit-permutation and only need to choose the other one.

Differential and Linear Attacks Since there are 1-bit to 1-bit differential and linear approximations over `SB` $\circ$ `SC` $\circ$ `SB` and the corresponding probability or correlation of those transitions are quite significant, it is necessary to choose a strong `MC` operation. The criterion of having branch number $bn = 8$ ensures that the maximum expected differential probability (EDP) of differential trails and the maximum expected linear potential (ELP) of linear trails over two rounds of `SPEEDY` is equal to $(2^{-6})^8 = 2^{-48}$.

To discuss the resistance of r-round `SPEEDY`, we use the *higher-order* branch number bn_r defined in Equation 5 to have an overview about the minimum number of active S-boxes in differential or linear trails. Therefore, using this estimation the maximum EDP of differentials and the ELP of linear trails over r-round `SPEEDY` is estimated by $2^{-6\cdot bn_r}$. In case of `SPEEDY-r-192`, with the recommended α parameter, we have

$$bn_3 = 13\,, \quad bn_4 = 20\,, \quad bn_5 = 25\,, \quad bn_6 = 32\,.$$

Hence, we estimate that EDP (resp. ELP) of any differential (resp. linear) trails over 3, 4, 5 and 6 rounds is smaller than 2^{-78}, 2^{-120}, 2^{-150} and 2^{-192}. Actually, assuming that all the 1-bit to 1-bit differential or linear transitions through the S-box are possible, and by considering that there are at most 8 active words (of 6-bit) per state of operations, we searched for the minimum number of active S-boxes. We found that this number is 13, 23 and 35 for 2, 3 and 4 rounds. Assuming that all these 1-bit to 1-bit transitions occur with differential probability (or linear potential) of 2^{-3}, the EDP (resp. ELP) of any differential (resp. linear) trails over 2, 3 and 4 rounds is smaller than 2^{-39}, 2^{-69} and 2^{-105}. We emphasize that these values are an upper bound, which means that a trail with such EDP or ELP must not necessarily exist.

Higher-Order Differential, Integral and Cube Attacks `SPEEDY`'s round function has a strong diffusion and high algebraic degree. While, we investigate these properties for one complete round precisely, for a larger number of rounds, we expect that the ANF representation would be dense with respect to the number of involved terms. Therefore, we believe that these attacks are weaker than differential and linear attacks and less of a concern.

Number of Rounds For a low-latency block cipher, a large security margin is not reasonable and is usually considered as wasteful. Since the attacker cannot add more than one round to extend a distinguisher and therefore to use the distinguisher in a key recovery attack, we believe a security margin of one round is sufficient. Therefore, we recommend to choose the number of rounds with respect to the required security level of the block cipher's application. For example, in case of the `SPEEDY-r-192` instance, we recommend to use `SPEEDY-6-192` and `SPEEDY-7-192` for 128-bit and 192-bit security levels, respectively, while for more practical applications, such as a security level of 2^{128} time and 2^{64} data complexity, we recommend to use `SPEEDY-5-192`.

Further Security Analysis Additional security analysis results with respect to impossible differential, zero-correlation linear-hull, meet-in-the-middle and implementation attacks can be found in Appendix D.

7 Hardware Implementation

In this section, we analyze the minimum achievable latency of fully-unrolled `SPEEDY` hardware implementations as well as the area required for the time-constrained circuits and compare them to a number of other cryptographic primitives that have been suggested for high-speed single-cycle encryption in literature. Implementing `SPEEDY` in hardware is rather straightforward since almost all round operations which require any logic and may not be realized through wiring alone are already chosen as circuit representations. In detail, Figure 2 shows the hardware circuitry for the 6-bit high-speed S-box while Figure 3 depicts the logic circuit that implements the combined $\mathtt{A}_{k_{r+1}} \circ \mathtt{A}_{c_r} \circ \mathtt{MC}$ function. The `ShiftColumns` operation does not require any logic, which means that only the initial and the final `AddRoundKey` functions remain. Obviously these are implemented with a single stage of regular XOR gates.

Table 7 presents the minimum latency results achieved for different instances of `Gimli`, `MANTIS`, `Midori`, `Orthros`, `PRINCE`, `PRINCEv2`, `QARMA`, and `SPEEDY` (in alphabetical order).

Table 7: Minimum latency of fully-unrolled encryption-only circuits of different cryptographic primitives.

	Minimum Latency [ns]					
	Commercial Foundry				NanGate OCL	
Cipher	90 nm LP	65 nm LP	40 nm LP	28 nm HPC	45 nm	15 nm
`Gimli E-M`	4.532467	3.330192	2.794736	1.178424	4.537304	0.435069
MANTIS$_6$	4.625529	3.405490	2.891383	1.278725	4.479773	0.437595
MANTIS$_7$	5.201681	3.722473	3.234409	1.421365	5.074452	0.492703
MANTIS$_8$	5.823127	4.233543	3.631438	1.594997	5.739020	0.552384
`Midori`	5.061255	3.582221	3.142355	1.362237	4.934847	0.481522
`Orthros`	3.862139	2.678637	2.401275	1.087139	3.774836	0.369497
`PRINCE`	4.101177	2.866749	2.521302	1.108886	4.059997	0.389144
`PRINCEv2`	4.047311	2.944367	2.509131	1.103273	4.077636	0.387146
QARMA$_5$-64-σ_0	4.075846	2.920377	2.498908	1.134901	4.014516	0.385281
QARMA$_6$-64-σ_0	4.770325	3.418600	2.951308	1.308331	4.554445	0.448931
QARMA$_7$-64-σ_0	5.449707	3.909138	3.389576	1.538606	5.336362	0.517093
QARMA$_8$-64-σ_0	6.103768	4.396543	3.814078	1.697027	5.966323	0.575525
QARMA$_5$-64-σ_1	4.515514	3.284252	2.815788	1.219624	4.367899	0.408580
QARMA$_6$-64-σ_1	5.297867	3.808675	3.271455	1.388353	4.944635	0.472798
QARMA$_7$-64-σ_1	6.014477	4.371963	3.745959	1.601572	5.800633	0.542712
QARMA$_8$-64-σ_1	6.720944	4.904521	4.202632	1.797539	6.498429	0.608985
`SPEEDY-5-192`	2.994643	2.178075	1.867064	0.847761	3.187368	0.300466
`SPEEDY-6-192`	3.637978	2.639186	2.277422	1.032206	3.848132	0.366762
`SPEEDY-7-192`	4.261928	3.087257	2.663004	1.217946	4.515505	0.431032
`SPEEDY-5-192` *	2.941130	2.121748	1.820950	0.826217	2.817971	0.290961
`SPEEDY-6-192` *	3.559981	2.573561	2.223863	1.011173	3.382270	0.353391
`SPEEDY-7-192` *	4.174183	3.029217	2.620612	1.186598	3.995325	0.413950

* = Optimized HDL code with direct instantiation of library cells based on Figures 2 and 3.

All results have been obtained by synthesizing the fully-unrolled cipher circuits between two register stages for minimum clock period using the *Synopsys Design Compiler Version O-2018.06-SP4* software while executing four stages of the `compile_ultra` command (three incremental). We have repeated the analysis with 6 different standard cell libraries, 4 of which are manufacturable cell libraries from a commercial foundry, while the remaining 2 are open-source libraries which are not manufacturable but can be used for producing universally comparable and reproducible synthesis results. Please note that `Gimli` is a key-less permutation. Therefore, in order to create an encryption circuit from the primitive we have realized it in Even-Mansour scheme [EM97] with two different keys at the beginning and end. With respect to our `SPEEDY` implementations we distinguish between results that are achieved when giving the regular behavioral (or dataflow) description of the cipher to the synthesis tool and those results we have obtained by optimizing the code and instantiating the desired standard cells directly in the HDL code (according to the gate-level descriptions shown in Figures 2 and 3). It is obvious that this optimization has a significant impact on the performance in NanGate libraries, but less of an impact in the commercial technologies. In order to force the synthesizer to use our suggested gate-level structures for `MC` and `SB` we set a `size-only` attribute on the relevant cells in *Synopsys Design Compiler* before the first `compile_ultra` command. The synthesizer then only scales the drive strengths of these cells. In a next step three `compile_ultra -incremental` commands are executed without `size-only` attribute, so that all optimizations are allowed again. With that technique the highest quality of results is achieved and the majority of manually-instantiated cells still remain unchanged.

Table 8: Area consumption of fully-unrolled encryption-only circuits of different cryptographic primitives when synthesized for minimum latency.

	Area [GE]					
	Commercial Foundry				NanGate OCL	
Cipher	90 nm LP	65 nm LP	40 nm LP	28 nm HPC	45 nm	15 nm
Gimli E-M	72644.00	82781.00	63100.50	144036.33	52038.67	57551.25
MANTIS$_6$	21045.75	23264.50	20448.25	36073.33	12660.67	15954.00
MANTIS$_7$	23229.25	26385.75	23192.50	43220.33	14225.67	17522.50
MANTIS$_8$	26365.75	30316.75	25429.75	50793.00	15663.33	19707.50
Midori	18678.50	21964.00	17562.25	41450.67	10675.33	13927.25
Orthros	49639.75	61657.00	44715.75	74384.67	31317.33	39165.00
PRINCE	16244.25	19877.75	17177.00	38145.33	9873.33	13291.00
PRINCEv2	17661.25	18798.25	16556.50	33470.33	10332.00	13069.50
QARMA$_5$-64-σ_0	19590.75	21706.75	20255.00	31703.00	11824.67	14880.75
QARMA$_6$-64-σ_0	22624.25	25349.50	22689.00	38813.67	14165.67	17621.75
QARMA$_7$-64-σ_0	25614.00	29323.00	24656.25	40494.33	15769.33	19770.25
QARMA$_8$-64-σ_0	28813.75	32780.75	28262.75	47952.33	17908.00	22074.00
QARMA$_5$-64-σ_1	20264.75	23753.00	20202.25	34302.00	12350.33	15588.75
QARMA$_6$-64-σ_1	23162.25	26941.25	23333.75	45419.00	15066.00	18164.00
QARMA$_7$-64-σ_1	26563.75	31495.00	27059.50	52108.00	16641.00	20670.25
QARMA$_8$-64-σ_1	30534.50	35787.75	29116.50	54967.00	18963.67	22761.75
SPEEDY-5-192	47364.00	53856.00	47528.50	74467.00	27903.33	34649.00
SPEEDY-6-192	57322.00	64438.25	56816.00	88932.00	34085.00	41443.25
SPEEDY-7-192	68370.00	75273.00	65422.00	95235.67	39853.33	48727.75
SPEEDY-5-192 *	49902.00	58796.25	55846.75	80313.33	29839.00	38075.25
SPEEDY-6-192 *	59688.00	70653.00	66553.00	98950.00	36523.33	46266.50
SPEEDY-7-192 *	73397.75	84745.00	77519.75	111754.33	42813.33	54193.25

* = Optimized HDL code with direct instantiation of library cells based on Figures 2 and 3.

It is obvious from Table 7 that SPEEDY-5-192 and SPEEDY-6-192 produce the smallest latencies among all implementations. The next fastest primitives are Orthros and PRINCE/PRINCEv2. Gimli, performs respectably well given its large state (384 bit) and number of rounds (24). Yet, the claim that it outperforms PRINCE by a significant margin, made in [GKD20], is very doubtful considering our results. Please note that for all ciphers except Midori we have used hardware implementations written by the original authors of the corresponding papers (Qameleon authors for QARMA).

Table 8 shows the corresponding area consumption for the fully-unrolled and highly latency constrained circuits. Clearly, SPEEDY requires a larger circuit area compared to all other ciphers except Gimli. However, this is mainly caused by its 192-bit state (which is larger than for all other ciphers in the table except Gimli). In more detail, when multiplying the area of the 64-bit ciphers by 3 (to encrypt 192 bit at once) many of them require a larger area than SPEEDY-5-192 and all MANTIS and QARMA instances even exceed the area of SPEEDY-6-192. Thus, we believe that for their block widths and the high security and performance levels that the SPEEDY instances provide, their area consumption is acceptable. Power consumption figures for all circuits are given in Appendix E, Table 12.

Because synthesis results disregard the impact of wire capacitances on the latency of hardware circuits, we have exemplarily taken all netlists generated for the 65 nm technology through a Place and Route (PnR) process in order to estimate the post-layout latencies. These are given in comparison to the pre-layout values in Table 9. Naturally, the overhead introduced by the physical layout is greater for the circuits that have a larger area footprint, e.g., Gimli, Orthros and SPEEDY, because connected cells might be wider

Table 9: Comparison of pre-layout and post-layout latencies in a commercial 65 nm CMOS technology.

	Minimum Latency [ns]		
	65 nm LP		
Cipher	Pre-Layout	Post-Layout	Overhead
`Gimli E-M`	3.330192	3.902397	17.18 %
`MANTIS`$_6$	3.405490	3.810253	11.89 %
`MANTIS`$_7$	3.722473	4.225445	13.51 %
`MANTIS`$_8$	4.233543	4.785156	13.03 %
`Midori`	3.582221	4.005088	11.80 %
`Orthros`	2.678637	3.166256	18.20 %
`PRINCE`	2.866749	3.236980	12.91 %
`PRINCEv2`	2.944367	3.324928	12.93 %
`QARMA`$_5$-64-σ_0	2.920377	3.302898	13.10 %
`QARMA`$_6$-64-σ_0	3.418600	3.869228	13.18 %
`QARMA`$_7$-64-σ_0	3.909138	4.432907	13.40 %
`QARMA`$_8$-64-σ_0	4.396543	5.078354	15.51 %
`QARMA`$_5$-64-σ_1	3.284252	3.696785	12.56 %
`QARMA`$_6$-64-σ_1	3.808675	4.294109	12.75 %
`QARMA`$_7$-64-σ_1	4.371963	4.929371	12.75 %
`QARMA`$_8$-64-σ_1	4.904521	5.519027	12.53 %
`SPEEDY-5-192`	2.178075	2.612023	19.92 %
`SPEEDY-6-192`	2.639186	3.142331	19.06 %
`SPEEDY-7-192`	3.087257	3.717537	20.42 %
`SPEEDY-5-192` *	2.121748	2.572030	21.22 %
`SPEEDY-6-192` *	2.573561	3.136378	21.87 %
`SPEEDY-7-192` *	3.029217	3.696695	22.03 %

* = Optimized HDL code with direct instantiation of library cells based on Figures 2 and 3.

apart from each other and longer wire lengths are required to connect them (also because metal utilization increases and wires have to be routed on higher, thicker metal layers). However, despite the slightly larger overhead `SPEEDY-5-192` and `SPEEDY-6-192` are still the fastest encryption primitives after PnR.
Details about the `SPEEDY` decryption and associated implementation results are provided in Appendix F.

7.1 Code and Reproducibility

A reference software implementation in C and hardware implementations of `SPEEDY-r-192` encryption and decryption in VHDL, along with synthesized netlists in NanGate libraries and associated synthesis scripts, are all available in our GitHub repository found here: https://github.com/Chair-for-Security-Engineering/SPEEDY.

8 Conclusion

In this work we have introduced `SPEEDY`, a family of ultra low-latency block ciphers developed for extremely high execution speed in CMOS hardware and dedicated to semi-custom, i.e., standard-cell-based, integrated circuit design. The primary targets for `SPEEDY` are security architectures in high-end CPUs which require ultra low-latency encryption, such as secure caches, dedicated hardware extensions, memory encryption, pointer authentication and many more. `SPEEDY` achieves higher performance than any competitor because of hardware-specific gate- and transistor-level observations that have been exploited in its design to make it extremely performant in CMOS hardware. While `SPEEDY` can be instantiated with different block and key sizes, the default is 192 bit. Based on our analysis, we are confident that 7 rounds provide full security, while 5 rounds already provide a higher security level than `PRINCE` or `PRINCEv2` for example. Our extensive evaluation of hardware implementations demonstrates that both `SPEEDY-5-192` and `SPEEDY-6-192` are faster than any proposed version of `PRINCE`, `PRINCEv2`, `MANTIS`, `QARMA`, `Midori`, `Gimli` and `Orthros`. Thus, `SPEEDY` is a significant upgrade over the state of the art for any application where area and energy are secondary design goals while high performance is the number one priority.

Acknowledgments

The work described in this paper has been supported in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and through the project 406956718 "SymmetriC CiphEr design with inherent phySical Security (SuCCESS)". Besides, S. Rasoolzadeh is supported by the Netherlands Organisation for Scientific Research (NWO) under TOP grant TOP1.18.002 SCALAR.

References

[ABP$^+$18] Victor Arribas, Begül Bilgin, George Petrides, Svetla Nikova, and Vincent Rijmen. Rhythmic keccak: SCA security and low latency in HW. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):269–290, 2018.

[Ava17] Roberto Avanzi. The QARMA block cipher family. almost MDS matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency s-boxes. *IACR Trans. Symmetric Cryptol.*, 2017(1):4–44, 2017.

[BBI$^+$15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 411–436. Springer, 2015.

[BCG$^+$12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2012.

[BEK$^+$20] Dušan Božilov, Maria Eichlseder, Miroslav Knezevic, Baptiste Lambin, Gregor Leander, Thorben Moos, Ventzislav Nikov, Shahram Rasoolzadeh, Yosuke Todo, and Friedrich Wiemer. Princev2 - more security for (almost) no overhead.

In *Selected Areas in Cryptography - SAC 2020*, Lecture Notes in Computer Science, 2020.

[BFP19] Joan Boyar, Magnus Gausdal Find, and René Peralta. Small low-depth circuits for cryptographic applications. *Cryptogr. Commun.*, 11(1):109–127, 2019.

[BIL+21] Subhadeep Banik, Takanori Isobe, Fukang Liu, Kazuhiko Minematsu, and Kosei Sakamoto. Orthros: A low-latency PRF. *IACR Trans. Symmetric Cryptol.*, 2021(1):37–77, 2021.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016.

[BKL+17] Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 299–320. Springer, 2017.

[BKN19] Dusan Bozilov, Miroslav Knezevic, and Ventzislav Nikov. Optimized threshold implementations: Minimizing the latency of secure cryptographic accelerators. In Sonia Belaïd and Tim Güneysu, editors, *Smart Card Research and Advanced Applications - 18th International Conference, CARDIS 2019, Prague, Czech Republic, November 11-13, 2019, Revised Selected Papers*, volume 11833 of *Lecture Notes in Computer Science*, pages 20–39. Springer, 2019.

[BMD+20] Begül Bilgin, Lauren De Meyer, Sébastien Duval, Itamar Levi, and François-Xavier Standaert. Low AND depth and efficient inverses: a guide on s-boxes for low-latency masking. *IACR Trans. Symmetric Cryptol.*, 2020(1):144–184, 2020.

[DEMS19] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2 submission to nist. `https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/ascon-spec-round2.pdf`, 2019. Accessed: 2021-07-02.

[DXS19] Shuwen Deng, Wenjie Xiong, and Jakub Szefer. Analysis of secure caches using a three-step model for timing-based attacks. *J. Hardware and Systems Security*, 3(4):397–425, 2019.

[EM97] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *J. Cryptol.*, 10(3):151–162, 1997.

[GIB18] Hannes Groß, Rinat Iusupov, and Roderick Bloem. Generic low-latency masking in hardware. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):1–21, 2018.

[GKD20] Santosh Ghosh, Michael E. Kounavis, and Sergej Deutsch. Gimli encryption in 715.9 psec. *IACR Cryptol. ePrint Arch.*, 2020:336, 2020.

[KHF+19] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. Spectre attacks: Exploiting speculative execution. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 1–19. IEEE, 2019.

[KNR12] Miroslav Knezevic, Ventzislav Nikov, and Peter Rombouts. Low-latency encryption - is "lightweight = light + wait"? In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*, pages 426–446. Springer, 2012.

[LKO+21] Moritz Lipp, Andreas Kogler, David Oswald, Michael Schwarz, Catherine Easdon, Claudio Canella, and Daniel Gruss. PLATYPUS: Software-based Power Side-Channel Attacks on x86. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021.

[LP07] Gregor Leander and Axel Poschmann. On the classification of 4 bit s-boxes. In Claude Carlet and Berk Sunar, editors, *Arithmetic of Finite Fields, First International Workshop, WAIFI 2007, Madrid, Spain, June 21-22, 2007, Proceedings*, volume 4547 of *Lecture Notes in Computer Science*, pages 159–176. Springer, 2007.

[LSG+18] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg. Meltdown: Reading kernel memory from user space. In William Enck and Adrienne Porter Felt, editors, *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*, pages 973–990. USENIX Association, 2018.

[LSL+19] Shun Li, Siwei Sun, Chaoyun Li, Zihao Wei, and Lei Hu. Constructing low-latency involutory MDS matrices with lightweight circuits. *IACR Trans. Symmetric Cryptol.*, 2019(1):84–117, 2019.

[Moo65] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), April 1965.

[Moo20] Thorben Moos. Unrolled cryptography on silicon A physical security analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):416–442, 2020.

[MS16] Amir Moradi and Tobias Schneider. Side-channel analysis protection and low-latency in action - - case study of PRINCE and midori -. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 517–547, 2016.

[oST79] National Institute of Standards and Technology. Fips-46: Data encryption standard (des). `http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf`, 1979. Accessed: 2021-07-02.

[oST01] National Institute of Standards and Technology. Fips-197: Advanced encryption standard (aes). `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf`, 2001. Accessed: 2021-07-02.

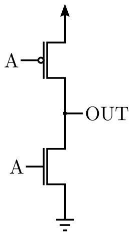
[Qur18] Moinuddin K. Qureshi. CEASER: mitigating conflict-based cache attacks via encrypted-address and remapping. In *51st Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 2018, Fukuoka, Japan, October 20-24, 2018*, pages 775–787. IEEE Computer Society, 2018.

[RCN04] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic. *Digital integrated circuits- A design perspective.* Prentice Hall, 2ed edition, 2004.

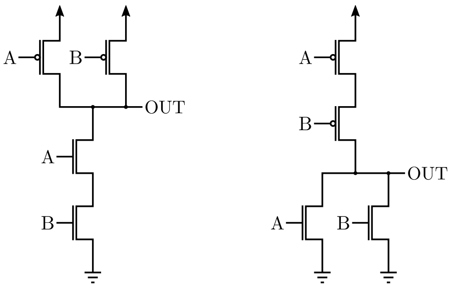
[SBHM20] Pascal Sasdrich, Begül Bilgin, Michael Hutter, and Mark E. Marson. Low-latency hardware masking with application to AES. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(2):300–326, 2020.

[WUG$^+$19] Mario Werner, Thomas Unterluggauer, Lukas Giner, Michael Schwarz, Daniel Gruss, and Stefan Mangard. Scattercache: Thwarting cache attacks via cache set randomization. In Nadia Heninger and Patrick Traynor, editors, *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, pages 675–692. USENIX Association, 2019.

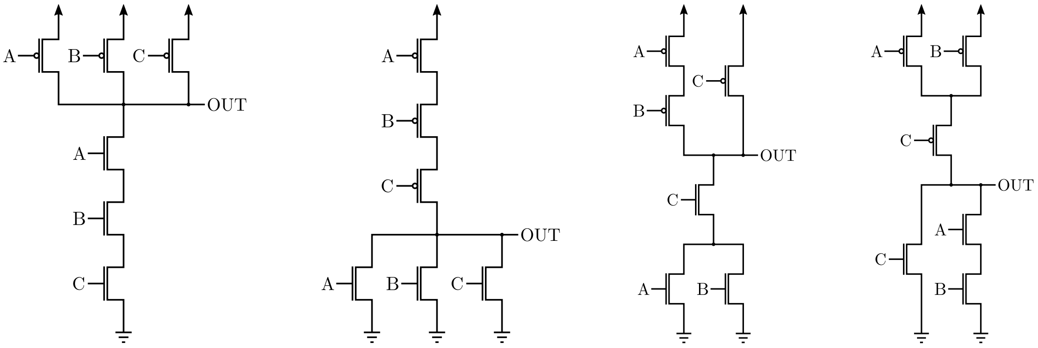
A NCGs



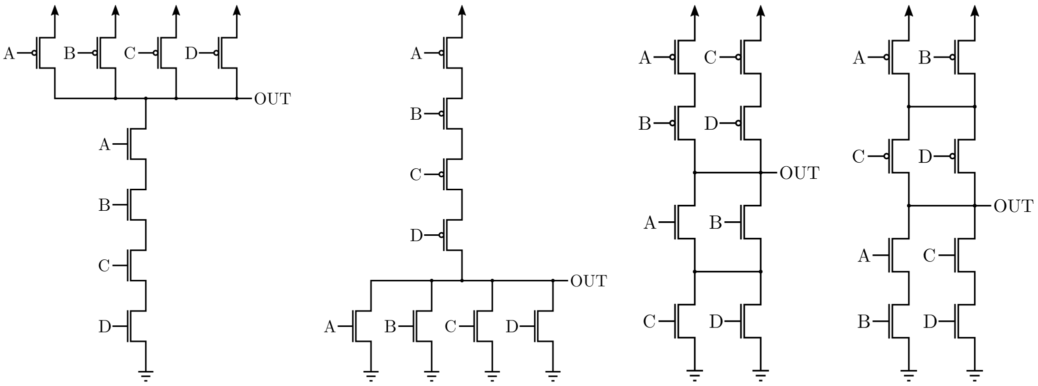
(a) INV



(b) NAND2 (c) NOR2



(d) NAND3 (e) NOR3 (f) OAI21 (g) AOI21



(h) NAND4 (i) NOR4 (j) OAI22 (k) AOI22

Figure 4: *Natural CMOS Gates (NCGs)*: Inverting logic cells realizable in only one stage of $2n$ MOSFETs as static CMOS gates, where n is the number of inputs.

B 1-bit to 1-bit Differential Probabilities

Table 10: 1-bit to 1-bit differential probabilities and linear correlations of the `SPEEDY` S-box.

differential ($\times 2^{-5}$)

$i\backslash j$	0	1	2	3	4	5
0	-	1	3	2	1	1
1	4	3	4	4	-	-
2	1	1	3	3	1	1
3	1	3	-	2	3	-
4	2	2	4	4	2	1
5	2	4	2	4	-	2

linear ($\times 2^{-4}$)

$i\backslash j$	0	1	2	3	4	5
0	3	-	4	-	4	4
1	6	4	4	4	2	4
2	1	-	-	4	4	6
3	6	4	4	-	6	2
4	4	4	-	4	-	3
5	4	4	4	4	4	5

Table 11: 1-bit to 1-bit differential probabilities and linear correlations over $\texttt{SB} \circ \texttt{SC} \circ \texttt{SB}$.

differential ($\times 2^{-10}$)

$i\backslash j$	0	1	2	3	4	5
0	4	6	9	9	6	3
1	12	12	12	12	12	4
2	4	9	9	9	9	3
3	12	9	12	12	6	3
4	8	12	12	12	12	4
5	16	12	16	16	12	4

linear ($\times 2^{-8}$)

$i\backslash j$	0	1	2	3	4	5
0	16	16	16	16	16	24
1	24	16	24	16	24	24
2	24	24	24	24	24	30
3	24	24	24	24	24	24
4	24	16	16	16	24	16
5	24	20	20	20	24	25

C ANF Representation of S and S^{-1}

$$
\begin{aligned}
y_0 =\ & x_3 \oplus x_5x_3 \oplus x_5x_4x_3x_2 \oplus x_5x_4x_1 \oplus x_5x_4x_3x_2x_1 \oplus x_1x_0 \oplus x_5x_4x_1x_0 \oplus x_3x_1x_0 \oplus \\
& x_5x_4x_3x_1x_0 \\
y_1 =\ & x_3 \oplus x_4x_3 \oplus x_5x_4x_3 \oplus x_5x_3x_2 \oplus x_1 \oplus x_3x_1 \oplus x_5x_2x_0 \oplus x_1x_0 \oplus x_3x_1x_0 \\
y_2 =\ & 1 \oplus x_5 \oplus x_5x_2 \oplus x_4x_2 \oplus x_3x_2 \oplus x_4x_3x_2 \oplus x_0 \oplus x_5x_0 \oplus x_4x_0 \oplus x_4x_3x_0 \oplus x_2x_0 \oplus \\
& x_5x_2x_0 \oplus x_3x_1x_0 \,, \\
y_3 =\ & x_2 \oplus x_3x_2 \oplus x_3x_1 \oplus x_5x_0 \oplus x_2x_0 \oplus x_5x_2x_0 \oplus x_4x_2x_0 \oplus x_3x_2x_0 \oplus x_3x_1x_0 \\
y_4 =\ & x_5x_4 \oplus x_1 \oplus x_4x_1 \oplus x_2x_1 \oplus x_4x_2x_1 \oplus x_0 \oplus x_5x_4x_0 \oplus x_4x_3x_0 \oplus x_3x_2x_0 \oplus x_4x_3x_2x_0 \oplus \\
& x_1x_0 \oplus x_4x_1x_0 \oplus x_2x_1x_0 \oplus x_4x_2x_1x_0 \,, \\
y_5 =\ & x_4 \oplus x_5x_2 \oplus x_4x_2 \oplus x_4x_1 \oplus x_4x_2x_1 \oplus x_3x_0 \oplus x_4x_3x_0 \oplus x_5x_3x_2x_0 \oplus x_4x_3x_2x_0 \oplus \\
& x_3x_1x_0 \oplus x_4x_3x_1x_0 \oplus x_2x_1x_0 \oplus x_5x_2x_1x_0 \oplus x_5x_3x_2x_1x_0 \oplus x_4x_3x_2x_1x_0 \,.
\end{aligned}
$$

$$
\begin{aligned}
x_0 =\ & y_4 \oplus y_5y_4 \oplus y_5y_4y_2 \oplus y_5y_1 \oplus y_4y_1 \oplus y_5y_4y_3y_1 \oplus y_5y_3y_2y_1 \oplus y_4y_3y_2y_1 \oplus y_5y_4y_3y_2y_1 \oplus \\
& y_5y_0 \oplus y_5y_4y_0 \oplus y_2y_0 \oplus y_4y_2y_0 \oplus y_3y_2y_0 \oplus y_4y_3y_2y_0 \oplus y_5y_1y_0 \oplus y_2y_1y_0 \,, \\
x_1 =\ & y_5y_3 \oplus y_5y_4y_3 \oplus y_5y_3y_2 \oplus y_5y_4y_3y_2 \oplus y_4y_1 \oplus y_5y_4y_1 \oplus y_3y_1 \oplus y_4y_3y_1 \oplus y_2y_1 \oplus \\
& y_4y_2y_1 \oplus y_3y_2y_1 \oplus y_4y_3y_2y_1 \oplus y_4y_0 \oplus y_5y_4y_0 \oplus y_3y_0 \oplus y_4y_3y_0 \oplus y_5y_4y_3y_0 \oplus y_2y_0 \oplus \\
& y_5y_2y_0 \oplus y_4y_2y_0 \oplus y_3y_2y_0 \oplus y_5y_3y_2y_0 \oplus y_4y_3y_2y_0 \oplus y_4y_1y_0 \oplus y_5y_4y_1y_0 \oplus y_3y_1y_0 \oplus \\
& y_4y_3y_1y_0 \,, \\
x_2 =\ & y_5 \oplus y_5y_4 \oplus y_3 \oplus y_5y_3 \oplus y_4y_3 \oplus y_5y_2 \oplus y_4y_2 \oplus y_5y_3y_2 \oplus y_5y_3y_1 \oplus y_5y_2y_1 \oplus y_4y_2y_1 \oplus \\
& y_5y_4y_2y_1 \oplus y_5y_4y_3y_2y_1 \oplus y_0 \oplus y_4y_0 \oplus y_5y_4y_0 \oplus y_3y_0 \oplus y_4y_3y_0 \oplus y_5y_4y_3y_0 \oplus y_2y_0 \oplus \\
& y_5y_4y_2y_0 \oplus y_5y_3y_2y_0 \oplus y_5y_1y_0 \oplus y_5y_2y_1y_0 \oplus y_4y_2y_1y_0 \,, \\
x_3 =\ & y_5 \oplus y_5y_4 \oplus y_5y_2 \oplus y_5y_4y_2 \oplus y_1 \oplus y_5y_1 \oplus y_4y_1 \oplus y_3y_1 \oplus y_5y_3y_1 \oplus y_2y_1 \oplus y_4y_2y_1 \oplus \\
& y_5y_4y_2y_1 \oplus y_3y_2y_1 \oplus y_4y_3y_2y_1 \oplus y_0 \oplus y_5y_0 \oplus y_4y_0 \oplus y_5y_2y_0 \oplus y_1y_0 \oplus y_5y_1y_0 \oplus \\
& y_4y_1y_0 \oplus y_3y_1y_0 \oplus y_5y_3y_1y_0 \oplus y_4y_3y_1y_0 \oplus y_2y_1y_0 \oplus y_3y_2y_1y_0 \,, \\
x_4 =\ & y_5y_4 \oplus y_3 \oplus y_5y_3 \oplus y_4y_3 \oplus y_5y_4y_3 \oplus y_5y_2 \oplus y_5y_4y_2 \oplus y_3y_2 \oplus y_5y_4y_3y_2 \oplus y_5y_3y_1 \oplus \\
& y_2y_1 \oplus y_4y_2y_1 \oplus y_5y_4y_2y_1 \oplus y_3y_2y_1 \oplus y_5y_3y_2y_1 \oplus y_0 \oplus y_4y_0 \oplus y_3y_0 \oplus y_5y_3y_0 \oplus \\
& y_4y_3y_0 \oplus y_5y_4y_3y_0 \oplus y_5y_2y_0 \oplus y_4y_2y_0 \oplus y_5y_4y_2y_0 \oplus y_3y_2y_0 \oplus y_1y_0 \oplus y_2y_1y_0 \oplus \\
& y_4y_2y_1y_0 \oplus y_4y_3y_2y_1y_0 \,, \\
x_5 =\ & 1 \oplus y_4 \oplus y_5y_4 \oplus y_3 \oplus y_5y_3 \oplus y_2 \oplus y_4y_2 \oplus y_5y_4y_2 \oplus y_3y_2 \oplus y_4y_1 \oplus y_5y_4y_1 \oplus y_4y_3y_1 \oplus \\
& y_5y_4y_3y_1 \oplus y_5y_2y_1 \oplus y_4y_2y_1 \oplus y_5y_4y_2y_1 \oplus y_5y_3y_2y_1 \oplus y_0 \oplus y_4y_0 \oplus y_3y_0 \oplus y_5y_3y_0 \oplus \\
& y_4y_3y_0 \oplus y_5y_4y_3y_0 \oplus y_2y_0 \oplus y_4y_2y_0 \oplus y_3y_2y_0 \oplus y_5y_4y_1y_0 \oplus y_5y_3y_1y_0 \oplus y_5y_2y_1y_0 \oplus \\
& y_3y_2y_1y_0 \oplus y_4y_3y_2y_1y_0 \,.
\end{aligned}
$$

D Additional Security Analysis

Impossible Differential and Zero-Correlation Linear-Hull Attacks One active bit, with respect to both differentials and linear correlations, and in both forward and backward directions can propagate to all the state bits over one (rotated) key-less `SPEEDY` round function and more importantly, none of this activeness is deterministic. But, it should be noted that the activeness of these bits can be related to each other if the last operation is `MC`. Therefore, by combining one round propagation in the forward direction and one round propagation in the backward direction, it might be possible to find impossible differentials or zero-correlation linear-hulls over two (rotated) key-less round functions. But, if we add one `SB` operation in the middle, we ensure that there are no such distinguishers; in other words, there are no impossible differentials or zero-correlation linear-hulls over

$$(\mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB} \circ \mathtt{SC} \circ \mathtt{MC}) \circ \mathtt{SB} \circ (\mathtt{SC} \circ \mathtt{SB} \circ \mathtt{SC} \circ \mathtt{MC} \circ \mathtt{SB})$$

or

$$(\mathtt{SB} \circ \mathtt{SC} \circ \mathtt{MC} \circ \mathtt{SB} \circ \mathtt{SC}) \circ \mathtt{SB} \circ (\mathtt{SC} \circ \mathtt{MC} \circ \mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB}).$$

Therefore, by applying the 2-round distinguisher and extending by one round for key recovery, it might be possible to have a successful attack on 3-round `SPEEDY`, but we expect that more than 3 rounds are secure against those attacks.

Meet-in-the-Middle Attack The maximum number of attacked rounds using meet-in-the-middle technique can be evaluated considering the maximum length of three features: partial-matching, initial structure and splice-and-cut. For partial-matching, the number of rounds in both forward and backward directions cannot reach the full diffusion rounds which for `SPEEDY` in both directions is smaller than one round. The condition for the initial structure is that the key differential trails in both forward and backward directions do not share active non-linear components. As any key differential in `SPEEDY` affects the whole state after one complete round in both directions, there is no such differential which shares active S-box(es) in more than one round. Therefore, it only works up to one round. Splice-and-cut may extend the number of attacked rounds up to the number of full diffusion rounds, i.e., again one round. Thus, it is not possible for the attacker to mount a successful meet-in-the-middle attack on a (2+1+1) = 4-round `SPEEDY`.

Implementation Attacks The protection of `SPEEDY` against implementation attacks like timing, power analysis or fault injection attacks is not a focus of this work. Clearly, a straightforward and unprotected implementation of `SPEEDY` will be susceptible to adversaries who are capable of observing the characteristics of the implementation during its execution. Although this attacker model traditionally requires physical access to the executing device and therefore is typically considered to be less of a concern for desktop and server CPUs (the targeted application area for `SPEEDY`) there have been more and more successful remote power analysis attacks on such devices recently, most notably the PLATYPUS attack [LKO+21]. Thus, even in such contexts, physical adversaries can no longer be ignored and protecting `SPEEDY` against said attacks is a great direction for future research.

In that regard, a recent work has pointed out that, although it is hardly feasible to apply hardware masking to unrolled low-latency cryptography without sacrificing a large portion of its performance due to the necessary inclusion of register stages, simple reset methods (i.e., randomly pre-charging the combinatorial circuit) deliver very promising results against passive side-channel attacks if applied properly [Moo20]. The parallelism, speed and asynchronicity of `SPEEDY` are assumed to be even higher than for the investigated `PRINCE` instance. Thus, we believe that this kind of protection mechanism can most reasonably be applied to unrolled `SPEEDY` in hardware without causing a large performance penalty. According to [Moo20], the cost of this countermeasure is either that the throughput is halved, or that the area is doubled when instantiating the unrolled cipher twice and alternating between pre-charging or encrypting with each circuit. Additionally, the cost for the Random Number Generator (RNG) has to be considered.

E Power Consumption

Table 12: Estimated power consumption of fully-unrolled encryption-only circuits of different cryptographic primitives when synthesized for minimum latency. Estimated for 100 MHz operation.

	Power [mW]					
	Commercial Foundry				NanGate OCL	
Cipher	90 nm LP	65 nm LP	40 nm LP	28 nm HPC	45 nm	15 nm
`Gimli E-M`	16.3489	12.4244	4.1035	8.5614	9.4797	2.7762
MANTIS$_6$	0.2848	0.2108	0.0889	0.3755	0.3680	0.2101
MANTIS$_7$	0.3140	0.2409	0.0986	0.4509	0.4107	0.2318
MANTIS$_8$	0.3503	0.2806	0.1072	0.5269	0.4479	0.2605
`Midori`	0.2652	0.2104	0.0798	0.4512	0.3131	0.1848
`Orthros`	0.6626	0.5814	0.1935	0.7978	0.8711	0.4959
`PRINCE`	0.2162	0.1856	0.0756	0.4079	0.2930	0.1759
`PRINCEv2`	0.2390	0.1827	0.0721	0.3629	0.3041	0.1708
QARMA$_5$-64-σ_0	0.2652	0.2044	0.0867	0.3285	0.3448	0.1997
QARMA$_6$-64-σ_0	0.2993	0.2364	0.0973	0.3973	0.4099	0.2332
QARMA$_7$-64-σ_0	0.3367	0.2640	0.1054	0.4087	0.4529	0.2614
QARMA$_8$-64-σ_0	0.3846	0.2964	0.1205	0.4935	0.5121	0.2896
QARMA$_5$-64-σ_1	0.2669	0.2187	0.0872	0.3672	0.3607	0.2059
QARMA$_6$-64-σ_1	0.3052	0.2443	0.1004	0.4879	0.4350	0.2385
QARMA$_7$-64-σ_1	0.3544	0.2795	0.1161	0.5599	0.4769	0.2700
QARMA$_8$-64-σ_1	0.3903	0.3246	0.1263	0.5906	0.5418	0.2946
`SPEEDY-5-192`	11.6227	7.9766	3.0922	3.9246	4.9508	1.7998
`SPEEDY-6-192`	14.2678	9.7228	3.7569	4.7595	6.1494	2.1764
`SPEEDY-7-192`	17.2552	11.5149	4.4061	5.1270	7.2578	2.5978
`SPEEDY-5-192` *	11.7005	8.6807	3.6014	5.8412	5.3485	2.0160
`SPEEDY-6-192` *	14.2010	10.6287	4.3671	5.1269	6.6413	2.4959
`SPEEDY-7-192` *	17.8889	12.9823	5.1331	5.8412	7.8866	2.9508

* = Optimized HDL code with direct instantiation of library cells based on Figures 2 and 3.

F `SPEEDY` Decryption

For the most part of this work we have ignored the `SPEEDY` decryption. `SPEEDY` is primarily designed to offer ultra fast encryption of data with a high level of security. As discussed by the authors of the `Orthros` low-latency PRF, it is sufficient for many use cases to have a one directional primitive [BIL+21]. Among these use cases are several popular block cipher modes of operation, such as CTR, CMAC and GCM, which all require no decryption routine, as well as applications such as pointer authentication and memory encryption schemes based on Merkle trees [BIL+21]. According to [BIL+21] even a memory encryption scheme applied inside Intel's Software Guard Extensions (SGX) uses adapted variants of GMAC and GCM without requiring the underlying primitive to be invertible. However, since `SPEEDY` does not lack invertibility like `Orthros` does, it can also be used in application scenarios where invertibility and decryption are indeed required, but where it is acceptable that only one direction is extremely efficient. In Table 13 implementation results (latency, area, power) are presented for the `SPEEDY` decryption. Although it is not nearly as efficient as the encryption, the `SPEEDY-5-192` decryption is faster than the `Midori` encryption and many others (cf. Table 7) and the `SPEEDY-6-192` decryption is still faster than the $\texttt{QARMA}_7$-64-σ_1 encryption and a few more (cf. Table 7).

Table 13: Estimated latency, area, and power consumption of the `SPEEDY` decryption routine.

	Minimum Latency [ns]					
	Commercial Foundry				NanGate OCL	
Cipher	90 nm LP	65 nm LP	40 nm LP	28 nm HPC	45 nm	15 nm
`SPEEDY-5-192`	4.827471	3.469787	2.953934	1.387975	5.088359	0.471568
`SPEEDY-6-192`	5.845453	4.197634	3.586378	1.680402	6.174353	0.572912
`SPEEDY-7-192`	6.887968	4.937893	4.240692	1.987920	7.259925	0.672681
	Area [GE]					
`SPEEDY-5-192`	101401.50	118295.50	107298.50	123458.67	70771.33	86302.50
`SPEEDY-6-192`	120336.75	138823.50	127010.00	146688.00	83632.67	102160.50
`SPEEDY-7-192`	138292.50	161802.50	142642.25	163059.67	97923.33	117827.25
	Power [mW]					
`SPEEDY-5-192`	21.6051	15.7708	6.4204	5.7405	11.6600	4.1493
`SPEEDY-6-192`	26.2426	18.7986	7.7360	6.9424	14.0370	4.9956
`SPEEDY-7-192`	30.3541	22.0906	8.6553	7.7193	16.5390	5.8020

G Test Vectors for `SPEEDY-r-192`

`SPEEDY-5-192`

K	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
P	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	E0	D5	6F	BD	95	56	A8	71	CA	49	35	7A	82	2D	04	81	A8	50	2D	DD	16	FE	CE	0F
K	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
P	01	23	45	67	89	AB	CD	EF	01	23	45	67	89	AB	CD	EF	01	23	45	67	89	AB	CD	EF
C	12	3A	5D	7A	D4	5D	E4	4A	27	64	0B	EF	01	F4	8D	42	01	7C	FA	D0	F2	22	3C	3C
K	01	23	45	67	89	AB	CD	EF	01	23	45	67	89	AB	CD	EF	01	23	45	67	89	AB	CD	EF
P	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	FC	FB	8E	9C	23	0A	07	81	B0	63	30	76	FD	62	BF	7D	CE	F4	98	BA	2C	2B	29	6C
K	76	4C	4F	62	54	E1	BF	F2	08	E9	58	62	42	8F	AE	D0	15	84	F4	20	7A	7E	84	77
P	A1	3A	63	24	51	07	0E	43	82	A2	7F	26	A4	06	82	F3	FE	9F	F6	80	28	D2	4F	DB
C	01	DA	25	A9	3D	1C	FC	5E	4C	0B	74	F6	77	EB	74	6C	28	1A	26	01	93	B7	75	5A

`SPEEDY-6-192`

K	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
P	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	A6	D5	18	A2	E5	73	75	15	15	93	11	0A	16	1E	D7	C6	27	8A	BC	D0	31	CB	E8	6C
K	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
P	01	23	45	67	89	AB	CD	EF	01	23	45	67	89	AB	CD	EF	01	23	45	67	89	AB	CD	EF
C	CB	44	11	34	1F	FF	B3	00	03	00	1A	8C	1F	06	FE	D8	7F	F6	89	C5	2D	1E	AB	65
K	01	23	45	67	89	AB	CD	EF	01	23	45	67	89	AB	CD	EF	01	23	45	67	89	AB	CD	EF
P	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	4B	F4	3B	6A	64	8E	81	6A	EF	4F	C9	88	A9	4C	76	7F	A8	36	BA	25	A8	D2	A3	EF
K	76	4C	4F	62	54	E1	BF	F2	08	E9	58	62	42	8F	AE	D0	15	84	F4	20	7A	7E	84	77
P	A1	3A	63	24	51	07	0E	43	82	A2	7F	26	A4	06	82	F3	FE	9F	F6	80	28	D2	4F	DB
C	88	BF	D3	DC	14	0F	38	BC	53	A6	66	87	F5	30	78	60	56	0E	BE	C4	11	00	66	2D

`SPEEDY-7-192`

K	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
P	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	24	7D	30	80	D2	63	F7	4C	B0	3D	DE	6E	57	5C	68	EE	68	EE	E9	57	E1	C2	9C	50
K	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
P	01	23	45	67	89	AB	CD	EF	01	23	45	67	89	AB	CD	EF	01	23	45	67	89	AB	CD	EF
C	B4	8F	32	16	AB	33	AE	01	99	14	2F	6A	07	43	E8	48	1B	FC	37	62	5C	BB	DC	4F
K	01	23	45	67	89	AB	CD	EF	01	23	45	67	89	AB	CD	EF	01	23	45	67	89	AB	CD	EF
P	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	55	65	08	92	98	E4	C1	34	CE	03	12	B2	7E	75	BA	21	A6	8C	0B	4F	46	33	7F	2D
K	76	4C	4F	62	54	E1	BF	F2	08	E9	58	62	42	8F	AE	D0	15	84	F4	20	7A	7E	84	77
P	A1	3A	63	24	51	07	0E	43	82	A2	7F	26	A4	06	82	F3	FE	9F	F6	80	28	D2	4F	DB
C	ED	3D	0E	A1	1C	42	7B	D3	25	70	DF	41	C6	FD	66	EB	BF	49	16	E7	60	ED	09	43

Part III

Conclusion

Chapter 6

Conclusion and Open Problems

The main goal of this thesis was to provide solutions that sustain and advance the physical security of cryptographic hardware implementations in current and future device technologies. This objective is not trivial to achieve as the technological progress in the manufacturing process of integrated circuits significantly affects the security and performance of future device generations. Thus, it is crucial to analyze in detail which challenges, but also opportunities, the continuous evolution of semiconductor technology presents for modern hardware security applications.

The first part of this thesis focuses on one of the most prominent changes in device behavior caused by the continuous shrinking of device geometries, namely the static power consumption of CMOS-based hardware. Researchers had already speculated in the past that this relatively new source of energy dissipation might lead to the emergence of a new side channel which could endanger the security of cryptographic devices in nanometer-scaled technology generations. In order to analyze the evolution of this security threat, multiple custom IC prototypes in continuously smaller feature sizes have been developed, manufactured and analyzed in this thesis. In the course of our experimental analyses on this subject it was discovered that several characteristics of the static power consumption make it particularly dangerous as a source of information leakage. For instance, adversaries can significantly increase the exploitable leakage through this side channel by increasing the supply voltage or the temperature of the device under test to figuratively squeeze the secrets out of the target. Nothing comparable is possible with respect to dynamic power side-channel analysis. Furthermore, while traditional dynamic power analysis attacks can only learn information about secrets while they are actively processed by the hardware, this new form of adversary can often extract information as long as it is present or saved anywhere in the circuit. Finally, it has been discovered that the inherent noise reduction of static power side-channel attacks allows adversaries to circumvent countermeasures that rely on significant noise levels to be effective. Considering all these discoveries it is clearly necessary to develop dedicated protection mechanisms against this threat. In this thesis multiple combined masking and hiding countermeasures have been evaluated for their ability to prevent the extraction of secret information through the static power consumption from devices manufactured in advanced nanometer technologies. Those results will be helpful for the design of high-security cryptographic hardware in nanometer semiconductor technologies in the future. However, it has also been shown that the ability of static power adversaries to obtain low-noise measurements allows to perform attacks with relatively low data complexities even when targeting cryptographic implementations that are protected by masking and hiding combined. Thus, the quest for better solutions has to continue. Developing masking schemes that remain resistant at low noise levels or can be computed without exhibiting univariate static power side-channel leakage are one direction for future research that might be worthwhile.

While common first-order hardware-based masking schemes were shown to be of limited effectiveness against static power side-channel adversaries with clock control, due to the low noise influence in measurements, they are still one of the most important building blocks for the development of secure combined countermeasures in the future. When the noise level is limited (yet sufficient for masking to be effective) it can be of special interest to employ higher-order masking in order to improve the security level of the resulting circuit. Masked gadgets which are probing secure and composable in the presence of glitches and may be scaled to arbitrary orders are essential for the secure assembly of complete higher-order masked implementations of cryptographic primitives. The development and formal analysis of such elements is as important as the practical verification of the desired security guarantees in the resulting hardware. Hence, in this thesis multiple contributions with respect to the formal analysis of masked gadgets and the practical evaluation of the side-channel leakage exhibited by masked cryptographic implementations are made. The methods we have developed for the latter purpose are able to cover multiple statistical moments at a time and possess even the natural ability to find multivariate leakages in measurements using deep learning without any manual effort.

Finally, the continuous progress in semiconductor fabrication opens the door to new applications. In fact, the decreasing cost per logic element manufactured on an integrated circuit allows to shift the focus from area-driven approaches to performance-driven design of cryptographic primitives. Together with the decreasing propagation delays of logic gates in nanometer technologies, this development enables the realization of high-performance cryptography, achieving execution times that were unattainable before. The design of such high-performance primitives clearly benefits the current trend of introducing more encrypted communication to the internal architectures of modern high-end processors in order to prevent the exploitation of microarchitectural defaults for software-based side-channel attacks in future generations of secure computing devices. While the two block ciphers proposed in this thesis, `PRINCEv2` and `SPEEDY`, fulfill the demand for high-performance cryptography to some extent, it is clear that more cryptographic building blocks dedicated to maximum execution speed need to be developed, such as permutations, hash functions and tweakable block ciphers.

Part IV

Appendix

Bibliography

[ABD^{+}14] Massimo Alioto, Simone Bongiovanni, Milena Djukanovic, Giuseppe Scotti, and Alessandro Trifiletti. Effectiveness of Leakage Power Analysis Attacks on DPA-Resistant Logic Styles Under Process Variations. *Transactions on Circuits and Systems I: Regular Papers*, 61(2):429–442, February 2014.

[ABST14] Massimo Alioto, Simone Bongiovanni, Giuseppe Scotti, and Alessandro Trifiletti. Leakage Power Analysis Attacks Against a Bit Slice Implementation of the Serpent Block Cipher. In *MIXDES 2014*, pages 241–246. IEEE, June 2014.

[Acca] Acceptance Rates in IACR Conferences. `https://www.iacr.org/cryptodb/data/acceptance.php`. accessed October 25th, 2021.

[Accb] Selected Areas in Cryptography. `https://link.springer.com/book/10.1007/978-3-030-81652-0`. accessed October 25th, 2021.

[Accc] Welcome to COSADE 2017. `https://cosade.telecom-paristech.fr/presentations/intro.pdf`. accessed October 25th, 2021.

[AGST09] M. Alioto, L. Giancane, G. Scotti, and A. Trifiletti. Leakage power analysis attacks: Well-defined procedure and first experimental results. In *2009 International Conference on Microelectronics - ICM*, pages 46–49, Dec 2009.

[Ava17] Roberto Avanzi. The QARMA block cipher family. almost MDS matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency s-boxes. *IACR Trans. Symmetric Cryptol.*, 2017(1):4–44, 2017.

[BBD^{+}16] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. Strong non-interference and type-directed higher-order masking. In *CCS 2016*, pages 116–129. ACM, 2016.

[BBI^{+}15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 411–436. Springer, 2015.

[BBM^{+}16] Davide Bellizia, Simone Bongiovanni, Pietro MonsurrĂ^{2}, Giuseppe Scotti, and Alessandro Trifiletti. Univariate Power Analysis Attacks Exploiting Static Dissipation of Nanometer CMOS VLSI Circuits for Cryptographic Applications. *Transactions on Emerging Topics in Computing*, 5(3):329–339, May 2016.

[BCG+12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2012.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

[BEK+20] Dusan Bozilov, Maria Eichlseder, Miroslav Knezevic, Baptiste Lambin, Gregor Leander, Thorben Moos, Ventzislav Nikov, Shahram Rasoolzadeh, Yosuke Todo, and Friedrich Wiemer. Princev2 - more security for (almost) no overhead. In Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn, editors, *Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers*, volume 12804 of *Lecture Notes in Computer Science*, pages 483–511. Springer, 2020.

[Ben03] Charles H. Bennett. Notes on landauer's principle, reversible computation, and maxwell's demon. *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, 34(3):501–510, 2003.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

[BL85] Charles H. Bennett and Rolf Landauer. The fundamental physical limits of computation. 253(1):48–56, July 1985.

[BS21] Olivier Bronchain and François-Xavier Standaert. Breaking masked implementations with many shares on 32-bit software platforms or when the security order does not matter. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):202–234, 2021.

[BST16] Davide Bellizia, Giuseppe Scotti, and Alessandro Trifiletti. Implementation of the PRESENT-80 Block Cipher and Analysis of its Vulnerability to Side Channel Attacks Exploiting Static Power. In *MIXDES 2016*, pages 211–216. IEEE, June 2016.

[CDP17] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Convolutional neural networks with data augmentation against jitter-based countermeasures - profiling attacks without pre-processing. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 45–68. Springer, 2017.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.

[Cli] CTS-Climatic Test Chambers. `https://www.cts-umweltsimulation.de/images/produkte/ps-baureihe-c/download/CTS_Climatic_Test_Chambers_eng.pdf`. accessed October 25th, 2021.

[CRB$^+$16] Thomas De Cnudde, Oscar Reparaz, Begül Bilgin, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Masking AES with d+1 shares in hardware. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 194–212. Springer, 2016.

[DS16] François Durvaux and François-Xavier Standaert. From improved leakage detection to the detection of points of interests in leakage traces. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 240–262. Springer, 2016.

[Dun61] Olive Jean Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64, 1961.

[FGP$^+$18] Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and François-Xavier Standaert. Composable masking schemes in the presence of physical defaults & the robust probing model. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):89–120, 2018.

[FMM20] Bijan Fadaeinia, Thorben Moos, and Amir Moradi. BSPL: balanced static power logic. *IACR Cryptol. ePrint Arch.*, page 558, 2020.

[FMM21] Bijan Fadaeinia, Thorben Moos, and Amir Moradi. Balancing the leakage currents in nanometer CMOS logic - a challenging goal. *Applied Sciences*, 11(15), 2021.

[GIB18] Hannes Groß, Rinat Iusupov, and Roderick Bloem. Generic low-latency masking in hardware. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):1–21, 2018.

[GJJR11] G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi. A testing methodology for side channel resistance validation. In *NIST non-invasive attack testing workshop*, 2011.

[GM17] Hannes Groß and Stefan Mangard. Reconciling d+1 masking in hardware and software. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 115–136. Springer, 2017.

[GM18] Hannes Groß and Stefan Mangard. A unified masking approach. *J. Cryptographic Engineering*, 8(2):109–124, 2018.

[GMK16] Hannes Groß, Stefan Mangard, and Thomas Korak. Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. In Begül Bilgin, Svetla Nikova, and Vincent Rijmen, editors, *Proceedings of the ACM Workshop on Theory of Implementation Security, TISCCS 2016 Vienna, Austria, October, 2016*, page 3. ACM, 2016.

[GMK17] Hannes Groß, Stefan Mangard, and Thomas Korak. An efficient side-channel protected AES implementation with arbitrary protection order. In Helena Handschuh, editor, *Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings*, volume 10159 of *Lecture Notes in Computer Science*, pages 95–112. Springer, 2017.

[GMMP20] Samaneh Ghandali, Thorben Moos, Amir Moradi, and Christof Paar. Side-channel hardware trojan for provably-secure sca-protected implementations. *IEEE Trans. Very Large Scale Integr. Syst.*, 28(6):1435–1448, 2020.

[GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.

[GST14] Daniel Genkin, Adi Shamir, and Eran Tromer. RSA key extraction via low-bandwidth acoustic cryptanalysis. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 444–461. Springer, 2014.

[Hel09] Domenik Helms. *Leakage Models for High Level Power Estimation.* PhD thesis, Carl von Ossietzky Universität Oldenburg, 2009.

[HGM$^+$11] Gabriel Hospodar, Benedikt Gierlichs, Elke De Mulder, Ingrid Verbauwhede, and Joos Vandewalle. Machine learning in side-channel analysis: a first study. *J. Cryptographic Engineering*, 1(4):293–302, 2011.

[Hol79] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[HRO] Teledyne LeCroy HRO Series Data Sheet. `http://cdn.teledynelecroy.com/files/pdf/hro-12bit_datasheet.pdf`. accessed October 25th, 2021.

[HS14] Michael Hutter and Jörn-Marc Schmidt. The temperature side channel and heating fault attacks. *IACR Cryptology ePrint Archive*, 2014:190, 2014.

[IM14] S. Shiney Immaculate and K. Manoharan. Analysis of Leakage Power Attacks on DPA Resistant Logic Styles: A Survey. *International Journal of Computer Science Trends and Technology*, 2(5):136–141, September 2014.

[Ins] Analog Devices AD8421 Data Sheet Rev. 0. `http://www.analog.com/media/en/technical-documentation/data-sheets/AD8421.pdf`. accessed October 25th, 2021.

[ISW03] Yuval Ishai, Amit Sahai, and David A. Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.

[JS17] Anthony Journault and François-Xavier Standaert. Very high order masking: Efficient implementation and security evaluation. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 623–643. Springer, 2017.

[Kei] Tektronix keithley 2450 sourcemeter smu instrument datasheet. `https://de.tek.com/datasheet/smu-2400-graphical-sourcemeter/model-2450-touchscreen-source-measure-unit-smu-instrument-`. accessed October 25th, 2021.

[KHF$^+$19] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. Spectre attacks: Exploiting speculative execution. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 1–19. IEEE, 2019.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO ’99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[KMM19] Naghmeh Karimi, Thorben Moos, and Amir Moradi. Exploring the effect of device aging on static power analysis attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):233–256, 2019.

[KMM20] David Knichel, Thorben Moos, and Amir Moradi. The risk of outsourcing: Hidden SCA trojans in third-party ip-cores threaten cryptographic ics. In *IEEE European Test Symposium, ETS 2020, Tallinn, Estonia, May 25-29, 2020*, pages 1–6. IEEE, 2020.

[Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[KR07] Ian Kuon and Jonathan Rose. Measuring the gap between fpgas and asics. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 26(2):203–215, 2007.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[Lan61] Rolf Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191, 1961.

[LB08] Lang Lin and Wayne Burleson. Leakage-Based Differential Power Analysis (LDPA) on Sub-90nm CMOS Cryptosystems. In *ISCAS 2008*, pages 252–255. IEEE, May 2008.

[LKMM21] Oleksiy Lisovets, David Knichel, Thorben Moos, and Amir Moradi. Let's take it offline: Boosting brute-force attacks on iphone's user authentication through SCA. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):496–519, 2021.

[Llo00] Seth Lloyd. Ultimate physical limits to computation. *Nature*, 406:1047–1054, 2000.

[LMMR21] Gregor Leander, Thorben Moos, Amir Moradi, and Shahram Rasoolzadeh. The SPEEDY family of block ciphers engineering an ultra low-latency cipher from gate level for secure processor architectures. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(4):510–545, 2021.

[LSG$^+$18] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg. Meltdown: Reading kernel memory from user space. In William Enck and Adrienne Porter Felt, editors, *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*, pages 973–990. USENIX Association, 2018.

[MDP20] Loïc Masure, Cécile Dumas, and Emmanuel Prouff. A comprehensive study of deep learning for side-channel analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(1):348–375, 2020.

[MM17] Thorben Moos and Amir Moradi. On the easiness of turning higher-order leakages into first-order. In Sylvain Guilley, editor, *Constructive Side-Channel Analysis and*

Secure Design - 8th International Workshop, COSADE 2017, Paris, France, April 13-14, 2017, Revised Selected Papers, volume 10348 of *Lecture Notes in Computer Science*, pages 153–170. Springer, 2017.

[MM21] Thorben Moos and Amir Moradi. Countermeasures against static power attacks - comparing exhaustive logic balancing and other protection schemes in 28 nm CMOS -. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):780–805, 2021.

[MME10] Amir Moradi, Oliver Mischke, and Thomas Eisenbarth. Correlation-enhanced power analysis collision attack. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 125–139. Springer, 2010.

[MMM21] Nicolai Müller, Thorben Moos, and Amir Moradi. Low-latency hardware masking of PRINCE. In *Constructive Side-Channel Analysis and Secure Design - 12th International Workshop, COSADE 2021, Lugano, Switzerland, October 25-27, 2021*, Lecture Notes in Computer Science. Springer, 2021.

[MMR17] Thorben Moos, Amir Moradi, and Bastian Richter. Static power side-channel analysis of a threshold implementation prototype chip. In David Atienza and Giorgio Di Natale, editors, *Design, Automation & Test in Europe Conference & Exhibition, DATE 2017, Lausanne, Switzerland, March 27-31, 2017*, pages 1324–1329. IEEE, 2017.

[MMR20] Thorben Moos, Amir Moradi, and Bastian Richter. Static power side-channel analysis - an investigation of measurement factors. *IEEE Trans. Very Large Scale Integr. Syst.*, 28(2):376–389, 2020.

[MMSS19] Thorben Moos, Amir Moradi, Tobias Schneider, and François-Xavier Standaert. Glitch-resistant masking revisited or why proofs in the robust probing model are needed. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):256–292, 2019.

[Moo65] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), April 1965.

[Moo19] Thorben Moos. Static power SCA of sub-100 nm CMOS asics and the insecurity of masking schemes in low-noise environments. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):202–232, 2019.

[Moo20] Thorben Moos. Unrolled cryptography on silicon A physical security analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):416–442, 2020.

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards.* Springer, 2007.

[Mor14] Amir Moradi. Side-channel leakage through static power - should we care about in practice? In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan,* *South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 562–579. Springer, 2014.

[MPP16] Houssem Maghrebi, Thibault Portigliatti, and Emmanuel Prouff. Breaking cryptographic implementations using deep learning techniques. In Claude Carlet, M. Anwar Hasan, and Vishal Saraswat, editors, *Security, Privacy, and Applied Cryptography Engineering - 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings*, volume 10076 of *Lecture Notes in Computer Science*, pages 3–26. Springer, 2016.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 278–296. Springer, 2004.

[MRSS18] Amir Moradi, Bastian Richter, Tobias Schneider, and François-Xavier Standaert. Leakage detection with the x2-test. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):209–237, 2018.

[MS16] Amir Moradi and François-Xavier Standaert. Moments-correlating DPA. In Begül Bilgin, Svetla Nikova, and Vincent Rijmen, editors, *Proceedings of the ACM Workshop on Theory of Implementation Security, TISCCS 2016 Vienna, Austria, October, 2016*, pages 5–15. ACM, 2016.

[MWM21] Thorben Moos, Felix Wegener, and Amir Moradi. DL-LA: deep learning leakage assessment A modern roadmap for SCA evaluations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):552–598, 2021.

[NIS18] NIST. Submission requirements and evaluation criteria for the lightweight cryptography standardization process. `https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf`, 2018.

[NRR06] Svetla Nikova, Christian Rechberger, and Vincent Rijmen. Threshold implementations against side-channel attacks and glitches. In Peng Ning, Sihan Qing, and Ninghui Li, editors, *Information and Communications Security, 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006, Proceedings*, volume 4307 of *Lecture Notes in Computer Science*, pages 529–545. Springer, 2006.

[Opa] Analog Devices AD8676 Data Sheet Rev. C. `http://www.analog.com/media/en/technical-documentation/data-sheets/AD8676.pdf`. accessed October 25th, 2021.

[oST01] National Institute of Standards and Technology. Fips-197: Advanced encryption standard (aes). `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf`, 2001. Accessed: 2021-10-25.

[PCBP21] Guilherme Perin, Lukasz Chmielewski, Lejla Batina, and Stjepan Picek. Keep it unsupervised: Horizontal attacks meet deep learning. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(1):343–372, 2021.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 142–159. Springer, 2013.

[PSK$^+$18] Stjepan Picek, Ioannis Petros Samiotis, Jaehun Kim, Annelie Heuser, Shivam Bhasin, and Axel Legay. On the performance of convolutional neural networks for side-channel analysis. In Anupam Chattopadhyay, Chester Rebeiro, and Yuval Yarom, editors, *Security, Privacy, and Applied Cryptography Engineering - 8th International Conference, SPACE 2018, Kanpur, India, December 15-19, 2018, Proceedings*, volume 11348 of *Lecture Notes in Computer Science*, pages 157–176. Springer, 2018.

[PSKM15] Santos Merino Del Pozo, François-Xavier Standaert, Dina Kamel, and Amir Moradi. Side-channel attacks from static power: when should we care? In Wolfgang Nebel and David Atienza, editors, *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, DATE 2015, Grenoble, France, March 9-13, 2015*, pages 145–150. ACM, 2015.

[RBN$^+$15] Oscar Reparaz, Begül Bilgin, Svetla Nikova, Benedikt Gierlichs, and Ingrid Verbauwhede. Consolidating masking schemes. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 764–783. Springer, 2015.

[RCN04] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic. *Digital Integrated Circuits- A Design Perspective.* Prentice Hall, 2nd ed edition, 2004.

[Riv90] Ronald L. Rivest. Cryptography. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, pages 717–755. Elsevier and MIT Press, 1990.

[RMM03] Kauschick Roy, Saibal Mukhopadhyay, and Hamid Mahmoodi-Meimand. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits. *Proc. IEEE*, 91(2):305–327, 2003.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[Sak] SAKURA-G. `https://satoh.cs.uec.ac.jp/SAKURA/hardware/SAKURA-G.html`. accessed October 25th, 2021.

[Sas] SASEBO-R. `https://satoh.cs.uec.ac.jp/SASEBO/en/board/sasebo-r.html`. accessed October 25th, 2021.

[SM15] Tobias Schneider and Amir Moradi. Leakage assessment methodology - A clear roadmap for side-channel evaluations. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.

[SM16] Tobias Schneider and Amir Moradi. Leakage assessment methodology - extended version. *J. Cryptogr. Eng.*, 6(2):85–99, 2016.

[SNK$^+$12] Alexander Schlösser, Dmitry Nedospasov, Juliane Krämer, Susanna Orlic, and Jean-Pierre Seifert. Simple photonic emission analysis of AES - photonic side channel analysis for the rest of us. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*, pages 41–57. Springer, 2012.

[Sta18] François-Xavier Standaert. How (not) to use welch's t-test in side-channel security evaluations. In Begül Bilgin and Jean-Bernard Fischer, editors, *Smart Card Research and Advanced Applications, 17th International Conference, CARDIS 2018, Montpellier, France, November 12-14, 2018, Revised Selected Papers*, volume 11389 of *Lecture Notes in Computer Science*, pages 65–79. Springer, 2018.

[SVO$^+$10] François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The world is not enough: Another look on second-order DPA. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.

[Tim19] Benjamin Timon. Non-profiled deep learning-based side-channel attacks with sensitivity analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):107–131, 2019.

[TV04] Kris Tiri and Ingrid Verbauwhede. A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In *2004 Design, Automation and Test in Europe Conference and Exposition (DATE 2004), 16-20 February 2004, Paris, France*, pages 246–251. IEEE Computer Society, 2004.

[VMKS12] Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. Shuffling against side-channel attacks: A comprehensive study with cautionary note. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 740–757. Springer, 2012.

[Wav] Teledyne LeCroy WaveRunner 8000 Series Data Sheet. `http://cdn.teledynelecroy.com/files/pdf/waverunner8000-datasheet.pdf`. accessed October 25th, 2021.

[WO19] Carolyn Whitnall and Elisabeth Oswald. A cautionary note regarding the usage of leakage detection tests in security evaluation. *IACR Cryptol. ePrint Arch.*, page 703, 2019.

[XH17] J. Xu and H. M. Heys. Template attacks based on static power analysis of block ciphers in 45-nm cmos environment. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1256–1259, Aug 2017.

[Ši67] Zbyněk Šidák. Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American Statistical Association*, 62(318):626–633, 1967.

List of Figures

1.1 This figure shows the four different ASIC prototypes which have been developed, manufactured and analyzed specifically for this thesis. 12
1.2 This figure illustrates the difference between (a) moment-based second-order analysis of two distributions and (b) first-order analysis of a chosen slice of distributions. 14
1.3 This bar graph compares the average normalized latency of different cryptographic S-boxes in hardware across 6 CMOS standard cell libraries. 18
1.4 This bar graph compares the average normalized latency of different encryption functions in hardware across 6 CMOS standard cell libraries. 18

2.1 These photos show some of the ASICs we developed and manufactured for this thesis from the naked dies to the packaged and mounted chips. 25
2.2 These photos show the PCBs we have developed as measurement boards for the different ASIC prototypes. 26
2.3 This figure illustrates the charging and discharging currents in a CMOS inverter gate. 29
2.4 This figure illustrates the short-circuit current in a CMOS inverter gate. 29
2.5 This figure illustrates the leakage currents occurring in a CMOS inverter gate for static input signals. 30
2.6 Two-input CMOS NAND gate (left) and formation of inactive transistors across the power supply path for different inputs (right), when replacing conducting transistors with ideal wires. 31
2.7 This figure shows two hypothetical leakage distributions for different values of an intermediate result in an unmasked implementation. 35
2.8 This figure shows two hypothetical leakage distributions for different values of a shared intermediate result in a first-order masked implementation. 36
2.9 This figure shows a sample trace of a serialized PRESENT hardware implementation measured on a SAKURA-G board. 42
2.10 These photographs show a setup to perform static power measurements inside a climate chamber. 43
2.11 This figure shows a custom low-noise DC amplifier for static power measurements. 44
2.12 This figure shows a third-order (Butterworth Pi) LC low pass filter with cutoff-frequency of approximately 100 Hz. 45
2.13 This figure illustrates the complete static power measurement setup based on an oscilloscope. 45
2.14 This figure shows a Keithley 2450 Source Measure Unit (SMU). 46
2.15 This figure illustrates the complete static power measurement setup based on a Source Measure Unit (SMU). 46

2.16 This figure illustrates the post-processing technique for 100 exemplary measurements. 47

List of Tables

1.1 This table contains implementation details about the four ASIC prototypes and references to the corresponding publications. 13

About the Author

Author information as of November 2021.

Personal Data

Name	Thorben Moos
Address	Chair for Security Engineering Universitätsstr. 150, ID 2/621 44801 Bochum, Germany
E-Mail	thorben.moos@rub.de / thorben.moos@uclouvain.be
Date of birth	February 3, 1992
Place of birth	Unna, Germany

Education

Since 10/2016	**PhD-student**, *Ruhr-Universität Bochum*, Electrical and Information Engineering.
10/2014 - 09/2016	**M.Sc.**, *Ruhr-Universität Bochum*, IT Security/Information Engineering. Average Score: Excellent (96%)
10/2011 - 09/2014	**B.Sc.**, *Ruhr-Universität Bochum*, IT Security/Information Engineering.

Professional Experience

Since 10/2016	**Research Assistant**, *Ruhr-Universität Bochum.*
04/2014 - 09/2014	**Intern**, *Konzernforschung Volkswagen AG*, Wolfsburg.
04/2013 - 09/2016	**Student Assistant**, *Ruhr-Universität Bochum.*

Publications and Academic Activities

Peer-Reviewed Journal Papers

- Naghmeh Karimi, Thorben Moos, and Amir Moradi. Exploring the effect of device aging on static power analysis attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):233–256, 2019
- Thorben Moos. Static power SCA of sub-100 nm CMOS asics and the insecurity of masking schemes in low-noise environments. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):202–232, 2019
- Thorben Moos, Amir Moradi, Tobias Schneider, and François-Xavier Standaert. Glitch-resistant masking revisited or why proofs in the robust probing model are needed. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):256–292, 2019
- Samaneh Ghandali, Thorben Moos, Amir Moradi, and Christof Paar. Side-channel hardware trojan for provably-secure sca-protected implementations. *IEEE Trans. Very Large Scale Integr. Syst.*, 28(6):1435–1448, 2020
- Thorben Moos, Amir Moradi, and Bastian Richter. Static power side-channel analysis - an investigation of measurement factors. *IEEE Trans. Very Large Scale Integr. Syst.*, 28(2):376–389, 2020
- Thorben Moos. Unrolled cryptography on silicon A physical security analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):416–442, 2020
- Gregor Leander, Thorben Moos, Amir Moradi, and Shahram Rasoolzadeh. The SPEEDY family of block ciphers engineering an ultra low-latency cipher from gate level for secure processor architectures. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(4):510–545, 2021
- Thorben Moos and Amir Moradi. Countermeasures against static power attacks - comparing exhaustive logic balancing and other protection schemes in 28 nm CMOS -. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):780–805, 2021
- Thorben Moos, Felix Wegener, and Amir Moradi. DL-LA: deep learning leakage assessment A modern roadmap for SCA evaluations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):552–598, 2021
- Oleksiy Lisovets, David Knichel, Thorben Moos, and Amir Moradi. Let's take it offline: Boosting brute-force attacks on iphone's user authentication through SCA. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):496–519, 2021

- Bijan Fadaeinia, Thorben Moos, and Amir Moradi. Balancing the leakage currents in nanometer CMOS logic - a challenging goal. *Applied Sciences*, 11(15), 2021

Peer-Reviewed Conference Proceedings

- Thorben Moos, Amir Moradi, and Bastian Richter. Static power side-channel analysis of a threshold implementation prototype chip. In David Atienza and Giorgio Di Natale, editors, *Design, Automation & Test in Europe Conference & Exhibition, DATE 2017, Lausanne, Switzerland, March 27-31, 2017*, pages 1324–1329. IEEE, 2017

- Thorben Moos and Amir Moradi. On the easiness of turning higher-order leakages into first-order. In Sylvain Guilley, editor, *Constructive Side-Channel Analysis and Secure Design - 8th International Workshop, COSADE 2017, Paris, France, April 13-14, 2017, Revised Selected Papers*, volume 10348 of *Lecture Notes in Computer Science*, pages 153–170. Springer, 2017

- Dusan Bozilov, Maria Eichlseder, Miroslav Knezevic, Baptiste Lambin, Gregor Leander, Thorben Moos, Ventzislav Nikov, Shahram Rasoolzadeh, Yosuke Todo, and Friedrich Wiemer. Princev2 - more security for (almost) no overhead. In Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn, editors, *Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers*, volume 12804 of *Lecture Notes in Computer Science*, pages 483–511. Springer, 2020

- David Knichel, Thorben Moos, and Amir Moradi. The risk of outsourcing: Hidden SCA trojans in third-party ip-cores threaten cryptographic ics. In *IEEE European Test Symposium, ETS 2020, Tallinn, Estonia, May 25-29, 2020*, pages 1–6. IEEE, 2020

- Nicolai Müller, Thorben Moos, and Amir Moradi. Low-latency hardware masking of PRINCE. In *Constructive Side-Channel Analysis and Secure Design - 12th International Workshop, COSADE 2021, Lugano, Switzerland, October 25-27, 2021*, Lecture Notes in Computer Science. Springer, 2021

Technical Reports

- Bijan Fadaeinia, Thorben Moos, and Amir Moradi. BSPL: balanced static power logic. *IACR Cryptol. ePrint Arch.*, page 558, 2020

Awards

03/2017	**DATE 2017 Nominated for Best Paper Award** Thorben Moos, Amir Moradi, Bastian Richter *Static power side-channel analysis of a threshold implementation prototype chip*
08/2019	**CHES 2019 Best Paper Award** Thorben Moos, Amir Moradi, Tobias Schneider, François-Xavier Standaert *Glitch-Resistant Masking Revisited or Why Proofs in the Robust Probing Model are Needed*

Participation in Selected Conferences and Workshops

- CHES 2021, *Virtual Conference*
- CHES 2020, *Virtual Conference*
- CHES & FDTC 2019, *Atlanta, USA*
- Kangacrypt & ASEC 2018, *Adelaide, Australia*
- CHES & FDTC 2018, *Amsterdam, The Netherlands*
- CHES & FDTC & PROOFS 2017, *Taipei, Taiwan*
- Summer school on real-world crypto and privacy 2017, *Šibenik, Croatia*
- COSADE 2017, *Paris, France*
- DATE 2017, *Lausanne, Switzerland*