Low-Cost Side-Channel Analysis and Fault Analysis Setups and Their Application Targeting Secured ASICs

by

Thomas Korak

A PhD Thesis
Presented to the Faculty of Computer Science in Partial Fulfillment of the Requirements for the PhD Degree

Assessors

Prof. Stefan Mangard (TU Graz, Austria)
Prof. Tim Güneysu (University of Bremen, Germany)

October 2015

Institute for Applied Information Processing and Communications (IAIK)
Faculty of Computer Science
Graz University of Technology, Austria

Abstract

Physical attacks aim at revealing secret data that is processed by an electronic device by observing side-channel information such as the power consumption or by intentionally injecting faults during a critical computation. A wide range of electronic devices is vulnerable to this kind of attack. However, literature published in this field often lacks clear statements whether the presented attacks can be performed by applying off-the-shelf, low-cost equipment or if specialized tools are required.

With this work we aim at demonstrating the power of measurement setups enabling physical attacks, built-up with off-the-shelf, low-cost equipment exclusively. Attacks which can be conducted with such setups pose a serious threat as they can be performed by a wide range of attackers. They are therefore more likely to lead to real-world exploits than attacks which require highly specialized equipment that is hard to obtain.

The main contribution consists in verifying the functionality of the low-cost setups by performing exemplary side-channel analysis (SCA) and fault analysis (FA) attacks. The attacked devices include chips providing authentication services for radio-frequency identification (RFID) systems, microcontrollers for sensor nodes, and application-specific integrated circuits (ASICs) implementing cryptographic algorithms. When targeting RFID systems, we show that key-recovery attacks can be mounted at a distance of up to one meter with low-cost equipment, posing a serious threat for real-world systems. In addition, we are the first to present SCA-attack results targeting an SCA-protected, keyed Keccak instance implemented on a taped-out ASIC. The results show that the proposed secret-sharing countermeasure applied on low-resource devices does not lead to a sufficient protection level when keeping in mind the immense overhead in terms of area and runtime. We further study the influence of setup parameters (laser pulse length, laser power, and laser focus) on the success of invasive, optical fault injections. The parameters have to be chosen with great care for a successful, reproducible fault injection. Our non-invasive, low-cost fault injection setup proved to be an effective tool for disturbing the correct operation of a wide range of microcontrollers frequently applied for sensor-node applications.

The results presented in this thesis clearly show that the power of low-cost setups for SCA attacks and FA attacks must not be underestimated. These attacks pose a serious threat for electronic devices executing security-relevant applications and have to be considered already in the early design phase.

Acknowledgements

I would like to thank all people who supported me during the last years. Special thanks to all former and current colleagues at IAIK. It was always interesting to do joint research leading to this thesis and have some fruitful discussions during the coffee breaks. Special thanks to my former colleague Thomas Plos for his great support during my PhD start. I also want to thank my former group leader Jörn-Marc Schmidt and my current group leader Stefan Mangard for their support during the PhD thesis.

I would also like to thank all the coauthors and all the partners who have contributed in some way to my work.

I would also like to thank my assessors, Stefan Mangard and Tim Güneysu for their reviews and detailed comments leading to the final version of this thesis.

Finally and most importantly, I would like to thank my family and my friends for their great support during my studies. Especially my parents endorsed me in every situation, they always found the motivating words.

Thomas Korak
Graz, October 2015

Table of Contents

List of Tables

List of Figures

Acronyms

AE	Authenticated Encryption
AES	Advanced Encryption Standard
AFE	Analog Front-End
AP	Access Point
APDU	Application Protocol Data Unit
API	Application Programming Interface
ASIC	Application-Specific Integrated Circuit
CAESAR	Competition for Authenticated Encryption: Security, Applicability, and Robustness
CED	Concurrent Error Detection
CEMA	Correlation Electromagnetic Analysis
CMOS	Complementary Metal-Oxide Semiconductor
CPA	Correlation Power Analysis
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CRYPTA	Cryptographically Protected Tag
DCM	Digital Clock Manager
DEMA	Differential Electromagnetic Analysis
DES	Data Encryption Standard
DPA	Differential Power Analysis
DST	Digital Signature Transponder
DUT	Device Under Test
ECC	Elliptic Curve Cryptography
EM	Electromagnetic
FA	Fault Analysis
FDT	Frame Delay Time
FET	Field-Effect Transistor
FWT	Frame Waiting Time
FIB	Focused Ion Beam
FPGA	Field-Programmable Gate Array
GE	Gate Equivalent
HDL	Hardware Description Language
HF	High Frequency
IC	Integrated Circuit
IV	Intermediate Value
IoT	Internet of Things

ISO	international Organization for Standardization
LIVA	Light-Induced Voltage Alteration
LSB	Least Significant Bit
LSI	Large Scale Integration
LRPRF	Leakage-Resilient Pseudo-Random Function
MAC	Message Authentication Code
MCU	Microcontroller Unit
MSB	Most Significant Bit
NIST	National Institute of Standards and Technology
NFC	Near-Field Communication
nrdp	Nearest-Rival Distinguishing Power
PCB	Printed Circuit Board
PCD	Proximity Coupling Device
PICC	Proximity Integrated Circuit Card
PTV	Process, Temperature, and Voltage
RAM	Random-Access Memory
RF	Radio Frequency
RFID	Radio-Frequency Identification
RSA	Rivest, Shamir, and Adleman
SCA	Side-Channel Analysis
SHA	Secure Hash Algorithm
SNR	Signal-to-Noise Ratio
SRAM	Static Random-Access Memory
TAMPRES	Tamper Resistant Sensor Nodes
TI	Texas Instruments
UART	Universal Asynchronous Receiver Transmitter
UHF	Ultra-High Frequency
UID	Used Identification
USB	Universal Serial Bus
UV	Ultraviolet
WSN	Wireless Sensor Network
WTX	Waiting Time Extension

1 Introduction

The amount of data shared and transmitted between mobile devices has grown rapidly during the last decade. It is a fact that the majority of digital media consumption now takes place on mobile devices. Since 2014 the number of mobile Internet users exceeds the number of the traditional desktop Internet users and the gap is likely to grow in future [29]. Also other new technologies like Radio Frequency Identification (RFID) systems, contactless smartcards based on Near-Field Communication (NFC), or wireless sensor nodes distributing their measurement data, contribute to the growth of transmitted data. Besides the convenience, the trend to mobile Internet usage and the penetration of new technologies in our everyday life also poses several challenges. One of these challenges is the secure handling of data. The majority of data is sent over insecure channels allowing non-authorized parties to eavesdrop, modify, or redirect a communication of potentially confidential data. In order to prevent this, sender and receiver of confidential data have to put security measures in place. The goal of these measures is to provide confidentiality, integrity and authenticity of the transmitted data. A wide range of cryptosystems are available to realize these properties. Following Kerckhoffs's principle, all information of the cryptosystem is typically public knowledge and only the key is kept secret by the communication partners. So a main asset in cryptosystems is the secrecy of the private key. Hence an attacker should not be able to gain information about the private key used during a communication. To ensure this, different attacks have to be considered during the design and in particular the implementation of a cryptosystem. In fact, attacks often target weaknesses in the implementation of cryptographic algorithms and can be grouped into logical and physical attacks.

Logical attacks assume that an attacker can use the existing communication interface of a device to mount an attack. Software vulnerabilities, security APIs,

or test and debug interfaces can be used for performing logical attacks [25, 88]. A wide range of logical attacks can also be mounted remotely.

If an attacker has physical access to the attacked device for some time, physical attacks become feasible. The goal of a physical attack is to reveal the secret key or other relevant, secret information, by observing and measuring side-channel information during the execution of the cryptographic algorithm. These side-channel information can for example be the power consumption, the electromagnetic emanation, or the timing behavior. Whenever the side-channel information is correlated with some secret values processed inside the device, we speak of exploitable leakage of secret information through the side channel. Examples are key-dependent power consumption or key-dependent runtime of the algorithm. In this case an analysis of the side-channel measurements allows to extract the value of the secret key or narrow down the search space for a brute-force attack. In addition to passively measuring side-channel information, the physical access also allows to actively manipulate the device to force erroneous computations. Analyzing the erroneous output allows to gain additional knowledge about internal data. This knowledge about internal data can assist in revealing the secret key. Active manipulations can be achieved in a non-invasive, semi-invasive, or invasive manner. In non-invasive scenarios, no modification of the attacked device is required, examples are tampering with the ambient temperature or the clock signal. Semi-invasive and invasive scenarios require a modification of the attacked device. This modification often consists of a decapsulation step to access the chip die. Semi-invasive attacks do not contact or modify the inner chip structure (e.g., optical fault injections). Invasive attacks require contact or modification of the inner chip structure what can be done with probing cards or focused ion-beam (FIB) stations. Publications in the past 15 years have shown that the full range of electronic devices can be vulnerable to physical attacks. Examples are notebooks [45], smart phones [133], smart cards [91], and contactless chip cards based on the radio-frequency identification (RFID) and near-field communication (NFC) technology [106].

1.1 Motivation

In general, physical attacks are always successful if an attacker is equipped with sufficient resources (money, time, expertise, ...). In practice, the main goal is to make attacks so hard that the effort exceeds the profit for the attack. As a result, attacks requiring only a limited amount of resources have to be considered. Many attacks published in the past lack the clear statement regarding the applicability in case of a budget-limited attacker (One exception is e.g. [65]). In this work we present a wide range of low-cost setups enabling physical attacks to showcase the power of budget-limited attackers. We target different devices, all exposed to physical attacks due to their fields of application. Among these devices are RFID and NFC tags, microcontrollers for sensor nodes, and application-specific integrated circuits (ASICs) implementing state-of-the-art cryptographic algorithms. The fact if a setup can be categorized as low-cost depends on sev-

eral factors. The first and most important factor is that all parts required for the setup are off-the-shelf parts and can be bought by everybody. For some custom-made equipment no mass market exists, leading to limited availability and high costs, what leads to the second factor. The price of all devices included in the setup represents the second factor. It is influenced by the budget of the attacker. We upper-bound the equipment costs with 40 000 EUR. The value of 40 000 EUR has been chosen because this value reflects the maximum costs for the equipment we have used during our research. The devices with the biggest influence on the costs are a high-end oscilloscope (*'LeCroy WP 725 Zi'*, approx. 20 000 EUR), a fully-automated three axes stepper table (*'Marzhauser SCAN 75x50'*, approx. 15 000 EUR), and a microscope (*'Zeiss AxioScope 40A'*, approx. 5 000 EUR). Note that not all attacks presented in this thesis require all the previously mentioned equipment. For most measurements, e.g. a cheaper oscilloscope (*'PicoScope 6404c'*, approx. 5 000 EUR) is sufficient. Therefore the value of 40 000 EUR is only given as an upper bound. The costs for most of the attacks presented in this thesis are significantly below that bound.

Next we want to shortly discuss two factors, which significantly influence the equipment costs for physical attacks. These factors are the type of physical attack and the countermeasures implemented on the attacked device to make it harder for an attacker to succeed with a physical attack.

1. **Type of physical attack:** The first factor affecting the equipment costs is the type of the physical attack. Physical attacks can be classified into *side-channel analysis (SCA) attacks* and *fault analysis (FA) attacks* on the highest level.

   SCA attacks are passive, which means that they only observe the device while performing the cryptographic computation under normal conditions. Popular side-channels are the power consumption, the electromagnetic (EM) emanation or the timing behavior. The required equipment and its costs depend on the properties of the device. The amount of leakage incorporated in the measurements dictates the required performance of the measurement devices, e.g. the oscilloscope. In case of EM attacks, the measurement location and resolution also influence the equipment cost. High-resolution measurements typically increase the success probability for the attack, but also require more expensive and specialized EM probes. Timing measurements can typically be performed with lower effort. The runtime of a cryptographic algorithm on most devices can be measured by observing the power consumption, the EM emanation, the communication interface, or some external pins.

   FA attacks, on the other hand, actively influence the operating conditions of the attacked device to force some erroneous behavior. Popular means for causing errors in non-invasive FA scenarios are modifications of the clock frequency, the power supply or the temperature. Semi-invasive fault attacks require modifications of the attacked device, e.g. decapsulation to access the inner structure. In contrast to invasive fault attacks, semi-invasive

fault attacks do not require direct contact to the inner structure. Optical attacks using laser beams can be categorized as semi-invasive attacks. Microprobing can be categorized as invasive attack. Both, the equipment costs and also the power of the attack increase from non-invasive to invasive.

2. **Implemented countermeasures:** Next to the type of the attack, also the implemented countermeasures on the targeted device can have a significant influence on the costs of the required equipment. Here we have to differ between countermeasures against SCA attacks and FA attacks.

   Hiding and masking techniques are popular countermeasures to harden devices against SCA attacks [86]. Successful SCA attacks targeting devices secured by one of the aforementioned countermeasures typically require a significantly higher amount of measurements compared to their unprotected counterpart. Next to that, the observation time for every single measurement increases in many cases because of the longer runtime of the algorithm caused by the countermeasure. To handle this increased effort the measurement setup can be improved by using e.g. oscilloscopes providing higher performance.

   To harden a device against fault attacks, Concurrent Error Detection (CED) schemes have been developed. An overview of the most-common CED schemes is given in [148]. CED can be realized by performing the critical calculation twice and the final result is only returned if the results of both calculations are identical. The two calculations can be performed in parallel (hardware redundancy) or one after the other (time redundancy). If the targeted device is protected by a CED scheme, one option for a successful FA attack is to inject a similar fault twice. This requirement has a significant impact on the required equipment. In case of optical fault injections, one might require two accurately positionable laser beams. This requirement significantly increases the equipment costs. If such an equipment is not available, safe-error attacks [149] are a second option for conducting successful FA attacks targeting devices protected by a CED scheme.

   Other countermeasures apply sensor-based approaches (e.g. light sensors or active shields for detecting decapsulation). However, these countermeasures are out of the scope of this thesis.

1.2 Contributions and Outline

Most SCA attacks and FA attacks reported in literature put the focus on the results achieved when attacking a specific algorithm. A discussion of the equipment required for performing the attack is often missing. In this thesis we address this issue by presenting low-cost measurement setups for preforming a wide range of SCA and FA attacks. All the setups exclusively use off-the-shelf

parts and their applicability has been verified by exemplary attacks. We looked at low-cost in three different settings: power/EM, fault, and relay attacks.

1.2.1 Low-Cost Power and EM Attacks

In the context of EM SCA in the RFID domain we have investigated one prototype chip for RFID applications named *CRYPTA*. This chip integrates one hardware AES module with two countermeasures. These countermeasures are the random insertion of dummy rounds and shuffling. For verifying the location-dependent EM leakage, we have chosen an ATxmega 256 microcontroller. Here we compare the EM side-channel leakage of one software AES implementation with the leakage of the integrated hardware AES module. Both implementations do not include protection mechanisms against side-channel attacks. For performing high-resolution EM measurements we target a prototype chip named *TAMPRES* ASIC. Experiments with this chip include the analysis of the AES hardware implementation and the analysis of the LRPRF[1] implementation. The AES module has no SCA countermeasures integrated. Parallel processing and a limitation in the number of different plaintexts are the mechanisms to provide side-channel security for the LRPRF module. For low-cost power analysis we target an authenticated encryption algorithm based on the Keccak-f permutation. This algorithm is implemented on a prototype ASIC and two countermeasures to harden the chip against SCA attacks are integrated. These countermeasures are hiding and secret sharing. The main contributions regarding low-cost power and EM measurement setups can be summarized as follows:

- We propose a novel measurement approach for measuring EM side-channel information of contactless RFID and NFC systems, named resolution optimization.

  This approach is presented in Chapter 2. Results of SCA attacks targeting a prototype ASIC for RFID applications show that the resolution optimization performs better compared to an existing approach if the distance between measurement antenna and attacked device exceeds 8 cm.

- The novel approach, the resolution optimization, is used to mount remote SCA attacks.

  These remote SCA attacks are discussed in Chapter 2 and they allow to measure exploitable side-channel information up to a distance of 1 meter. We show this by successfully revealing the secret key used in a cryptographic protocol implemented on the prototype ASIC for RFID applications. Parts of these results were presented at COSADE 2012 [80], at CT-RSA 2013 [78], and at ARES 2013 [81].

- We study the spatial EM leakage of a hardware and a software AES implementations on microcontrollers.

[1] Leakage-resilient pseudo-random function.

In Chapter 3, results achieved with spatial EM measurements are discussed. The chapter consists of two main contributions. First, we examine the spatial EM leakage of a software AES implementation and the hardware AES module integrated on an off-the-shelf microcontroller. Next to that we study high-resolution EM measurements for evaluating an AES and a LRPRF hardware module integrated on a prototype ASIC, respectively. The results can assist in finding the correct order of key-byte values. Results of this chapter have been published at FPS 2013 [73] and in Deliverable 5.2 of the TAMPRES project [74].

- We present first practical DPA attacks targeting a keyed KECCAK instance implemented on a taped-out ASIC.

  Results of these attacks are presented in Chapter 4. We show that the frequently proposed secret-sharing countermeasure for sponge-based algorithms does not lead to the expected security gain if applied on low-resource implementations. We verify this by applying a cryptographic ASIC named ZORRO. This ASIC implements an AE algorithm based on KECCAK. Power measurements captured with a low-cost measurement setup are used for the SCA experiments. Parts of this chapter have been published at COSADE 2015 [97].

1.2.2 Low-Cost Fault Injection

In the context of non-invasive fault injections, we target three different microcontroller types, all well-suited for usage in sensor nodes. These selection includes one ATmega 162/v, one ATxmega 256, and one ARM Cortex-M0. We do not target a specific cryptographic implementation during our investigations. The focus is put on the vulnerability of selected instructions to the fault injections. For studying semi-invasive fault injection, the target devices are one ATmega 162/v and one PIC 16F84 microcontroller. Here we study the influence of injection parameters on the ability to inject faults in volatile memory. The main contributions regarding low-cost fault-injection setups can be summarized as follows:

- We developed, improved, and applied a low-cost, FPGA-based fault board.

  This fault-board is used in Chapter 5 to tamper with the clock signal and the supply voltage for performing FA attacks targeting microcontrollers. In Chapter 6 we apply the fault-board for controlling the laser pulses in optical fault scenarios.

- We improve an existing optical fault-injection setup.

  The improvements consist in using high-power, pulsed laser diodes which allow front-side and rear-side attacks. Additionally, we use a specialized laser-diode mount to fix the diodes on top of a microscope. This allows to easily exchange the laser diodes. Using the fully-automated stepper table

it is possible to perform laser scans of the whole chip area of the device under test.

- We are among the first to study the effects of similar fault injections on two different microcontroller platforms.

  In Chapter 5, we apply non-invasive fault injections and study their impact on two different microcontroller platforms. Also the effectiveness of a combination of fault-injection methods is investigated. The results of these evaluations have been presented at FDTC 2014 [75, 77] and in the 10^{th} volume of IEEE Transactions on Information Forensics and Security [87].

- We discuss the most-important parameters influencing the success of optical fault injections.

  This evaluations can be found in Chapter 6. It turned out that the main parameters are the focus of the laser beam, the laser power and the laser pulse length. For the experiments we have applied a low-cost fault injection setup. This setup allows optical attacks from the front side and the rear side. The results have been presented at FPS 2014 [72].

1.2.3 Low-Cost Relay Attacks

The main contributions regarding low-cost relay-attack setups can be summarized as follows:

- Introducing low-cost relay attacks applying two NFC-enabled smart phones.

  With the relay setup presented in Chapter 7 relay distances of up to 110 meters can be reached by applying two/three NFC-enabled smart phones without the usage of a public network.

- We reveal and discuss limitations which arise when using smart phones for relay attacks.

  A UID, which cannot be modified and limited access to low-level commands are two limitations. By replacing one smart phone by a low-cost, custom-made proxy device, most of the revealed limitations can be circumvented.

- We compare the two most relevant relay channels, Bluetooth and WLAN.

  By performing exemplary relay attacks we evaluate the performance of the two relay channels regarding relay distance and speed.

- We discuss improvements when using the custom-made proxy device.

  These improvements include the cloning of the victim's UID, extending the relay time, modifying commands, and adding or skipping specific commands. The results relating to relay attacks have been presented at IEEE RFID 2014 [76].

2

SCA-Attacks Targeting Contactless Devices

This chapter presents side-channel measurement setups and the corresponding results targeting devices operating in the RFID and NFC domain. This means, the devices, which will be referred to as tags in the following, typically do not have an integrated power supply. The required energy is delivered contactlessly via the EM field generated by a reader device. Power measurements are only possible on prototype devices or with significant modifications of the circuit. Therefore, real-world attacks take advantage of the electromagnetic (EM) emanation of the device in order to extract side-channel information. A prototype RFID-tag chip named *CRYPTA* served as target for the side-channel experiments performed in this chapter. The results presented in this chapter have been published in [78, 80, 81] and the main contributions can be summarized as follows.

Contribution

- Comparison of two measurement approaches for RFID/NFC scenarios with the parameter measurement distance.
- Remote SCA attacks proving that exploitable leakage can be measured at distances up to one meter with low-cost equipment.
- Introduction of the term "parasitic load modulation" and verification of the existence.
- Comparison of the amount of exploitable leakage of the ASIC version of *CRYPTA* and its functional-equivalent FPGA prototype.

- Investigation of the effectiveness of the SCA countermeasures implemented on *CRYPTA*.

This chapter is structured as follows. Section 2.1 gives an introduction to the RFID domain and discusses relevant related work. Preliminary information required for the rest of the chapter is given in Section 2.2. The results for the comparison of two measurement approaches are presented in Section 2.3. Remote SCA attacks and their application are presented in Section 2.4. In Section 2.5 close-proximity measurements are performed in order to compare the exploitable leakage of the *CRYPTA* ASIC with the leakage from an FPGA prototype. In the course of this comparison also the countermeasures on the *CRYPTA* ASIC are evaluated. Section 2.6 concludes the chapter with a short discussion.

2.1 Introduction

Radio-frequency identification (RFID) technology has gained a lot of attention during the last decade and is used in many applications like ticketing, supply-chain management, electronic passports, access-control systems, immobilizers, and payment systems. The relevance of this technology is underlined by the integration of RFID functionality into the latest generation of smart phones, using so-called near-field communication (NFC). With this widespread use of RFID technology, new applications like the future Internet of Things (IoT) will arise where security plays an important role. When integrating security to RFID systems, not only the selected cryptographic algorithms have to be secure, but also their implementation has to be protected against attacks such as side-channel analysis.

An RFID system consists of a reader (e.g. a smart phone) and a tag that communicate contactlessly by means of a radio frequency (RF) field. The tag is a small microchip attached to an antenna. Passive tags also receive their power supply from the RF field, which limits the available power budget of the tags. Especially passive tags that can be produced at low cost can be used in applications like the future IoT, where tags have to be competitive in price. In order to keep the price low, tags have to be produced in high volume and with smallest possible chip size. A very limited power budget together with smallest chip size make the integration of cryptographic security to RFID tags challenging.

Recent incidents like the reverse engineering of the CRYPTO 1 algorithm in Mifare tags [101], the breaking of the Digital Signature Transponder (DST) [27], or the attacks on the Hitag 2 cipher [34] and the KeeLoq remote entry system [36] have emphasized the need for integrating strong cryptographic security to RFID tags. A lot of effort has been made by the research community to bring strong security to resource-constrained RFID tags. Well-known examples are symmetric-key schemes like the Advanced Encryption Standard (AES) [39, 49, 94] and PRESENT [110], or public-key schemes like Elliptic Curve Cryptography (ECC) [11, 14, 51, 143] and NTRU [53].

Applying a strong cryptographic algorithm alone is not enough. Also the implementation of the algorithm has to be protected against SCA attacks. There is a large number of published articles about DPA attacks on contact-based devices, but only a handful of them about attacks on RFID devices. Hutter *et al.* [54, 55] have presented several DPA attacks on high-frequency (HF) RFID prototype devices. Oren and Shamir [105] have inspected the EM emissions of ultra-high frequency (UHF) tags to deduce the secret kill password. Kasper *et al.* [65] and Oswald [106] have successfully applied SCA attacks on a contactless smart card that computes Triple DES (3DES). The authors have applied an analogue demodulation circuit in order to improve the measurements, i.e. to filter out the interfering reader field.

As a main contribution in this chapter, we present measurement setups which do not require a demodulation circuit. A comparison of the performance of our setup with the one presented in [65] reveals that especially for larger distances between measurement probe and attacked device, our approach is advantageous. It is shown that with this measurement setup, exploitable side-channel information can be measured at distances up to one meter. This is mainly achieved by applying a self-made measurement antenna customized for our needs. A relation between measurement distance and attack effort is derived in theory and approved by practical results. An AES implementation on a prototype RFID-tag chip named *CRYPTA* served as target for the experiments. Although *CRYPTA* has countermeasures against SCA attacks integrated, they were disabled for the experiments.

As a further contribution a comparison in terms of exploitable side-channel leakage of the taped-out *CRYPTA* ASIC and a functional equivalent FPGA prototype was performed. In the course of these experiments, where EM traces were measured at close proximity, the effectiveness of the SCA countermeasures were examined.

2.2 Preliminaries

In this section preliminary information required for the rest of the chapter is given. First, the investigated RFID-tag chip, the *CRYPTA* ASIC and its functional-equivalent FPGA prototype are introduced. Next, an introduction to *parasitic load modulation* is given. The parasitic load modulation describes how side-channel information of RFID chips is modulated on the reader signal. Finally, the two measurement approaches, which are compared are introduced. The first approach, the *analogue demodulation approach*, has been proposed by Kasper *et al.* in [65]. The second approach has been developed by us and we refer to it as *resolution optimization*.

2.2.1 *CRYPTA* ASIC

In the following paragraph, an introduction to the *CRYPTA* chip is given. First the focus is put on the ASIC realization of *CRYPTA* followed by its FPGA-

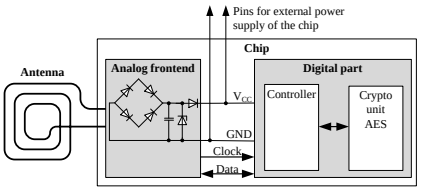


Figure 2.1: Architecture of the evaluated chip.

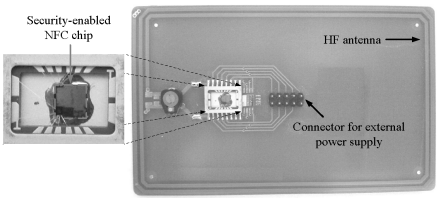


Figure 2.2: The development board with the evaluated chip.

prototype version.

The *CRYPTA*-ASIC chip is mounted in a prototype metal package which is open on top as depicted in Figure 2.3. If the two package pins are connected to an antenna, it behaves like a normal contactless chip card and can communicate with a reader device. *CRYPTA* operates in the HF frequency range of 13.56 MHz and the communication protocol is implemented according to the ISO 14443A standard [60]. The chip consists of two main parts as depicted in the architecture overview in Figure 2.1: the analog front-end (AFE) and the digital part. The antenna is connected to the AFE that provides the power supply and the clock signal to the digital part. The digital part is responsible for processing the commands to communicate with the reader. This part also contains a crypto unit with an AES implementation supporting the key length of 128 bits to provide symmetric-key cryptography. The AES part is implemented as special-purpose hardware to meet the most important requirements for RFID-tag chips: low power consumption and small chip area.

In order to increase the resistance against SCA attacks, the AES implementation has two countermeasures integrated: the insertion of dummy rounds and shuffling. The chip processes in total 25 rounds during an AES encryption/decryption. Ten rounds relate to the real computation of AES and fifteen rounds are dummy rounds that process random data. The dummy rounds are randomly inserted before the first and after the last real AES round. With shuffling, the processing order of the bytes of the state is randomized. As the AES state consists of sixteen bytes every byte can be processed at sixteen different points in time. In a typical DPA-attack scenario it is important to know at which point in time a specific byte of the state is processed. Because of that fact shuffling increases the attack complexity. It is important to mention that *CRYPTA* allows to switch off the countermeasures. This option is advantageous for several of our investigations.

Figure 2.2 depicts the ASIC chip mounted on a development board that contains an antenna with four windings. The board also allows to power the chip with an external power supply. If an external power supply with a voltage of at least 3.3 V is connected, the chip does not use the power supply extracted from the reader field. This allows to measure the power consumption of the chip with a resistor in the ground line.

In addition to the ASIC-chip version we also use an FPGA-prototype version

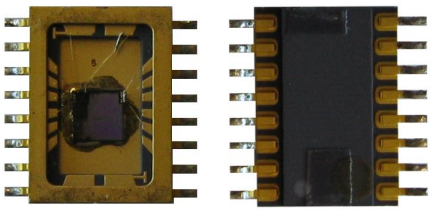


Figure 2.3: The CRYPTA ASIC-chip.

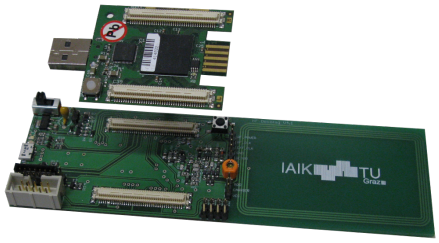


Figure 2.4: The IAIK demotag consisting of the FPGA (top) and the main board (bottom).

of *CRYPTA* for the evaluation. This FPGA-prototype version is based on the IAIK demotag which consists of two parts. The first part is the FPGA (XILINX Spartan-3) itself, which implements the digital part of *CRYPTA*. The second part is the so-called main board, which consists of the analog front-end and the antenna. The IAIK demotag is depicted in Figure 2.4. For a reader device, the FPGA-prototype tag appears like a regular, passive RFID tag. It uses an external power supply but the reader field is used for communication and for extracting the clock signal. We used the FPGA-prototype version to show that the DPA-attack results achieved with this device are comparable with the results from the ASIC-chip versions. Therefore, one output pin was used as trigger pin to signalize, when the AES encryption starts. One advantage of the FPGA-prototype version is that small modifications can be integrated and tested with low effort. The FPGA prototype further gives the ability to correct bugs detected on the real chip and evaluate the effects of the modification and chip developers can test the implementation before manufacturing the ASIC chip.

2.2.2 Parasitic Load Modulation

For remote SCA attacks we present an approach that requires neither the separation of the tag chip from its antenna nor the application of special analog preprocessing circuits. In contrast to close-proximity measurements we exploit the strong reader field as a carrier of the weak data-dependent information emitted by the tag for the remote SCA-attack scenarios. As we assume that most of the data-dependent information is amplitude modulated on the reader signal, it is sufficient to measure only the peaks of the reader signal. This simple measurement concept was originally used for analyzing the emissions of UHF tags [109]. In this work we show that this concept is also suitable for gathering the data-dependent emissions of RFID/NFC tags operating in the HF range, even at greater distances.

NFC and HF RFID systems are inductively coupled. This means that the antennas of reader and tag are loosely coupled and act like an air-core transformer [40], which is illustrated in Figure 2.5. Applying an alternating voltage at the reader antenna (U_{Reader}) results in a magnetic field that itself induces

an alternating voltage at the tag antenna (U_{Tag}). The voltage at the tag is not only used for data transmission from the reader to the tag (i.e., by modulating the reader field in step with the data), but also to provide the power supply for passive tags. Data transmission from the tag to the reader is done by so-called load modulation, where an impedance Z_{Mod} is switched in step with the data. Switching the impedance Z_{Mod} changes also the overall impedance of the tag Z_{Tag}, which in turn results in changes in the magnetic field and in detectable voltage variations at the reader antenna.

However, not only the intended load modulation influences the magnetic field, but also changes in the power consumption of the tag chip. As the power consumption directly relates to the chip's effective impedance Z_{Chip}, the overall impedance of the tag is changed as well. In that way data-dependent information present in the power consumption of the tag chip is modulated on the reader field. We call this effect *parasitic load modulation*, according to a similar effect named *parasitic backscatter* that was observed by Oren and Shamir for tags operating in the UHF range [105]. In the following, we use this *parasitic load modulation* for conducting remote SCA attacks targeting the *CRYPTA* tag.

It is not obvious that remote attacks are applicable on HF tags. First, HF tags are inductively coupled and operate in the near field where RF signals are attenuated with $1/d^3$ (with d being the distance). UHF tags on the other hand operate in the far field, where RF signals are only attenuated with $1/d$. Moreover, the *parasitic backscatter* observed by Oren and Shamir does not influence the reader field, rather it relates to independent electromagnetic waves emitted by the tag antenna. Consequently, a favorable placement of the measurement antenna is possible that allows to gather mainly the signal emitted by the tag antenna. However, this is not possible when measuring the *parasitic load modulation* of a tag, as it is directly modulated on the strong reader field.

2.2.3 Resolution Optimization

A special recording technique has been applied which does not require the application of any analog preprocessing circuits. Due to *parasitic load modulation*, recording only the peaks of the reader signal should be sufficient and further increase the resolution of the measurement. The upper plot in Figure 2.6 shows one trace where the whole amplitude is recorded and the lower plot shows the same trace zoomed into the peaks. For this approach it is also sufficient to store only the maximum value per period, where the period equals $\frac{1}{13.56MHz} = 73.75ns$. The following experiment shows that this method is similar to an analog demodulation followed by a lowpass filtering. For this experiment we have applied these two steps on the trace from the top plot of Figure 2.6. The two steps were performed in MATLAB®. Using the build-in function `amdemod` we demodulated the trace. Afterwards a lowpass filter was applied on the demodulated trace using the function `filtfilt` with a cut-off frequency $f_{co} = 15\,MHz$. The comparison plot at the bottom of Figure 2.6 clearly depicts that there are only small deviations in the results of the two approaches. It turned out that except of the downsampling step no other preprocessing steps (e.g., alignment, filter-

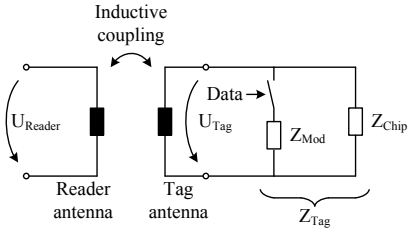


Figure 2.5: Basic principle of inductive coupling between reader and tag antenna.

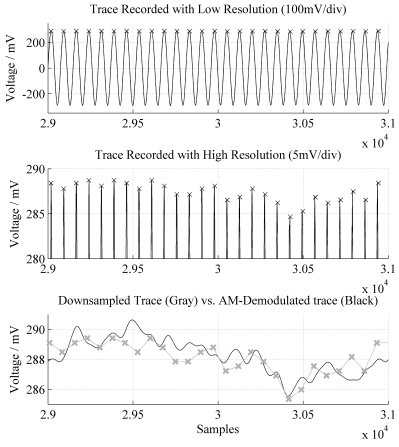


Figure 2.6: Resolution enhancement of EM measurements.

ing) are required in order to perform successful DPA attacks. For enhancing the resolution of the measurements, the voltage-scaling setting *voltage per division* (V/div) of the oscilloscope has been used. With increasing resolution also the number of different voltage values increases. As a result of only measuring the peaks of the signal the voltage values are fine grained what increases the efficiency of the DPA attack.

In the following, we define a factor f_{ws}, which brings the V/div setting of the oscilloscope and the peak-to-peak voltage U_{pp} of the EM signal in relation. This factor can be interpreted as **window size** on the voltage axis, describing the percentage of the signal being recorded with the current settings (N_{div} equals the number of divisions of the used oscilloscope on the voltage axis):

$$f_{ws} = \frac{V/div \cdot N_{div}}{U_{pp}} \cdot 100\,[\%] \tag{2.1}$$

Besides the **window size**, the **window position** is the second parameter which has to be considered when using the resolution optimization approach. This parameter is set using the *voltage offset* setting of the oscilloscope and defines the part of the signal actually recorded. Both parameters, size and position of the recording window need to be found during a profiling phase. It is comparable to finding the correct time interval for successfully mounting a DPA attack.

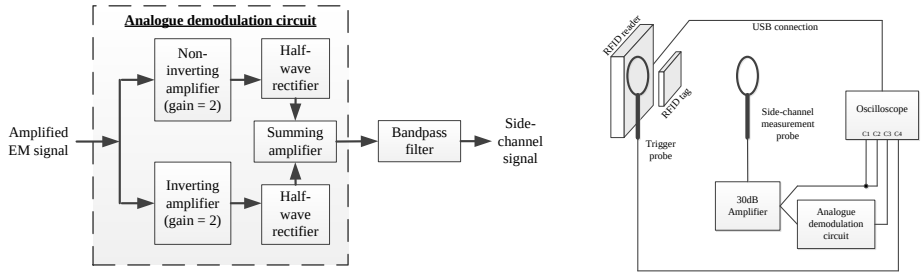


Figure 2.7: Analogue demodulation circuit (left) and setup used for comparing two measurement approaches (right).

2.2.4 Analogue Demodulation Approach

In the following the analogue demodulation approach as published in [65] is explained. In a first step, the leakage model is established according to Equation 2.2.

$$s(t) = (P_{const} + p(t)) \cdot cos(\omega_{reader} \cdot t) \tag{2.2}$$

$s(t)$ denotes the reader signal, which can be measured by an adversary. It is amplitude modulated with a constant part of the power consumption P_{const} and a time-varying part $p(t)$. This time variations can be caused by the execution of different operations on the tag or by executing a similar operation on different values. Furthermore, $\omega_{reader} = 2\pi \cdot f_{reader}$ with $f_{reader} = 13.56\,\text{MHz}$ being the carrier frequency. $p(t)$ includes the relevant side-channel information to carry out side-channel attacks. Typically $P_{const} >> |p(t)|$, this fact makes it a crucial task to filter the exploitable side-channel information out of $s(t)$. The authors of [65] propose to use an analogue demodulation circuit to perform this task. A block diagram of an exemplary analogue demodulation circuit is depicted in the left part of Figure 2.7. In a first step, the EM signal measured with the EM probe is split up into two parts. Both parts are then amplified, one with a gain equal 2 and one with a gain equal -2. Next, the negative signal parts are removed using half-wave rectifiers. In a last step, both signals are summed up again and the result can already be used as side-channel signal. To further improve the measurement process, a bandpass filer can be applied to suppress the DC part of the demodulated signal on the one hand and high frequencies carrying no information on the other hand.

2.3 Comparison of Measurement Approaches

In the following section the results of the comparison of the two previously introduced measurement approaches are discussed. By performing experiments with measurement distances between 3 cm and 10 cm the influence of the measurement distance on the two approaches has been studied. It figured out that the

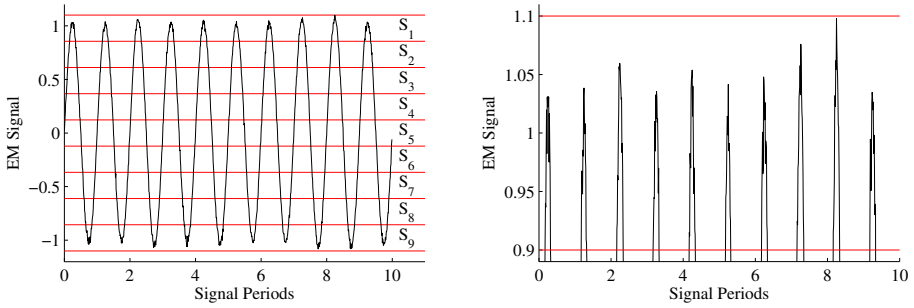


Figure 2.8: Measurement of the full EM signal (left) and improved resolution by only measuring the peaks of the signal (right).

resolution optimization approach is advantageous for measurements at distances exceeding 8 cm. The results of this approach can further be improved by applying a moving-average filter on the measured traces. The same filter applied on the measurements from the analogue demodulation circuit did not further improve the results compared to unfiltered measurements. A disadvantage of the resolution optimization is the requirement for a profiling step in order to find the best parameters for window size and position. Before the achieved results are discussed in detail, the measurement setup is introduced.

2.3.1 Measurement Setup

A schematic view of the measurement setup is depicted in Figure 2.7 on the right. The measurements are controlled by a *'LeCroy WP 725 Zi'* oscilloscope. This device is not only used for measuring the side-channel information, but it also establishes the communication with the RFID tag. The communication consists of a simple command triggering an AES encryption on the RFID tag. The *CRYPTA* chip was used as device under test. A trigger probe (model '*LF R 400*' from '*Langer EMV Technik*') placed in close proximity to the reader provided the trigger information. For measuring the side-channel signal, a self-made loop antenna with $N_{ant} = 5$ windings and a diameter $d_{ant} = 8$ cm has been applied. Different values for N_{ant} as well as for d_{ant} have been investigated, but the best results have been achieved with the values mentioned above. Experiments using a second '*LF R 400*' EM probe revealed that the diameter of this probe (2.5 cm) is too small for meaningful measurements at higher distances. The signal measured with the self-made antenna has been amplified with a 30 dB amplifier. The output of the amplifier is directly connected to two inputs of the oscilloscope and to the input of the analogue demodulation circuit. With the third channel of the oscilloscope the output signal of the analogue demodulation circuit is captured. Capturing the amplified EM signal twice allows to evaluate two different oscilloscope settings (*voltage/division, offset*) in parallel.

2.3.2 Profiling for Resolution Optimization

In contrast to the amplitude demodulation approach, the resolution optimization as applied for the remote SCA attacks requires an initial profiling phase. Results of this profiling phase are the optimal **window position** (adjusted by the *voltage offset* setting of the oscilloscope) and the **window size** (set by the *voltage/division* setting of the oscilloscope).

In order to find the best-fitting **window position**, the following approach is applied. In a first step, the peak-to-peak value of the EM signal U_{pp} is split into M equally-sized segments $S_1 \dots S_M$. Using the *voltage offset* and the *voltage/division* setting of the oscilloscope allows to select each segment separately for recording. The left plot in Figure 2.8 depicts one recorded EM signal measuring the whole amplitude. Furthermore, the $M = 9$ trace segments S_1 to S_9 are marked. For the right plot of Figure 2.8 the settings of the oscilloscope were modified in order to measure segment S_1. For each segment, $N = 20\,000$ measurements were recorded each including one AES execution. The same set of input values for the AES were used for all segments and the key was fixed and known for all the experiments. After the measurement process, a DPA attack was performed on each segment $S_i, i = 1 \dots M$ separately, leading to a maximum correlation coefficient for the correct key hypothesis $\rho_{max,i}$ for each segment. The segment containing most side-channel information is found by evaluating $S_m | m = argmax(\rho_{max,i})$. Evaluations at two different distances between reader and measurement antenna both yield segments S_1 and S_9 as the best window positions. Results are depicted in Figure 2.9. Concluding this experiment, most of the exploitable side-channel leakage for the specific device appears in the peaks of the signal.

In order to find the best-matching value for f_{ws}, i.e. the value maximizing the amount of leakage in the measurements, the following approach is used. For a fixed distance between measurement antenna and tag of 3 cm, $M = 9$ trace sets containing 20 000 traces were recorded for f_{ws} settings between 2 % and 90 % by choosing the appropriate *voltage/division* setting on the oscilloscope. The window position for all window sizes was chosen to include the positive peaks of the EM signal according to the outcome of the previous experiment. DPA attacks were performed using each trace set separately, leading to a maximum correlation coefficient for the correct key hypothesis $\rho_{max,i}$ for $i = 1 \dots M$. The best f_{ws} value equals the setting which maximizes the correlation value. The result of this experiment is depicted in the right plot of Figure 2.9. The maximum correlation coefficient values for the analyzed f_{ws} settings are shown. $f_{ws} = 3\%$ turns out to lead to the best results. It might be intuitive to use the smallest-possible value for f_{ws} but results show that this is not the case. We conclude that too small values for f_{ws} chop parts of the trace containing exploitable leakage.

2.3.3 Results of the Comparison

After the proper settings for window position and window size for the resolution optimization approach have been found, the comparison between the two mea-

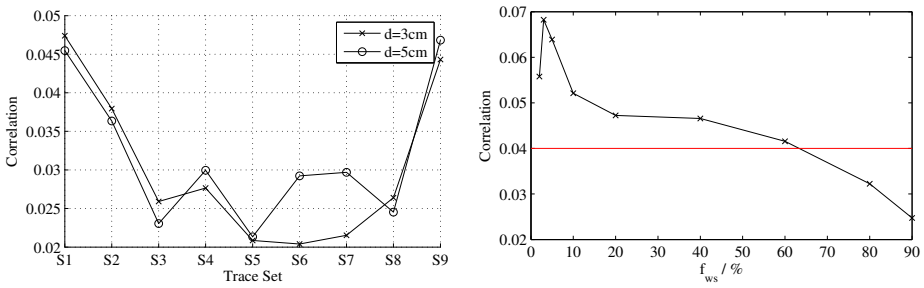


Figure 2.9: Correlation coefficients for the correct key hypothesis for different window positions (left). Correlation coefficient as a function of the window size f_{ws} (right).

surement approaches resolution optimization and analogue demodulation has been performed. In order to analyze the impact of the signal amplitude, distances between $d_{start} = 3\,\text{cm}$ and $d_{end} = 10\,\text{cm}$ with a step size of $\Delta d = 1\,\text{cm}$ have been investigated. At each distance d_i, $N = 20\,000$ traces were recorded twice by applying both measurement approaches. DPA attacks using the correlation coefficient as distinguisher for each trace set yield a maximum correlation coefficient for the correct key hypothesis $\rho_{AM}[i]$ for the analogue demodulation approach and $\rho_{ResOpt}[i]$ for the resolution optimization approach, respectively. The results of this experiment are summarized in Table 2.1 and depicted in Figure 2.10.

In order to evaluate the attack performance for every distance we additionally used the *nearest-rival distinguishing power (nrdp)* measure as presented in [147]. This value is calculated according to Equation 2.3. $\rho_{maxCorrect}$ is the correlation value of the correct key hypothesis, $\rho_{maxWrong}$ is the maximum correlation value of the wrong key hypotheses and σ relates to the standard deviation. The number of traces used for the attack is N. If $nrdp > 0$ the attack is successful, else ($nrdp \leq 0$) the correct key hypothesis cannot be distinguished from wrong key hypotheses.

$$nrdp = \frac{1}{\sigma} \cdot (\rho_{maxCorrect} - \rho_{maxWrong}); \quad \sigma = \frac{1}{\sqrt{N}} \tag{2.3}$$

For the resolution optimization approach, a significant increase of the correlation coefficient was achieved by applying a post-processing step on the recorded traces. The post-processing step consists of a moving average filtering. Different numbers of sample points for calculating the average value have been investigated. The best results were achieved by including 75 % of the sample points of one clock period of the attacked device. The same post-processing technique did not influence the result for the analogue demodulation approach. Results achieved with post processing are plotted with solid graphs while results achieved without post processing are plotted with dashed graphs in Figure 2.10. By analyzing the graphs, the following conclusions can be drawn:

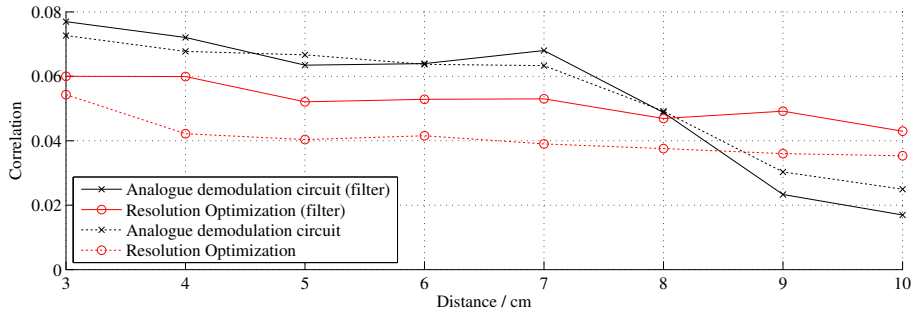


Figure 2.10: Correlation coefficient for the correct key hypothesis for the two measurement approaches without and with post processing as a function of the measurement distance.

Impact of the measurement distance Up to a distance of 8 cm, the analogue demodulation outperforms the resolution optimization. For distances exceeding 8 cm, the resolution optimization leads to better results. According to Table 2.1, the peak-to-peak voltage U_{pp} of the measured EM signal equals 310 mV. This voltage values are too small to allow a proper operation of the half-wave rectifier used in the analogue demodulation circuit. This leads to the decreased performance at higher distances.

Impact of post processing Moving-average filtering does not improve the result of the analogue demodulation approach because the applied circuit already performs a bandpass filtering. That means that subsequent samples are already combined by hardware making the software-filtering unnecessary.

2.4 Remote SCA Attacks

In the previous section it was shown that the resolution optimization is a well-suited approach for performing so-called remote SCA attacks. In this section we show that this setup allows to extract exploitable side-channel information

Table 2.1: Input-voltage values for the analogue demodulation circuit and correlation values.

Distance	cm	3	4	5	6	7	8	9	10
U_{pp}	mV	1670	1185	805	560	405	310	236	185
ρ_{AM}		0.073	0.072	0.063	0.064	0.068	0.049	0.030	0.025
$nrdp$		6.61	5.80	4.99	4.80	5.48	1.87	-0.08	-0.39
$\rho_{AM,sum}$		0.077	0.068	0.067	0.064	0.063	0.049	0.023	0.017
$nrdp$		7.62	6.54	6.16	5.43	6.06	2.29	-0.42	-0.48
ρ_{ResOpt}		0.054	0.042	0.040	0.041	0.039	0.038	0.036	0.035
$nrdp$		4.43	1.43	2.15	2.15	1.93	0.86	0.76	0.74
$\rho_{ResOpt,Sum}$		0.060	0.060	0.052	0.053	0.053	0.047	0.049	0.043
$nrdp$		5.09	5.01	3.60	3.75	3.97	2.26	2.63	1.34

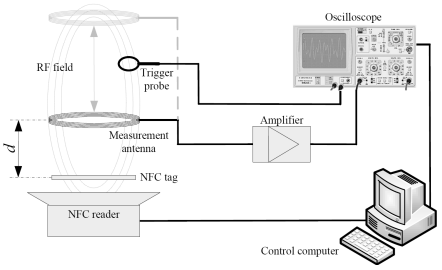


Figure 2.11: The schematic measurement setup for remote SCA attacks.

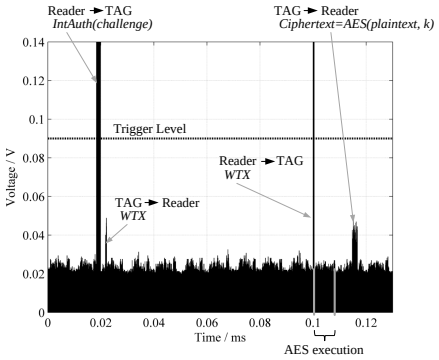


Figure 2.12: Sequence of commands for the internal authentication using AES.

up to a distance of one meter by only applying low-cost equipment. RFID tags operating in the frequency range of 13.56 MHz (HF tags) can be targeted with this setup and the results presented in the following are achieved by targeting the *CRYPTA* chip with deactivated countermeasures. Successful attacks revealing the AES key of the attacked device show that even though the communication range of typical RFID and NFC systems is limited by approximately 10 cm, data-dependent information can be measured at distances up to 1 m by taking advantage of the *parasitic load modulation*. Applying low-cost equipment is sufficient for succeeding with the attacks. The vulnerability of HF tags to remote side-channel attacks highlight the importance of integrating appropriate countermeasures.

2.4.1 Measurement Setup and Post Processing

In this section the measurement setup enabling remote SCA attacks is introduced. Furthermore, required post processing steps on the recorded traces that turned out to be necessary for achieving appropriate results, are discussed. One important remark is that our setup does not require any additional circuits like a signal-cancellation circuit or a demodulation circuit (c.f. [65]). We only use an amplifier at the measurement antenna for larger distances between tag and measurement antenna in order to increase the amplitude of the measured signal.

In Figure 2.11 the schematic measurement setup is depicted and Figure 2.13 shows a picture taken during a measurement at a distance of 100 cm. A Tagnology TagScan RFID reader for communication with the tag and the *'LeCroy LC 584'* oscilloscope for measuring the EM signal are connected to a workstation. This workstation controls the measurement process by storing the traces and sending commands to the NFC tag using MATLAB® scripts. Furthermore we use the same self-made loop antenna as for the previous experiments with $N_{ant} = 5$ windings and $d_{ant} = 8\,cm$ in order to measure the reader signal. For

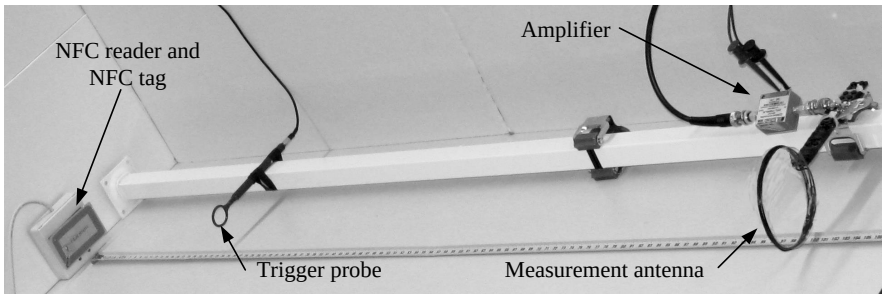


Figure 2.13: Measurement setup for d = 100 cm.

amplifying the signal measured with the antenna a broadband amplifier with a gain of 30 dB was used. A small deviation compared to the previous measurement setup consists in adding an RC-matching circuit between measurement antenna and amplifier. It turned out that this matching circuit increases the signal quality and therefore the attack performance.

For increasing the practical relevance, the timing information indicating the start for the recording of the EM trace (the trigger signal) was also extracted out of the EM signal. Therefore a pattern in the communication between reader and tag has been used, this pattern is depicted in Figure 2.12. This figure shows an EM trace of the whole *Internal Authenticate (IntAuth)* procedure, which triggers an AES encryption. The time interval where the AES encryption takes place is highlighted in the trace. This part was recorded in order to perform the DPA attacks. For recording the traces the trigger probe was placed at a distance of 25 cm. In order to reduce the effort we have fixed the trigger probe (model '$LFR400$' from '*Langer EMV Technik*') at $d = 25\,cm$ for all measurements. However, as Figure 2.12 illustrates, commands sent from the reader to the tag that are used for triggering can be clearly identified in the trace (tag answers are smaller but can also be easily identified). Hence, placing the trigger probe at larger distances is also possible. Experiments yielded that the trigger information can be easily detected at distances exceeding 100 cm.

Next, the focus is put on the relationship between the amplitude of the measured reader signal U_{pp} and the distance d between reader and measurement antenna. According to [40] this relationship can be described with the following equation: $U_{pp} \approx \frac{1}{d^3}$. The measurements of the amplitude of the reader signal at distances d between 25 cm and 100 cm confirmed the theory. Figure 2.14 illustrates the performed comparison of measured values and values calculated based on the equation given above. The black graph corresponds to values calculated using the equation and the data points marked with 'x' correspond to measured values.

The angular offset of the measurement antenna also has a crucial impact on the measured signal strength, this relationship is discussed in the following. To examine this relationship, U_{pp} was measured for angles between 0° (i.e., reader antenna and measurement antenna are coaxial as shown in Figures 2.11

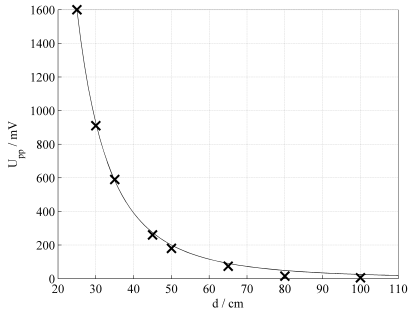


Figure 2.14: Peak-to-peak voltage (U_{pp}) of the reader signal as a function of the distance between reader and measurement antenna ($U_{pp} = f(d)$).

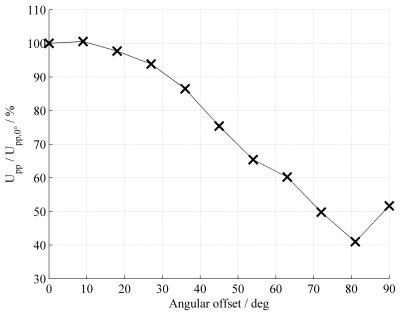


Figure 2.15: Peak-to-peak voltage (U_{pp}) of the reader signal as a function of the angular offset ($U_{pp} = f(angular\,offset)$).

and 2.13) and 90° with a step size of 9° at a constant distance $d = 30\,cm$. The results are summarized in Table 2.2. Figure 2.15 depicts the relationship between angular offset and voltage where the values of the y-axis are normalized to U_{pp} at 0° ($U_{pp,0°}$). At the angle of 90° an increase of the voltage value compared to the voltage value measured at 81° can be observed, what is against the trend of decreasing voltage values for increasing angular offset. We assume that this effect is mainly caused by the geometry of the reader antenna and the measurement antenna. We did not further investigate this effect because all the following experiments were performed at an angular offset of 0°. Additionally, at 90° the U_{pp} value can be increased from 470 mV to 618 mV by rotating the measurement antenna by 90° (i.e., reader antenna and measurement antenna are coplanar). In general, the influence of the angular offset highly depends on the antenna design of the used reader.

2.4.2 Verification of the Parasitic Load Modulation

With increasing distance between *CRYPTA* chip and measurement antenna, the assumption that the EM emanation of the chip is modulated on the reader signal and that it is not the direct emanation of the chip, becomes more important. In

Table 2.2: Relation between U_{pp} and angle for $d = 30\,cm$.

Angle	U_{pp}		*Angle*	U_{pp}		*Angle*	U_{pp}		*Angle*	U_{pp}	
deg	*mV*	%	*deg*	*mV*	%	*deg*	*mV*	%	*deg*	*mV*	%
0	910	100	27	854	94	54	595	65	81	373	41
9	915	101	36	787	86	63	548	60	90	470	51
18	889	98	45	686	75	72	453	50			

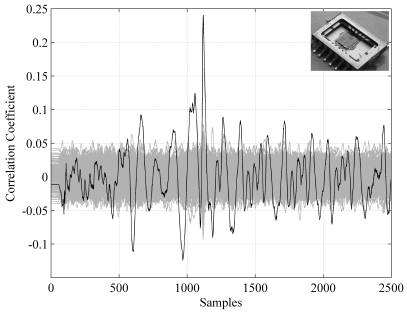


Figure 2.16: DPA-attack result with opened chip housing.

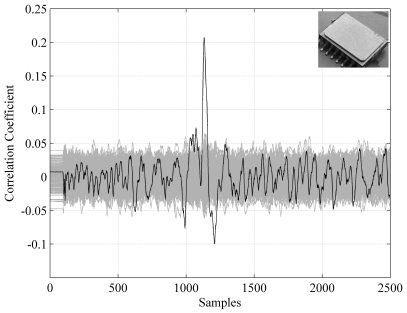


Figure 2.17: DPA-attack result with closed chip housing.

order to verify this assumption we first performed attacks at a low distance of only 7 cm for two scenarios. In the first scenario the chip housing was opened and in the second scenario it was closed (shielding the direct EM emanation of the chip). 20 sets each containing 5000 EM traces were recorded for each scenario and a DPA attack was performed on each set. Next the mean ($\bar{\rho}_{opened}$, $\bar{\rho}_{closed}$) and the standard deviation (σ_{opened}, σ_{closed}) of the highest correlation values were calculated for both scenarios yielding to the following results: $\bar{\rho}_{opened} = 0.246$, $\sigma_{opened} = 0.032$, $\bar{\rho}_{closed} = 0.244$, $\sigma_{closed} = 0.025$. Figure 2.16 shows one DPA-attack result for the scenario with opened chip housing and Figure 2.17 shows one DPA-attack result for the scenario with closed chip housing. Gray traces correspond to wrong key hypotheses and black traces correspond to the correct key hypothesis. Small variations in the highest correlation values can be observed but the statistical analysis yield that the direct emanation of the chip does not influence the results significantly.

2.4.3 Remote SCA-Attack Results

In the following the results of the remote DPA attacks for high distances are presented. We have performed attacks for the following distances: 25 cm, 35 cm, 45 cm, 50 cm, 65 cm, 80 cm, and 100 cm.

For the following experiments, the resolution optimization approach was applied, so the first step was finding appropriate parameters for **window position** and **window size**. For the best window position we already had the information from the profiling phase in the last section. In order to find the best window size for the remote SCA attacks, f_{ws} was examined for $d = 35\,cm$. The result of this experiment is depicted in the upper plot in Figure 2.18. The left-most data-point corresponds to the highest possible resolution which can be achieved with the used oscilloscope, namely $2\,mV/div$. With $U_{pp} = 590\,mV$ for this distance 3.4 % of the amplitude are recorded as a consequence ($f_{ws} = 3.4\,\%$). The important observation here is that the highest resolution does not lead to the attack with the highest correlation coefficient. Using $f_{ws} = 8.5\,\%$ leads to the best results.

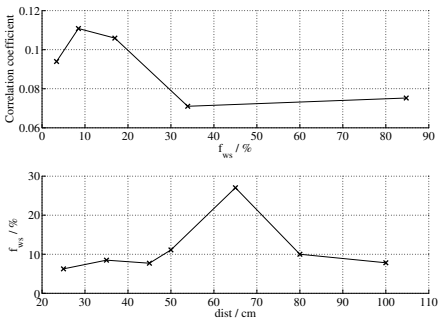


Figure 2.18: Upper plot: correlation coefficient as function of f_{ws} ($d = 35\,cm$). Lower plot: used f_{ws} values for analyzed distances.

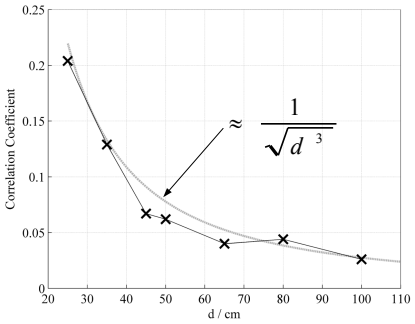


Figure 2.19: The resulting correlation coefficient ρ for the analyzed distances.

This observation can be explained by the fact that relevant information is cut off if f_{ws} is too small. In the case of $d = 35\,cm$ $f_{ws} = 8.5\,\%$ corresponds to a resolution of $5\,mV/div$. In the lower plot in Figure 2.18 the used values for f_{ws} for the different distances are depicted. The high f_{ws} value of over 25 % for $d = 65\,cm$ appears because with the highest resolution of the oscilloscope ($2\,mV/div$) and the U_{pp} value for that distance a smaller value was not achievable. In order to achieve f_{ws} values in the region around 10 % for higher distances (smaller U_{pp} values), a second amplifier stage was used to increase the gain. With this modification f_{ws} values around 10 % can be reached again at $d = 80\,cm$ and $d = 100\,cm$.

The plot in Figure 2.19 shows the correlation coefficient for the correct key hypothesis for the analyzed distances. There is a big descent for the value of ρ between 25 cm and 45 cm: $\Delta\rho_{25cm-45cm} = \rho_{25cm} - \rho_{45cm} = 0.129$. The difference of the ρ values between 45 cm and 100 cm is comparatively small: $\Delta\rho_{45cm-100cm} = \rho_{45cm} - \rho_{100cm} = 0.049$.

Comparing Figure 2.14 and Figure 2.19 shows that the relations $U_{pp} \leftrightarrow d$ and $\rho \leftrightarrow d$ are similar. This similarity can be described as follows: The power consumption and as a consequence also the EM emanation of a device at each point in time depends on a noise part P_{noise}, a constant part P_{const} and the exploitable part P_{exp} as explained in the book of Mangard *et al.* [86]. The total power consumption in every point in time is the sum of these three parts according to Equation 2.4.

$$P_{total} = P_{noise} + P_{const} + P_{exp} \tag{2.4}$$

With the information given above the signal-to-noise ratio SNR can be calculated. The SNR is defined as the ratio between the variance of the signal and the variance of the noise. As $Var(P_{const}) = 0$, the SNR can be calculated using Equation 2.5.

$$SNR = \frac{Var(P_{exp})}{Var(P_{noise})} \tag{2.5}$$

The higher the SNR at the targeted point in time for an attack is, the better are the results of the correlation attack (higher correlation value ρ). In [86] the relation between ρ and SNR is given according to Equation 2.6. One important remark is that the approximation given in Equation 2.6 is only valid for small values of SNR and for $\mid \rho \leq 0.2 \mid$. For the scenario presented in this work these limitations hold.

$$\rho \approx \sqrt{SNR} \tag{2.6}$$

In a next step relations between d and $Var(P_{exp})$ as well as between d and $Var(P_{noise})$ have to be found. In our model P_{exp} can be seen as the exploitable part of the EM signal of the chip which is modulated on the reader signal. As we could show in the previous section, the relation between U_{pp} of the reader signal and d is the following: $U_{pp} \approx \frac{1}{d^3}$. As a result also the variations caused by P_{exp} decrease with the factor $\frac{1}{d^3}$. So the first observation is that $Var(P_{exp}) \approx \frac{1}{d^3}$. Next the focus is put on P_{noise}. In our remote scenario P_{noise} can be split up into two parts, $P_{noiseIC}$ and $P_{noiseENV}$. $P_{noiseIC}$ is the noise part contributed from the chip and $P_{noiseENV}$ is the environmental noise recorded with the antenna. $P_{noiseENV}$ is independent of d and $P_{noiseIC} \approx \frac{1}{d^3}$ and it can furthermore be assumed that $P_{noiseENV} \gg P_{noiseIC}$. Combining the upper results the relation between SNR and d and as a consequence also the relation between ρ and d (using Equation 2.6) can be given according to Equation 2.7. This theoretical result confirms our practical measurements (cf. Figure 2.19).

$$SNR \approx \frac{1}{d^3} \rightarrow \rho \approx \sqrt{SNR} \approx \frac{1}{\sqrt{d^3}} \tag{2.7}$$

2.4.4 Discussion of the Results

Table 2.3 provides an overview of the results achieved with the remote SCA attacks for distances between 25 cm and 100 cm. The values for the *nearest-rival distinguishing power (nrdp)* show that the attacks for all distances allow to distinguish the correct key from wrong key guesses. The correlation coefficient $\rho_{maxCorrect}$ decreases according to Equation 2.7 with increasing distance. As a result the number of traces used for the attack in order to achieve correct results increases. The starting point of the analyses was $d = 25\,cm$ and the distance was increased as soon as the SCA-attack result was expressive. This leads to the different number of traces for the different attack distances. Using a different number of traces decreases the comparability between the attacks on the one hand. On the other hand the achieved results are sufficient in order to confirm the theoretical assumptions like the relation between ρ and d given in Equation 2.7.

In order to achieve f_{ws} values of 10 % we have used a second amplifier stage for the distances 80 cm and 100 cm. This explains the increased U_{pp} values given

in Table 2.3 for these two distances. By examining the results for $d = 65\,cm$ and $d = 80\,cm$, the importance of the correct setting for f_{ws} can be seen. The optimal value of 10 % for f_{ws} has been found during the profiling phase. A big deviation to this optimal value, as it is the case for $d = 65\,cm$ leads to a worse DPA-attack result. This is reflected in the values of $\rho_{maxCorrect}$ and $nrdp$.

For our remote SCA attacks we have placed reader and tag close to each other. However, in a real-world attack scenario, it would be advantageous for an attacker to place the reader also at a certain distance from the tag. In that way the whole attack can be applied completely remotely without being noticed by the tag owner. As demonstrated by Kfir *et al.* [67], remotely powering and also communicating with an NFC tag up to ranges of 40 cm can be easily realized with low-cost equipment (below 100 $). Kfir *et al.* achieved this range extension by using a larger reader antenna and by increasing the strength of the reader field. Using such a setting a large amount of traces can be recorded unnoticed by the owner of the NFC tag. For HF tags operating in the so-called vicinity range (e.g., according to ISO15693 [59]) that can anyway achieve larger communication ranges of up to 1.5 m, such remote attacks are even much easier to conduct as no modification of the reader device is necessary.

2.5 Close-proximity Measurements

In this section results regarding the comparison of the exploitable SCA leakage for the *CRYPTA* ASIC and the corresponding FPGA prototype are presented. For this comparison the EM emanation of both devices has been measured at close proximity, i.e. directly above the chip surface. The same measurement setup as introduced in the previous sections has been applied with the exception of a different EM measurement probe. The self-made loop antenna has been replaced by a '*LF-B 3*' EM probe from '*Langer EMV Technik*'. This probe features a better spatial resolution and is therefore better-suited for close-proximity EM measurements. In the course of the close-proximity measurements, also the SCA countermeasures implemented on the *CRYPTA* ASIC have been verified.

Table 2.3: SCA-attack results achieved at the analyzed distances (for $d = 80\,cm$ and $d = 100\,cm$ a second amplifier stage was used).

d	U_{pp}	*Resolution*	f_{ws}	*Traces used*	$\rho_{maxCorrect}$	$nrdp$
cm	mV	$\frac{mV}{div}$	%			
25	1 600	10	6.25	3 000	0.204	10.95
35	590	5	8.47	3 000	0.128	5.48
45	260	2	7.69	4 500	0.074	0.40
50	180	2	11.11	9 000	0.062	3.51
65	74	2	27.03	14 000	0.039	0.70
80	1 000	10	10.00	14 000	0.043	3.67
100	640	5	7.81	30 000	0.025	0.51

2.5.1 Comparison of ASIC and FPGA Version

DPA attacks targeting the ASIC version and the FPGA version lead to similar results. 47 (ASIC) and 55 (FPGA) measurements are sufficient for clearly distinguishing the correct key hypothesis from the wrong key hypotheses. In order to reach this result targeting the ASIC version a small modification in the measurement setup was necessary to minimize the influence of the strong reader field on the small EM emanations produced by the chip. The chip has therefore been separated from the antenna and both parts have been connected via thin copper wires with a length of 25 cm. This step has been proposed by Carluccio *et al.* [32], Hutter *et al.* [54] and allows to place the chip outside of the reader field while the communication with the RFID reader is still possible.

2.5.2 *CRYPTA* Countermeasure Evaluation

With the minimum number of required measurements for a successful DPA attack targeting the unprotected AES implementation on the *CRYPTA* ASIC it is possible to approximate the number of required measurements for the attacks with activated countermeasures. The insertion of 15 additional dummy rounds increases the number of required measurements by a factor of 256 and the same factor has to be considered for the shuffling countermeasure. Overall the number of required measurements for a successful DPA attack can be approximated with $47 \cdot 256 \cdot 256 = 3.1 \cdot 10^6$ measurements. Because of the strict timings defined by the ISO protocol for the communication between reader and tag, the recording time for one measurement was lower-bounded to approximately 500 ms. For recording $3.1 \cdot 10^6$ measurements this would result in a recording time of 18 days. We did not want to spend that amount of time for measuring, so we closer analyzed the countermeasures with the goal to find some weaknesses to decrease the recording effort.

A template-based approach allowed to distinguish dummy rounds from real rounds. This enables us to completely circumvent the dummy round countermeasure. In every single measurement, the first AES round working on the real state can be extracted. DPA attacks on this new trace set require approximately $47 \cdot 256 = 12 \cdot 10^3$ measurements, because the shuffling countermeasure is still active. By applying windowing [86], a well-known technique to mitigate the impact of the shuffling countermeasure, the number of required measurements can further be reduced to approximately $47 \cdot 16 = 752$ measurements. That means the theoretical security gain of the protected implementation compared to the unprotected implementation is 16. This is the effect of shuffling.

This theoretical assumptions have been verified with practical measurements in order to compare the practical security gain with the theoretical security gain. Results of the practical experiments are depicted in Figure 2.20. For attack #1 no countermeasure mitigation steps have been performed. It is clearly visible that $nrdp < 0$ even for 100 000 measurements. This indicates that for at least 100 000 measurements the correct key hypothesis cannot be distinguished from the wrong hypotheses. For attack #2, the dummy rounds have been removed.

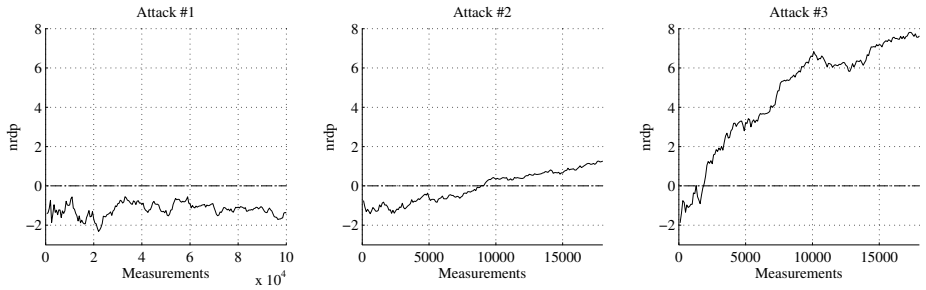


Figure 2.20: DPA attack results: No post-processing (left), dummy rounds removed (middle), dummy rounds removed and windowing (right).

Positive *nrdp* values for DPA attacks applying more than 9 000 measurements indicate that with that amount one can distinguish the correct key hypothesis from the wrong ones. In a last step windowing has been applied to further improve the results. Attack #3 depicts that approximately 2 000 measurements are sufficient to succeed with a DPA attack. These practical results correlate well with the approximated results from theory.

With the best practical DPA attack the security gain of the protected implementation is $\frac{2\,000}{47} = 43$, being approximately 2.7 times higher than the theoretical value of 16. For applying two countermeasures this security gain is not acceptable. A preceding template matching renders ineffective the dummy-round countermeasure. So the protected AES implementation can be compared to an implementation only applying the shuffling countermeasure. It further has to be considered that the (ineffective) dummy-round countermeasure increases the runtime of the algorithm by a factor of 2.5. The shuffling countermeasure does not impact the runtime. Therefore, two options can be proposed to improve the next version of *CRYPTA*. First, disable the dummy-round countermeasure and upper-bound the number of measurements under the same key to a value smaller than $47 \cdot 16$. A re-keying scheme can be applied therefore. Second, improve the dummy-round countermeasure by identifying and removing the source of leakage which allows to distinguish real rounds from dummy rounds.

Next to the hiding countermeasure, masking can also be applied to secure a block cipher against first-order DPA attacks. Although not supported by the *CRYPTA* chip, we want to discuss the security gain in case of a first-order secure masking scheme. A first-order DPA attack does not lead to a successful key recovery, independent of the number of measurements. In such a scenario a second-order DPA attack can be used for key recovery. That means a preprocessing step has to be performed on the measurements before the DPA attack can be performed. The preprocessing step consists of a combination of samples at two different points in time, e.g. the time where the mask is generated and the time where the masked intermediate value is processed. As combination function, the absolute difference [86] and the centralized product [111] have been shown to be good choices for devices leaking the Hamming weight of the processed values.

For the approximation of the security gain in case of first-order secure masking on the *CRYPTA* chip we followed a similar approach like presented in [85]. In a first step we profiled the leakage of the unprotected implementation. This was achieved by calculating mean and standard deviation of all samples belonging to the Hamming weights 0 to 8 at the time instance with the maximum exploitable leakage (i.e. the time instance where the maximum correlation appears). This values were used for simulating an implementation secured by means of first-order masking. We used the centralized product to combine the leakage of the two shares. A DPA attack on the combined samples showed that approximately $N_{min} = 1\,200$ measurements are required for distinguishing the correct key hypothesis from the wrong key hypotheses. In order to validate the correctness of our simulation, we repeated the same experiments with the mask values set to zero, what equals an unprotected implementation. A first-order DPA attack yielded $N_{min} = 49$, what is in line with the practical results of the unprotected implementation. So we can conclude that the security gain in case of first-order secure masking is approximately 25 what is comparable to the security gain in case of shuffling. For a second-order DPA attack the effort for identifying the time samples which have to be combined has to be considered additionally. This effort highly depends on the knowledge about the attacked implementation. Finally, it can be concluded that for a low-resource AES implementation as applied on the *CRYPTA* chip, first-order masking alone does lead to a satisfying security gain if the attacker is able to perform close-proximity measurements. As solution a combination of masking and shuffling can be applied.

2.6 Discussion

In this chapter we have presented SCA attacks applicable to passively-powered contactless devices. An RFID-tag prototype chip called *CRYPTA* served as target device. This chip includes an AES module with two countermeasures against SCA attacks, the random insertion of dummy rounds and shuffling. A flaw in the implementation allowed to render ineffective the dummy-round countermeasure. Together with windowing, this leads to an insufficient security gain when applying DPA attacks. This has been shown in theory and verified with practical experiments.

Furthermore, we introduce a novel measurement approach for contactless SCA scenarios, named resolution optimization. This approach does not require any analogue preprocessing circuit. It also shows advantages compared to the analogue demodulation circuit approach by Kasper *et al.* [65] when the distance between attacked device and measurement antenna exceeds 8 cm.

Based on this observation, we have applied the resolution optimization approach in order to verify the feasibility of a so-called remote SCA attack. This attack allows to successfully reveal the AES key of *CRYPTA* up to a distance of 1 m between device and measurement antenna. The ability of measuring side-channel information at large distances can be described by the parasitic load modulation, which is introduced and verified in this work. For the remote SCA

attack the countermeasures on *CRYPTA* have been disabled in order to minimize the measurement effort.

From the results presented in this chapter it can be concluded that low-cost SCA attacks in the RFID domain at close proximity are feasible even if the device is secured by countermeasures. The number of required measurements for a successful attack and as a result the measurement time increases with the countermeasures. In order to ensure a correct communication with the tag one has to stick to the timings specified by the ISO standard. These timings mainly define the measurement time, so investments to better measurement equipment will not lead to a much faster recording time. When applying remote SCA attacks, we conclude that higher investments in the measurement equipment can increase the attack performance. With our presented low-cost setup, approximately 30 k measurements are required to succeed at a distance of 100 cm. We are sure that the development of specialized measurement antennas, amplifiers, or the application of spectrum analyzers would lead to a more efficient attack at that distance. But, at the same time the equipment costs would increase. The integration of countermeasures would also render our low-cost, remote SCA attack inefficient. We come to this conclusion because of the high measurement effort for the scenario without countermeasures. In the remote SCA scenario with countermeasures an investment to more sophisticated equipment is a solution. It would allow to reduce the number of measurements what is beneficial due to the limited measurement speed because of the ISO timings.

3

SCA-Attacks Targeting Sensor Nodes

This chapter deals with side-channel measurement setups targeting off-the-shelf microcontrollers for battery powered devices like sensor nodes. Compared to devices for the RFID domain, chip size is not the main limiting factor for sensor nodes. They typically have more computing power and apply a microcontroller as central processing unit. Data transfer in sensor networks is typically done wirelessly what makes these networks vulnerable to eavesdropping. Therefore, confidential data is encrypted before it is transferred. Several microcontrollers nowadays already have hardware modules for cryptographic algorithms like AES integrated (e.g., TI MSP 430 [137], Atmel ATxmega 256A3 [8]) in order to speed up the encryption and decryption process. This often leads to the following design decision, which has to be made. Should the cryptographic algorithm be implemented in software or should the hardware module be applied. On the one hand, the hardware module outperforms the software approach in terms of performance. On the other hand, applying the software implementation allows more flexibility (e.g., adding additional countermeasures to harden the implementation against side-channel attacks). In this chapter we investigate both options. The results from this chapter have been published in [73] and [74] and can be summarized as follows.

Contribution

- We highlight the importance of the measurement position of the EM side-channel signal by exemplary SCA evaluations targeting an ATxmega 256 microcontroller.
- We show that front-side EM measurements contain more exploitable leakage compared to rear-side EM measurements

- We introduce a high-resolution EM measurement setup for mounting DPA attacks targeting cryptographic modules (AES, LRPRF) implemented on a front-side opened, prototype ASIC.
- The high-resolution EM measurements allow decreasing the complexity for finding the correct order of key bytes. We evaluate this in the context of the LRPRF implementation integrated on the prototype ASIC. Our results extend results published in related work, where comparable evaluations have been performed targeting an LRPRF implementation on an FPGA.

This chapter is structured as follows. Section 3.1 gives a brief introduction to wireless sensor networks (WSNs) and sensor nodes including a discussion of related work on attacks targeting WSNs. In Section 3.2, a setup for comparing the exploitable electromagnetic side-channel leakage dependent on the measurement location, is presented. In Section 3.2.5 the results of location-dependent SCA attacks targeting the AES hardware module as well as an AES software implementation on an ATxmega 256 microcontroller are presented. Section 3.3 includes a discussion of a measurement setup for high-resolution EM measurements. This setup allows to target specific sub-modules on the chip. Section 3.4 presents results achieved with the semi-invasive, high-resolution EM measurement setup. Section 3.5 concludes the chapter.

3.1 Introduction

Microcontrollers are widely used in all kinds of applications nowadays. One reason for the exhaustive usage is the great amount of functionalities they provide as well as their flexibility compared to dedicated hardware. One popular field of application is that of wireless sensor networks (WSN). WSNs consist of several sensor nodes which communicate with each other over a wireless channel. Each WSN typically has one base station which acts as the master in the network and forwards the received data from the sensor nodes to a backend system. The data transmitted by the sensor nodes varies depending on the field of application. WSNs are employed e.g. in health-care systems, for environmental monitoring, energy monitoring or building administration. In order to make attacks like eavesdropping or data alteration infeasible, the data is encrypted before transmission. Due to the data encryption, additional computational costs are introduced. These additional computational costs need to be minimized because sensor nodes are typically battery powered and the battery lifetime needs to be maximized. In order to achieve a long battery lifetime, efficient cryptographic primitives need to be used and implemented in an efficient way. Besides energy, code size and RAM size are also limited resources on sensor nodes.

One popular cryptographic primitive for data encryption is the standardized block cipher AES. Software implementations of AES exist for nearly every microcontroller platform. Some microcontrollers (e.g., TI MSP 430 [137], Atmel ATxmega 256A3 [8]) also have an integrated AES hardware module. The usage of an AES hardware module allows faster data encryption/decryption compared

to the software equivalent. Furthermore, parallel execution of other tasks during the encryption/decryption process is possible. Besides the efficiency of the data encryption/decryption, the leakage of secret information (e.g., the secret key) caused by side channels has to be analyzed. Software implementations allow to add countermeasures to minimize the exploitable side-channel leakage. If the hardware module leaks exploitable information via side channels, this issue can only be fixed by modifications on the protocol layer. One solution is to use session keys to limit the number of observable encryptions with the same key. Examples are fresh re-keying schemes as proposed in [89]. This modification makes DPA attacks more difficult.

Considering the measured side-channel targeting sensor nodes, the electromagnetic emanation (e.g., applied by Gandolfi *et al.* [44]) has advantages compared to the power consumption (e.g., applied by Kocher *et al.* [71]). First, for EM measurements, a modification of the device is not mandatory. Only high resolution EM measurements require a decapsulation of the chip in order to minimize the distance between measurement probe and chip die. For power measurements, in contrast, inserting a measurement resistor into the supply line or the ground line of the chip is mandatory. This typically requires an invasive modification of the PCB where the chip is mounted on. Second, EM measurements have an additional parameter, the measurement location. The location has a significant influence on the exploitable side-channel information in the measurements. Heyszl *et al.* [52] have analyzed strength and limitations of localized EM measurements on an FPGA. They apply front-side measurements and rear-side measurements in order to analyze the exploitable leakage. They come to the conclusion that EM measurements from the front side contain more exploitable leakage.

There have been several SCA attacks on implementations of symmetric block ciphers on various platforms in the past. These attacks show that it is of great importance to precisely analyze the exploitable side-channel leakage of an implementation. That is also the motivation for the work presented in this chapter. Some examples are given in the following. In [125] the authors present an AES key extraction targeting an FPGA implementation in less than 0.01 s. In this work the authors point out that the signal-to-noise ratio on the attacked device is 30 dB to 40 dB lower than an implementation on an ATxmega microcontroller. Kizhvatov *et al.* [70] have performed a power-analysis attack on the AES hardware module included in ATxmega microcontrollers showing that it is vulnerable against SCA attacks.

Block ciphers like AES often serve as building block for higher-layer cryptographic protocols like authenticated encryption (AE) [46] or for the creation of hash-based signatures [139]. Several works also deal with the implementation issues for cryptographic primitives, which arise because of the existing constraints for sensor nodes used in WSNs. In [30], the authors focus on the energy efficiency of security algorithms for WSN devices. They also compare software implementations with hardware accelerators from the point of view of energy efficiency. Rehman *et al.* [113] compare different encryption techniques for mes-

sage authentication codes (MACs) in WSNs. In [142] an AES-based security mechanism for WSNs, called *MoteSec-Aware*, is presented.

3.2 Non-Invasive Location-dependent EM measurements

This section describes the process of non-invasive location-dependent EM measurements targeting an off-the-shelf microcontroller (Atmel ATxmega 256A3). The exploitable side-channel leakage during AES encryptions at different points on the chip from the front side and the rear side have been investigated. The leakage of one software implementation and the leakage of the AES hardware module have been analyzed.

Results show that a successful key recovery highly depends on the measurement location of the EM side-channel signal. For the software implementation, if the correct location is selected, less than 100 measurements are sufficient for revealing the correct key values. For the hardware module, we have investigated front-side and rear-side measurements. Here we can conclude that SCA attacks using front-side measurements require a smaller amount of measurements for key recovery (i.e. are more efficient) compared to rear-side measurements.

3.2.1 Used Microcontroller

In this section, we introduce the microcontroller which was used for the evaluations. We decided to use an AVR XMEGA microcontroller, the ATxmega 256A3 to be exact. Due to the low power consumption, the high integration as well as the real-time performance this microcontroller is frequently used on wireless sensor nodes. Furthermore it has an AES hardware module implemented.

In the following section we provide some more detailed facts about the ATxmega 256A3 microcontroller. The ATxmega 256A3 has 256 kB in-system self-programmable flash memory, 8 kB of boot-code section, 4 kB EEPROM, and 16 kB SRAM. The CPU is based on the AVR enhanced RISC architecture equipped with 32 general purpose working registers. The microcontroller can be operated with voltages between 1.6 V and 3.6 V and the maximum clock frequency is 32 MHz. Additional features are: One 16 bit real-time counter, 16 bit timer/counter for PWM and compare modes, serial interface, ADCs, one DAC, 50 general-purpose I/O lines, and several other microprocessor-specific features. In addition to the AES hardware module a DES hardware module is also included. The DES hardware module is out of the scope of this work, so no detailed description about this feature is provided. For more detailed information about the ATxmega 256A3 we refer to the datasheet [8]. Typical applications for this microcontroller are industrial control, factory automation, metering, and medical applications, to mention only a few. Although it is not explicitly noted as a high-security device we are sure that the cryptographic features the microcontroller provides are used frequently. So it makes sense to analyze possible weaknesses of these features.

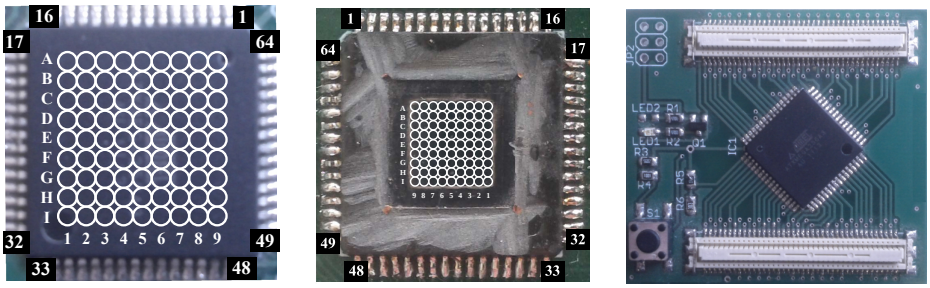


Figure 3.1: The attacked microcontroller.

3.2.2 AES Implementations

In the following paragraphs, an introduction to the software AES implementation for the ATxmega microcontroller is provided. Afterwards, the AES hardware module is introduced.

The software AES implementation is written in assembler. The implementation is optimized for fast encryption/decryption and no countermeasures against SCA attacks or fault attacks are implemented. The round key is calculated after each round and the byte substitution is implemented as a table lookup. It takes 4 054 clock cycles to encrypt one block of plain text. This is close to the 3 766 clock cycles given in [114].

The AES hardware module requires 375 clock cycles to encrypt one block of plain text and it supports a key length of 128 bits. Comparing the run time of the AES hardware module with the figures given for the software implementation it can be said that the AES hardware module is approximately ten times faster. The usage of the AES hardware module allows to perform other tasks in parallel to the encryption process. Detailed information about the hardware module can be found in [7].

3.2.3 Measurement Setup

In order to perform the measurements and automate them to a high degree a simple program has been developed for the microcontroller. After setting the AES key with an initial command, the plain text followed by a single control byte is sent to the microcontroller using the serial interface. This control byte is used to select which implementation should be used for the encryption, the AES software implementation or the AES hardware module. A trigger pin set to high indicates that the encryption is currently executed. In a last step the result is sent back to the control computer. This setup clearly indicates that the used key of the attacked device is known. Knowledge of the key simplifies the creation of two-dimensional EM-leakage landscapes.

A probe manufactured by '*Langer EMV Technik*' (model: '*ICR HH 100-27*') has been used to measure the EM emanation of the chip. The signal of the probe was amplified with a 30 dB amplifier. The amplified signal was digitized using a

'LeCroy WP 725 Zi' oscilloscope. The sampling rate on the oscilloscope was set to 1 GS/s and the microcontroller was clocked with a frequency of 13.56 MHz.

The left picture in Figure 3.1 shows the grid on the chip, where the probe has been placed in order to measure the EM signal, from the front side. The grid has a size of 9 x 9 points, leading to a total of 81 points. The distance between the points is 1 mm. The picture in the middle of Figure 3.1 depicts the grid on the chip for the measurements from the rear side. Here, we have removed the package material in order to measure directly on the chip die. The chip die has a size of 5 mm x 5 mm. For the rear-side measurements, the focus was put on the die area, so a step size of 0.55 mm was used leading to a grid of 9 x 9 points again. This approach is irreversible and has been performed for the sake of completeness to verify the amount of exploitable leakage of the investigated microcontroller for front-side measurements and rear-side measurements as done in [52]. For front-side measurements and rear-side measurements, the probe has been moved using a stepper table. This allowed us to automate the measurement process up to a high degree. The amplitude of the measured EM signal varied for different points. Because of this observation, a calibration step in each point was performed before the traces were recorded. This calibration step ensures the same resolution of the voltage values in each point.

The right picture in Figure 3.1 depicts the board where the microcontroller is mounted on. Using this board it is hardly possible to perform power measurements without irreversible modifications. This is true for most real-world devices. Therefore, EM measurements are the best choice for measuring side-channel information. However, in order to perform the semi-invasive rear-side measurements, we had to develop a custom board which allows to mount the chip inverted.

3.2.4 Evaluation Metric

For verification of the location-dependent EM leakage N EM measurements were recorded in every point P ($P = 1 \ldots 81$). The measurement interval covered the first AES round and the same set of N random plain texts was used in each point. In total $81 \cdot N$ measurements were recorded. A DPA attack with the correlation coefficient as distinguisher was performed for each point separately returning correlation values for every key guess. For determining whether the correct key guess can be distinguished from the wrong key guesses, the *nearest rival distinguishing power (nrdp)* as already introduced in Section 2.3 is used. This evaluation results in 81 *nrdp* values, one for each point. This vector is then transformed into a matrix of size 9×9 according to Equation 3.1. The 1^{st} entry in the 1^{st} row corresponds to point *A1*, and the 9^{th} entry in the last row corresponds to point *I9*.

$$M = \begin{bmatrix} nrdp_1 & nrdp_2 & \cdots & nrdp_9 \\ nrdp_{10} & nrdp_{11} & \cdots & nrdp_{18} \\ \vdots & \vdots & \ddots & \vdots \\ nrdp_{73} & nrdp_{74} & \cdots & nrdp_{81} \end{bmatrix} \tag{3.1}$$

For comparison, the previous evaluations have also been performed based on the maximum correlation value for the correct key hypothesis instead of the *nrdp* value. Here, values for the wrong key hypotheses are not taken into account and therefore this method is independent of the attacked algorithm.

For the AES software implementation $N = 2\,000$ traces were recorded in each point, which leads to a total of $2\,000 \cdot 81 = 162\,000$ traces. The choice of recording 2 000 traces in each point has several reasons. First, from attacks targeting software AES implementations on 8 bit microcontrollers based on power measurements we knew that the attack succeeds with less than 1 000 measurements. So this amount of traces should also lead to exploitable EM leakage in a vast majority of the points. Second, we approximated the overall measurement time. For 2 000 measurements in each point we approximated an overall recording time of 5 hours. This duration was acceptable for a first measurement run and there was still space for increasing N if the results were not satisfactory. The results of the evaluations showed that the number was sufficient for the location-dependent EM leakage verification. For the correlation attack the Hamming weight of the output of the first substitution box, $HW(Sbox(p_i \oplus k_{g,i}))$, served as power model ($p_i \ldots$ plain text byte i, $k_{g,i} \ldots$ guess g for key byte i, $i = 0 \ldots 15$, $g = 0 \ldots 255$). The suitability of the Hamming weight power model has been verified by preliminary experiments. The reason for the suitability of this power model can be described by the fact that the microcontroller sets the data bus to zero before writing a new value.

For the AES hardware module $N = 10\,000$ traces were recorded in each point, which leads to a total of $10\,000 \cdot 81 = 810\,000$ traces. Compared to the software AES implementation, we expected less exploitable leakage from the hardware AES module. Similar to the software implementation, we approximated the overall measurement duration. 10 000 measurements per point lead to an approximated measurement time of 5 hours, what is similar to the software AES. At the first glance, one could expect ten times more measurements at the same duration. This expectation is based on the fact that the software AES implementation required ten times more clock cycles per encryption compared to the AES hardware module. But the overhead for reading and writing the state register, and the configuration of the hardware module have to be considered. Additionally, the communication overhead is equal for both cases. This leads to the speedup factor of 5. As power model for the correlation attacks we applied the same model used in [70]. This model takes into account two subsequent bytes of the plain text (p_i, p_{i+1}) and two bytes of the key $(k_{g,i}, k_{g,i+1})$: $HW((p_i \oplus k_{g,i}) \oplus (p_{i+1} \oplus k_{g,i+1}))$.

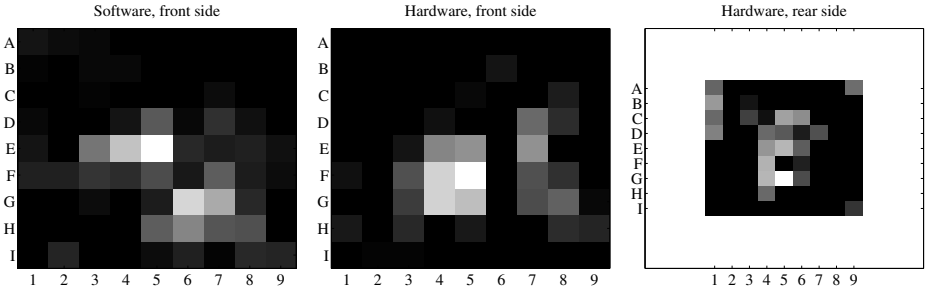


Figure 3.2: EM-leakage landscapes for the software implementation and the hardware AES module.

3.2.5 Location-Dependent SCA-Attack Results

Figure 3.2 depicts from left to right the leakage landscapes created for the SW implementation using front-side measurements, the hardware AES module using front-side measurements, and the hardware AES module using rear-side measurements. Bright points indicate a positive *nrdp* value at the respective position (correct key guess can be distinguished from the wrong key guesses). All positions with $nrdp < 0$ are plotted black. 51 locations lead to $nrdp > 0$ for the software implementation when 2 000 measurements are used. 32 (26) locations lead to $nrdp > 0$ for the hardware AES module for front-side (rear-side) measurements when 10 000 measurements are used. The evaluations based on the maximum correlation value of the correct key hypothesis lead to similar results. This is because wrong key hypotheses do not lead to a significant correlation value at the output of the first AES substitution box.

Figure 3.3 underlines the importance of the measurement location for a subsequent DPA attack if the number of measurements is limited. In all three scenarios, for a small amount of measurements only some specific measurement locations lead to $nrdp > 0$ what indicates that the correct key value can be distinguished from wrong key guesses. The size of the leakage landscapes reflects the size of the covered area by the EM measurements. The leakage landscapes corresponding to the rear-side evaluations are scaled to a smaller size because the observed area for the rear-side evaluations (third row in Figure 3.3 and right plot in Figure 3.2, respectively) was smaller (5 mm x 5 mm) compared to the front-side evaluations (9 mm x 9 mm).

From the previous figures, it is clearly visible that more points offer exploitable EM leakage for the software implementation compared to the hardware AES module. This can be described by the fact that the software implementation heavily uses the architecture of the microcontroller while the hardware AES module is a standalone module not using the architecture of the microcontroller during encryption. When the front-side results and the rear-side results for the hardware AES module are compared, it turns out that more measurement points for the front-side attacks lead to $nrdp > 0$. This observation can

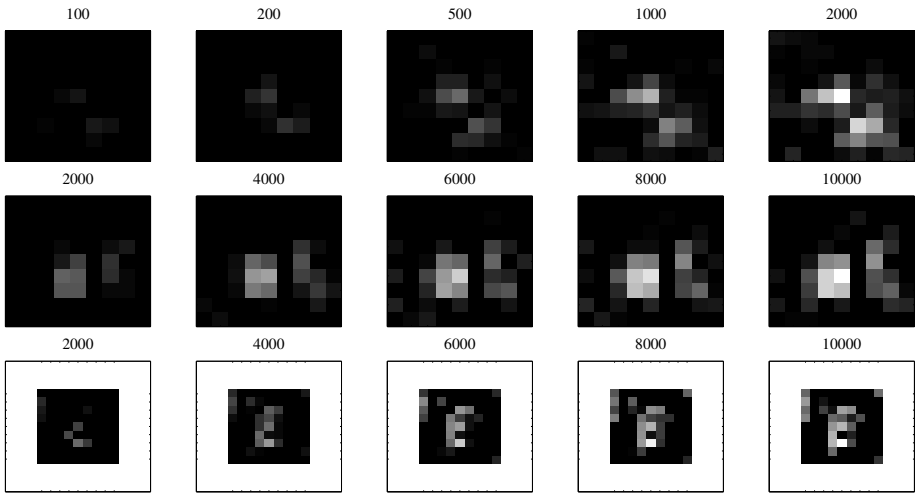


Figure 3.3: EM leakage evolution (Top: Software AES; Middle: Hardware AES, front side; Bottom: Hardware AES, rear side).

be described by the fact that the covered area for the front-side was larger and therefore leakage from the bonding wires increases the count. Furthermore, with a similar amount of traces, higher $nrdp$ values can be achieved by using front-side measurements: For the DPA attacks applying 10 000 measurements, $max(nrdp) = 2.63$ for the front side and $max(nrdp) = 1.61$ for the rear side respectively. This is an indicator, that the signal-to-noise ratio for front-side measurements is higher compared to rear-side measurements. Based on this result we can conclude that the maximum EM leakage is produced by the upper metal layers and as a result even non-invasive front-side measurements contain more exploitable EM leakage compared to semi-invasive rear-side measurements. This observation is in-line with the results presented in [52]. Table 3.1 provides a summary of the aforementioned results.

3.3 Semi-Invasive High-Resolution EM Measurements

In this section SCA-attack results targeting a prototype ASIC chip named *TAMPRES* ASIC designed for sensor nodes are presented. Compared to the previous section, the *TAMPRES* chip is placed in a prototype package which is open at the front side. This enables high-resolution EM measurements. We first introduce the *TAMPRES* ASIC, next we discuss the measurement setup followed by a discussion of the results which were achieved with the high-resolution EM measurements.

We first investigated the AES hardware module, which does not have side-channel countermeasures integrated. The SCA results show that 1 200 measure-

ments are sufficient for key recovery in a non-invasive measurement scenario. When applying the high-resolution measurement setup, the number of required measurements can be reduced to 210. This number has to be considered as upper bound when applying re-keying. Less than 210 encryptions are allowed under the same key, otherwise the construction is not secure against that type of attacks. Next, we investigate the LRPRF hardware module, which allows to generate session keys for the AES in a secure manner. The investigations show, that an attacker is faced with two challenges. First, finding the 24 key bytes is not trivial, with the best localized EM SCA attack we are only able to reveal 13 correct key byte values. Second, even after a profiling step, it is not possible to map all key byte values to their correct position, a complexity of approximately 2^{32} remains.

3.3.1 *TAMPRES* ASIC Introduction

An MSP430 microprocessor core is the central processing unit of the *TAMPRES* ASIC. The MSP430 microcontroller family from Texas Instruments (TI) [136] with its ultra-low power features together with a rich set of integrated peripherals is well suited for embedded applications. MSP430 microcontrollers are frequently used in battery-powered sensor nodes applied in wireless sensor networks (WSNs). This makes it well-suited for integrating it into the *TAMPRES* ASIC. TI does not provide the HDL code for the MSP430 but only complete devices with different hardware configurations. Due to the fact that several custom hardware modules had to be additionally integrated on the *TAMPRES* ASIC, the HDL description of the MSP430, called OpenMSP430, available from opencores.org [103] was used. This approach allows to integrate additional custom hardware modules, which include mainly security-relevant features, side-by-side to the standard hardware modules. One further advantage is that the standard MSP430 toolchain for programming the ASIC can be used. The architecture of the *TAMPRES* ASIC is shown in Figure 3.4, custom hardware modules are highlighted in orange and standard hardware modules are highlighted in green. The

Table 3.1: Results of the DPA attacks for the three scenarios when different numbers of measurements are used.

	Software front side		Hardware AES module front side		Hardware AES module rear side	
Measurements	Nr. Pts	$max(nrdp)$	Nr. Pts	$max(nrdp)$	Nr. Pts	$max(nrdp)$
100	8	1.74				
200	10	3.33				
500	30	6.53				
1 000	40	11.15	9	0.58	10	0.19
2 000	51	16.36	15	1.00	11	0.67
4 000			22	1.64	20	0.97
6 000			27	2.10	24	1.26
8 000			30	2.30	25	1.51
10 000			32	2.63	26	1.61

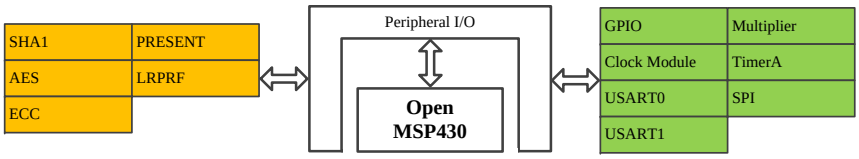


Figure 3.4: *TAMPRES* ASIC architecture (Orange: custom hardware modules, Green: standard hardware modules).

goal of integrating custom hardware modules on the *TAMPRES* ASIC was to accelerate cryptographic operations, therefore the following modules have been implemented:

- **SHA-1** The "Secure Hash Algorithm 1" is included in order to verify the integrity of received data like firmware updates.
- **AES** An AES hardware module is integrated in order to allow efficient data encryption and decryption to enable a secure data transfer. No countermeasures against implementation attacks are included as a session key derived using the LRPRF core is used which limits the number of encryptions under the same key.
- **ECC** The "Elliptic Curve Cryptography" hardware module supports elliptic curve point multiplications. This operation represents the most resource-consuming operation in elliptic curve cryptography. The remaining operations required to perform a protocol based on ECC have to be implemented in software.
- **PRESENT** A threshold implementation of the lightweight block cipher PRESENT [24] is included as dedicated hardware module. Compared to AES the structure of PRESENT allows to implement a masking countermeasure at reasonable cost. Therefore PRESENT does not require a frequent re-keying which can be advantageous for specific applications.
- **LRPRF** The "Leakage-Resilient Pseudo-Random Function" is implemented according to the work by Medwed *et al.* [90] with Rijandael-192 as underlying block cipher. This hardware module allows to efficiently generate session keys for the AES hardware module.

3.3.2 Measurement Setup

Two different EM probes for measuring the EM emanation of the *TAMPRES* ASIC have been used. The first probe, model '*LF-B 3*' from '*Langer EMV Technik*', features a resolution of approximately 2 mm. This resolution is not sufficient for semi-invasive high-resolution measurements, but we used this probe

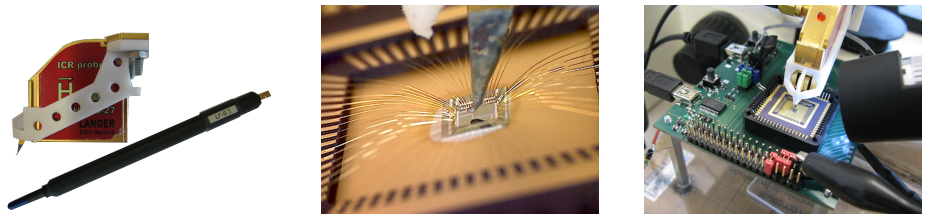


Figure 3.5: Left: EM probes. **Center:** '*ICR HH 100-27*' EM probe tip close to the chip die. **Right:** EM emanation measurement setup.

for some preliminary measurements during our evaluation. In particular, we followed a similar approach like presented in [130]. We measure the EM emanation of one decoupling capacitor as alternative to power measurements. The design of the PCB where the *TAMPRES* chip is mounted on does not allow to add a resistor in the power line to perform power measurements. For the high-resolution EM measurements, an '*ICR HH 100-27*' EM probe from '*Langer EMV Technik*' has been applied. This probe allows to achieve resolutions down to 30 μm. Both EM probes are depicted in the left picture of Figure 3.5. To achieve an accurate and automated positioning of the EM probes, they are mounted on a stepper table. The stepper table can be moved in x, y, and z direction with a resolution of 0.05 μm. This setup allows to automatically scan the whole chip surface and find points with high leakage. A USB microscope has been applied to assist in accurately positioning the EM probes. Especially when setting the distance between EM-probe tip and chip surface, the usage of the microscope is inevitable. A picture taken with this USB microscope, showing the '*ICR HH 100-27*' EM probe tip close to the chip die, is depicted in the center of Figure 3.5. The rightmost picture in Figure 3.5 provides an overview of the whole EM-measurement setup for the semi-invasive scenario. It includes the '*ICR HH 100-27*' EM probe mounted on the stepper table, the USB microscope, and the opened *TAMPRES* ASIC. As it can be seen in the pictures, the chip is mounted in a prototype package, this package does not allow semi-invasive, rear-side EM measurements.

3.4 Semi-invasive SCA-Attack Results

In the following we present the SCA-attack results achieved targeting the *TAMPRES* ASIC. The focus of the evaluations is put on the AES hardware module and the LRPRF hardware module.

3.4.1 AES Hardware Module

Preliminary results based on the EM measurements from the decoupling capacitors revealed that approximately 1 200 measurements are sufficient for $nrdp > 0$. We further evaluated the first-order success rate. 1 800 measurements are sufficient for a first-order success rate of 100%. The AES hardware module processes

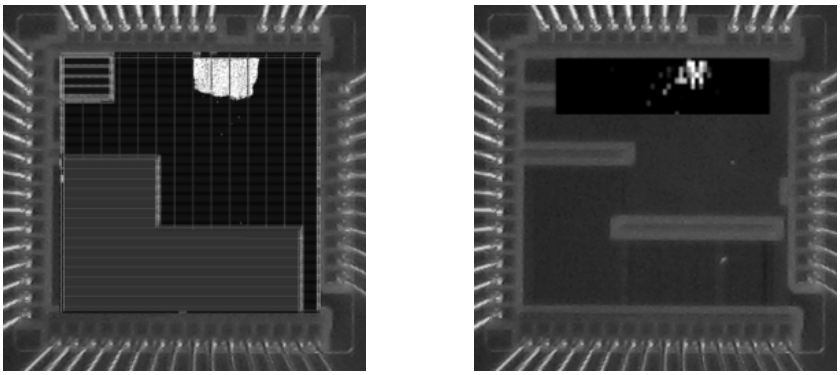


Figure 3.6: AES hardware module location based on implementation information (left) and by applying high-resolution EM measurements (right).

one column of the state in a single clock cycle. That means four byte substitutions and the mix-columns step are performed during one clock cycle. The register values serving as input are updated by the output of the mix-columns step. The choice of the Hamming distance power model is based on this implementation details. After the first key byte of the column is revealed, this information can be used to attack the following key bytes in the column. This additional information decreases the number of measurements, as can be seen in Table 3.2. If three key-byte values of one column are known, 210 measurements are sufficient for $nrdp > 0$, and 470 measurements for a success rate of 100%, when targeting the last key byte of the column.

Table 3.2: Number of measurements to reach a success rate of 100% and $nrdp > 0$ depending on the number of already revealed key bytes in the appropriate column.

	$nrdp > 0$	Success rate = 100%
0 key bytes known	1 300	1 800
1 key byte known	700	1 000
2 key bytes known	400	600
3 key bytes known	210	470

First results with the high-resolution EM probe show that the position of the AES core can be detected precisely. This is shown in Figure 3.6. In the left picture the location of the AES hardware module on the chip based on layout information is shown. The designers of the *TAMPRES* chip provided us this information. The right picture shows the results of our practical investigations. 5000 EM measurements were recorded in 594 points and a Hamming weight power model of the output of the first byte substitution was applied. Bright points indicate locations with exploitable leakage. When comparing both pictures, it is clearly visible that the locations with exploitable leakage correlate

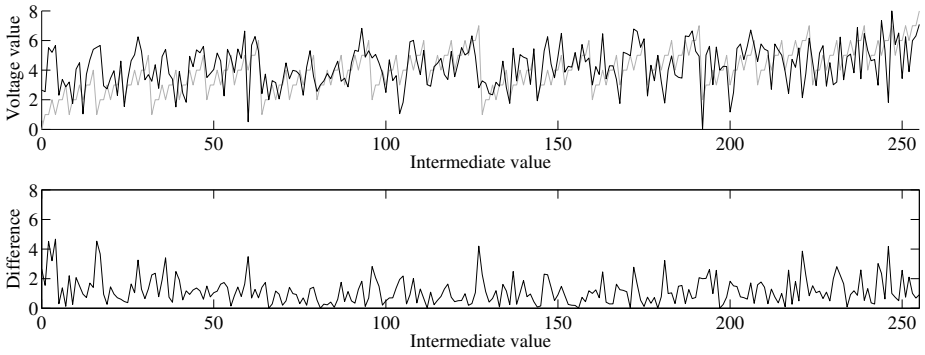


Figure 3.7: Result of the profiling step for finding an appropriate power model for the high-resolution EM measurements.

well with the actual location of the AES hardware module. Because of the measurement effort we only covered a small area of the chip but this does not change the relevance of the results.

One interesting observation is that the Hamming weight power model targeting single byte-substitution results works well for the high-resolution measurements while this power model applied to the previous measurements does not lead to exploitable results. In fact, the Hamming weight power model is very unusual for a hardware implementation. For verifying the applicability of this model a profiling step has been performed. Figure 3.7 depicts the result of this profiling. Voltage values corresponding to the intermediate values processed at a specific sample point are represented by the black graph. To fit the possible Hamming weights for an 8 bit intermediate value the voltage values have been normalized to values between 0 and 8. The gray graph in the upper plot corresponds to the actual Hamming weights for the intermediate values. In the lower plot the absolute difference for every intermediate value between the black graph and the gray graph from the upper plot is shown. By applying the Hamming weight power model in a DPA attack, the correct key hypothesis can be clearly distinguished from the wrong key hypotheses ($\rho_{correct} \approx 0.13; max(\rho_{wrong}) \approx 0.06$). This is one reason why we did not further refine the power model by e.g. adding weights to the single bits of the intermediate value. The second reason is that we wanted to keep the power model as general as possible.

Based on the previous results, we further decreased the observed area to investigate specific leakage characteristics of the four SBoxes working in parallel. The outcome of this investigation shows that each SBox has specific locations where exploitable leakage can be measured. Figure 3.8 depicts the results of these investigations. A correlation attack with the Hamming-weight power model has been performed for each point and the results are plotted according to the bytes arranged in the AES state. It can be observed that the leakage landscapes of the rows are very similar while the leakage landscapes in each column show significant differences. By evaluating the best location for every key byte in

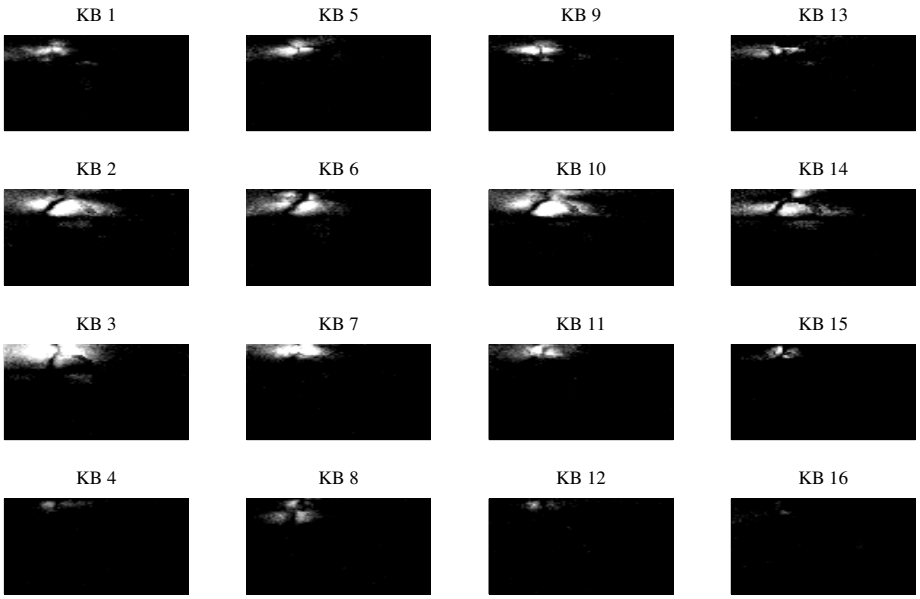


Figure 3.8: AES SBox characterization.

detail we extracted the minimum number of measurements for $nrdp > 0$ and for a success rate of 100%, respectively. Between 90 and 900 measurements are required for $nrdp > 0$ and between 140 and 2 500 measurements are required for a success rate of 100%, depending on the key byte.

Summing up the previous results we can conclude that by applying high-resolution EM measurements it is possible to create a mapping between key byte value and position in the AES state. So if an attack reveals the 16 key byte values but not their position, $16! \approx 2^{44.3}$ key candidates remain. Testing this amount of key bytes with a brute-force attack is still practical with today's computing power. But for AES-192 and AES-256 the number of eligible key candidates increases to $24! \approx 2^{79.0}$ and $32! \approx 2^{117.7}$, respectively. These numbers for key candidates exceed the practical number for a brute-force attack. But if the location information of the leakages is incorporated, the complexity for finding the correct order can be decreased. These findings have been used for the investigations targeting the LRPRF hardware module discussed in the next section.

3.4.2 LRPRF Hardware Module

The implementation of the "leakage-resilient pseudo-random function" (LRPRF) on the *TAMPRES* ASIC is motivated by the work of Medwed *et al.* [90]. It uses Rijandael-192 as underlying block cipher, that means the key length is 24 bytes. The side-channel security of the implementation is based on the following princi-

ple. First, the number of different plaintexts at the input is limited to $N_p = 256$. This is achieved by using the same value for all 24 bytes of the plaintext, i.e. $p_j[i] = j$ for $0 \leq i \leq 23$ and $0 \leq j \leq 255$. Additionally, the 24 substitution boxes are evaluated in parallel ($N_s = 24$). This ensures that an attacker can only measure leakages of the form

$$l_j = \sum_{i=1}^{N_s} L(S(p_j[i] \oplus k[i])) + n \tag{3.2}$$

with S representing the AES-SBox, L being the leakage function, and n a Gaussian-distributed noise. But, this assumption given in [90], does not necessarily be true if an attacker can perform high-resolution EM measurements. In such scenarios the measurement location has to be considered additionally. As the following experiments show, there exist measurement points m_I where the leakage of one AES-SBox I is much larger compared to the remaining ones. This can be modeled by a weighted sum of the leakages. The overall leakage for such a measurement point m_I can be expressed as

$$l_{j,m_I} = \sum_{i=1}^{N_s} w_{i,m_I} \cdot L(S(p_j[i] \oplus k[i])) + n \tag{3.3}$$

with w_{i,m_I} being the weight of the leakage produced by Sbox i, $w_{I,m_I} = 1$ and all the remaining weights being smaller 1.

The authors of [15] have also investigated the security of an LRPRF implementation. In contrast to our work, they have implemented the LRPRF on an FPGA and they use four-bit SBoxes. For their evaluations, they distinguish between two operation modes, *fixed mode* and *open mode*. The fixed mode poses the standard operation mode of the LRPRF, where all input words have the same value. The open mode allows to assign every input word an individual value. This mode is used for evaluation and should not be supported by a real-world product. Running the device in the open mode allows some worst-case profiling, giving the attacker significant advantage.

When attacking the LRPRF hardware module, the first step consists in finding the 24 correct key bytes. In order to do so we collected 25 600 EM measurements at 624 measurement positions. As our input space is limited to $N_p = 256$ by construction, that means we collected 100 measurements for each plaintext. Next, a DPA attack is performed for each measurement point separately. We have used the correlation coefficient as distinguisher and the Hamming weight power model. The choice of the power model was motivated by the results achieved targeting the AES implementation from the previous section. As quality measure we used the number of correct key values $|k_c|$ within the $N_s = 24$ best-ranked key values returned by the 624 DPA attacks. At measurement point 483, we found $|k_c| = 13$ correct key values within the $N_s = 24$ best-ranked key values. The ranks for each key byte are given in the following vector:

[**12 7** 47 173 **11** 25 **16 1 24** 48 215 **15 22 21 2** 135 73 55 32 210 **16 3** 242 **6**]

This was the best result, at all other measurement points we got $2 \leq |k_c| \leq 12$. This result clearly shows that in the scenario where the number N_p is fixed to 256, what equals the standard operation of the LRPRF module, this approach does not allow to reveal all 24 correct key-byte values. This operation mode is referred to as *fixed mode* in [15]. If an attacker does not have additional profiling capabilities, the lower bound for the attack complexity of 24! holds. The results show that in practice the attack is even harder, as the attacker additionally has to guess 11 key-byte values.

The LRPRF hardware module also allows a second mode, where the $N_s = 24$ plaintext bytes can have different, random values. This mode is equivalent to the *open mode* in [15]. This mode allows to target each of the 24 SBoxes separately and is only included in the prototype chip for evaluation purposes. We applied the same approach as for the fixed mode but, as this setting allows, we only varied the plaintext byte of the currently evaluated SBox. The remaining input bytes were set to zero. With this setting it was possible to reveal all 24 key byte values and also the correct order of the bytes. We also created leakage landscapes for all 24 SBoxes, which are shown in Figure 3.9. For creating these leakage landscapes, we plot the correlation coefficient of the correct key hypothesis at every measurement position. The figure shows that every SBox has a unique leakage characteristic. Our result can be interpreted as an extension to the results presented in [15]. The authors of [15] observe different leakage characteristics of SBox instances on an FPGA while our results are based on EM measurements from a taped-out ASIC. The similarity of both results allow the conclusion that the location information can be beneficial when attacking FPGA and ASIC implementations of LRPRFs.

Next, we investigate the possibility to combine the results from the open mode with the results from the fixed mode. In particular we try to use the leakage landscapes to map the revealed key bytes from the fixed mode to their correct position. We used the correlation coefficient for similarity measure. For every key byte the algorithm returned a sorted list of positions. The order is based on the value of the correlation coefficient. The following vector gives the index of the correct position for each of the 24 key bytes.

[**1** 2 **1** **1** **1** 3 4 **1** **1** 4 2 **1** 2 4 **1** 4 8 6 **1** 4 3 6 **1** **1**]

11 key bytes can be mapped to their correct position. 13 positions remain which need to be guessed. That equals a complexity of 13! $\approx 2^{32.5}$, what is already practical for a brute-force attack. But it has to be considered that this result is based on the fact, that the device features the open mode. This mode allows a profiling of each individual SBox and is typically not available on real-world devices.

3.5 Discussion

In this chapter, low-cost measurement setups for capturing the EM side channel of microcontrollers have been presented. All equipment which has been used is off-the-shelf equipment, accessible by everybody.

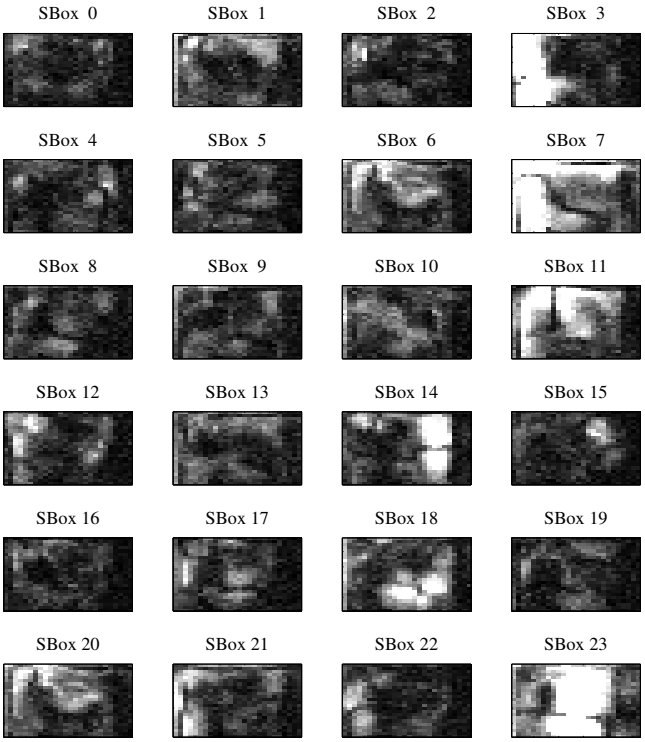


Figure 3.9: LRPRF Sbox leakage landscapes.

In the first part of this chapter, non-invasive and semi-invasive EM attacks targeting an ATxmega 256 microcontroller are discussed. The non-invasive attacks have the advantage that they do not require a modification of the device. Results show that both, hardware and software implementations on this device are vulnerable to SCA attacks. Here, only the software implementation allows to add countermeasures to increase the security level. Furthermore we show that a good choice of the measurement location allows to significantly decrease the number of required measurements for successful key recovery. When comparing front-side and rear-side measurements we conclude that the exploitable side-channel leakage for front-side measurements is higher, what is in-line with the results presented in [52]. Both AES implementations are not protected by SCA countermeasures allowing a efficient key recovery with a small number of measurements. For the hardware AES module it is not possible to add countermeasures as it was not intended by the chip manufacturer. Recent results targeting the hardware AES module of an FRAM-based, low-power MCU (TI MSP430 FR5969) have shown that this new technology significantly increases the effort for key recovery [93]. For the software AES implementation, countermeasures like hiding or masking can be added to secure the device against non-profiled, first-order DPA attacks. Nevertheless, several publications in the past

have shown that profiled attacks or higher-order DPA attacks are effective tools for attacking protected devices. In such scenarios, the number of required measurements for key recovery increases but the requirements for the measurement setup do not change. Based on the results targeting the unprotected implementations, we conclude that also protected implementations can be successfully attacked with the low-cost equipment.

In the second part of this chapter, semi-invasive, high-resolution EM measurements are performed. A prototype ASIC chip specialized for sensor-node applications serves as evaluation target. First investigation target the unprotected AES implementation. Results show that in the case of high-resolution EM measurements a simpler power model can be applied and the number of measurements for successful key recovery can be decreased. Furthermore we can show that different SBox instances have a different, spatial leakage characteristic. This finding has further been used to attack the LRPRF hardware module implemented on the ASIC. Investigations targeting the LRPRF hardware module reveal that a profiling step can assist in finding the correct ordering of the key bytes. During the profiling step, landscapes are created for every SBox instance (24 SBoxes in total). Here, a comparison of our results with the results presented in [15] shows that spatial leakage can be observed on FPGAs and on ASICs. During the attack step the leakage landscapes assist in finding the correct byte position for key-bytes. The evaluations of the LRPRF show that even if the attacker is able to perform an advanced profiling using the open mode, it is not possible to reveal the correct values of all key bytes nor successfully map the key bytes to the correct byte position. We conclude that here the low-cost approach limits the success. Multi-channel EM measurements are expected to increase the attack performance as e.g. shown in [131]. But due to the additional need for EM probes and positioning devices, also the equipment costs are beyond our definition for low cost.

4

SCA Attacks Targeting a Crypto ASIC

In the following chapter, we present results of SCA evaluations targeting an authenticated encryption (AE) algorithm implemented on an ASIC. The ASIC was designed with the goal to provide a power-analysis attack evaluation platform. Therefore the implemented countermeasures, hiding and masking, can be switched on and off. This allows to investigate the impact of the countermeasures. Furthermore, the ASIC targets low-resource applications leading to the additional design goals of minimum chip area and low power consumption.

DPA attacks have been performed targeting the implementation with activated hiding countermeasure and activated masking countermeasure. The results show that both approaches alone do not lead to a satisfying security level, especially when the implementation is designed for low-resource applications. Only the combination of both countermeasures leads to a satisfying security level. Results presented in this chapter have been published in [97] and the contributions can be summarized as follows.

Contribution

- First practical DPA attacks targeting an ASIC implementation of Keccak running in a keyed mode.
- Discussion of the security of three different secret-sharing approaches for low-resource devices. The security evaluations are based on non-profiled, higher-order DPA attacks.
- Investigation of the hiding countermeasure for keyed Keccak instances for low-resource devices. DPA attacks on windowed power traces have been performed for these investigations.

- Discussion of the security gain in case of the combination of the hiding and masking countermeasure compared to their standalone application, especially for low-resource devices.
- Comparison of runtime and implementation overhead for different security levels achieved by the hiding and masking countermeasure.

This chapter is structured as follows. In Section 4.1 we give a brief introduction to authenticated encryption and how KECCAK can be used in this context. Preliminary information is provided in Section 4.2 and implementation details of the ASIC are given in Section 4.3. Next, the SCA-attack setting is introduced in Section 4.4 followed by the results in Section 4.5. In Section 4.6 we conclude the chapter with a short discussion.

4.1 Introduction

Confidentiality and authenticity of data are among the most important cryptographic services required to transfer data securely over public communication channels. The former is commonly achieved by symmetric encryption algorithms while the latter is often obtained by message authentication codes (MACs). These cryptographic primitives have been treated independently in the past, which led to inefficient solutions and severe security problems [28, 31]. For this reason, researchers have started to develop new hybrid algorithms that offer the desired service of authenticated encryption (AE), for instance, as part of the on-going CAESAR competition [1].

In [17], an AE algorithm based on a sponge function is proposed, called SPONGEWRAP. Any sponge function can be used to implement this algorithm, one option is KECCAK [21], the winner of the NIST SHA-3 competition [99]. KECCAK has been analyzed for several years during the SHA-3 competition by researchers all over the world. This means that it provides state-of-the-art security from a mathematical point of view. However, when KECCAK is used in a keyed mode, like the SPONGEWRAP mode, the vulnerability to implementation attacks such as differential power analysis (DPA) attacks [71] must be considered. Especially smart cards and mobile devices are exposed to implementation attacks. Here, the adversary can measure physical leakage with low effort.

For devices which are vulnerable to implementation attacks, the integration of countermeasures like hiding or masking techniques is mandatory. The authors of KECCAK proposed to implement a secret sharing technique to protect keyed KECCAK instances [16, 19]. This technique is based on the idea to divide key-dependent intermediate values into unique parts (so-called shares) and to re-combine them after the processing. In order to achieve first-order DPA resistance, this sharing needs to fulfill three properties: correctness, non-completeness, and uniformity [100]. Interestingly, Bilgin et al. [22] reported that the implementation in [16, 19] does not fulfill the uniformity property and is therefore not provable secure against first-order DPA attacks. As a countermeasure, they proposed to inject fresh random bits in a 3-share implementation

or to add an additional share (4-share version) that avoids the need of fresh randomness.

In this chapter we present a taped-out application-specific integrated circuit called Zorro. Zorro is intended to be used as power-analysis attack evaluation platform. The ASIC has three distinct hardware architectures of a Keccak-based AE algorithm implemented. Two architectures apply three shares as proposed in [16, 19] and in [22], respectively. The third architecture applies four shares as proposed in [22]. In addition to the masking countermeasure, a hiding countermeasure can be activated. Zorro is intended to be used in low-resource applications like embedded systems of RFID-based devices. Due to the limited resources in the target domain, the design goals of the chip were low power consumption and minimal chip area. Zorro was fabricated in a 180 nm CMOS process technology by UMC and the smallest of the three architectures requires only 14.5 kGE. This represents the smallest reported masked Keccak ASIC implementation to date. Beside the un-keyed Keccak implementations available in literature [18, 66, 107], the smallest reported masking-secured designs so far require more than 30 kGE [16, 19, 22].

Zorro was used to evaluate the security of the previously proposed secret sharing schemes by means of non-profiled, HO-DPA attacks. Besides secret sharing, a hiding countermeasure can be enabled in order to further improve the security. Depending on the configuration, zero to fifteen dummy rounds are inserted. The conducted HO-DPA attacks in this work target the linear θ transformation in the first round of Keccak. Here the shares are processed in a sequential order due to the aforementioned design goals. The corresponding leakage samples are combined using the centralized product as combination function, which has been shown to be optimal by Prouff *et al.* [111].

Although secret sharing is frequently proposed as countermeasure to secure Keccak implementations against DPA attacks, we show that HO-DPA attacks targeting a masking-secured, low-resource implementation on a taped-out ASIC can be conducted with low effort. With 3 000 (10 000) measurements the attacker can already reach 2^{nd}-order success rates of 100 % targeting the three-share (four-share) architecture. This significantly reduces the key-search space for a subsequent brute-force attack. By only applying hiding, which introduces additional dummy rounds and randomizes the memory access, the security gain is also not satisfying. For the maximum number of dummy rounds supported by Zorro (15), approximately 2 500 measurements are sufficient for reaching a 2^{nd}-order success rates of 100 %. Therefore we conclude that, especially for low-resource implementations with a low noise level (high signal-to-noise ratio) due to sequential processing, a combination of the countermeasures is required in order to provide an adequate security level. The previous attack results allow us to approximate the security level of the combined countermeasures. In the most secure configuration, security against DPA attacks up to $5.1 \cdot 10^6$ measurements can be achieved. Detailed area numbers of the Zorro ASIC and runtime measurements further allow us to evaluate the area and runtime penalty for given security levels.

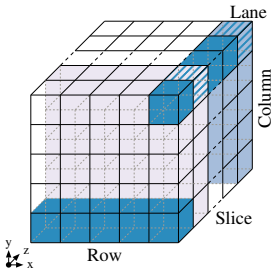


Figure 4.1: KECCAK state.

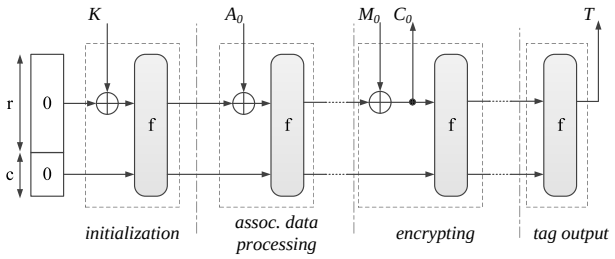


Figure 4.2: The SPONGEWRAP construction.

4.2 Preliminaries

This section provides preliminary information on KECCAK, the SPONGEWRAP mode, and higher-order DPA attacks.

4.2.1 Keccak

KECCAK has been announced as the winner of the NIST SHA-3 competition [99] in 2012. For creating cryptographic hash values, KECCAK applies the KECCAK-f permutation [20] on the input data. The KECCAK-f permutation works on a state with a size of b bits, which is made up of the rate r and the capacity c. The rate r defines the size of the input block that can be processed during one KECCAK-f permutation. The remaining c bits define the security level of the construction, $c = b - r$. KECCAK-f is defined for seven different state sizes, which can be calculated as follows: $b = 25 \cdot w$, where $w = 2^l$ and l is in the range from 0 to 6. The state is organized as a $5 \times 5 \times w$ matrix, where each bit is defined by the set of coordinates (x, y, z). A row is a set of 5 bits with fixed (y, z) coordinates, a column is a set of 5 bits with fixed (x, z) coordinates and a lane is a set of w bits with fixed (x, y) coordinates, see Figure 4.1. One KECCAK-f permutation consists of $12 + 2 \cdot l$ rounds and each round performs the following five transformations on the state:

θ **:** Used to integrate diffusion by a linear mixing layer (the parity of two nearby columns is added to each column).

ρ **:** Inter-slice dispersion (all lanes are rotated by a defined offset).

π **:** Breaking horizontal/vertical alignment (the 25 lanes are transposed in a fixed pattern).

χ **:** The non-linearity part of KECCAK-f (the 5 bits of each row are combined using AND gates and inverters and the shifted result is added to the row).

ι **:** A w-bit round constant is added (XORed) to a single lane.

4.2.2 SpongeWrap

The SpongeWrap construction [17] applies a sponge function to create an authenticated encryption mode. It can be divided into four phases, the initialization phase, the associated-data processing phase, the encryption phase, and the tag generation phase. Figure 4.2 depicts the structure of the SpongeWrap construction. During the initialization phase, the encryption key K is loaded into the cleared state followed by a call to the permutation f. In the second phase, associated data blocks A_i are processed. These blocks are authenticated but not encrypted. Message blocks M_j serve as input during the third phase, here the construction outputs the corresponding cipher text blocks C_j. After the last message block has been processed, the tag generation starts. The result of this last phase is the authentication tag T. For decryption, the corresponding plain text and cipher text blocks are swapped and the resulting tag is compared with the received tag. Only if the two tags are similar, the construction outputs the plain text, otherwise an error message is returned.

4.2.3 Higher-order DPA Attacks

The knowledge of specific intermediate values occurring during the computation of a cryptographic algorithm can significantly decrease the security of this algorithm. We name this specific intermediate values *sensitive intermediate values* v^*_{int}. In a DPA attack scenario, v^*_{int} depends on some known input value p (e.g. one part of the plain text) and some secret value k (e.g. one part of the secret key), we can write $v^*_{int} = f(p, k)$. Here the attacker takes advantage of the relation between the power consumption and the value of v^*_{int} in order to reveal the secret value k. The exact relation between power consumption and v^*_{int} depends on the underlying hardware. One appropriate model for several hardware implementation is the Hamming distance power model. This model assumes that the power consumption is proportional to the number of bits changing in a register at a specific clock cycle.

In order to protect the implementation against DPA attacks, masking has been proposed (e.g., [119]) as countermeasure. A masked implementation splits v^*_{int} into d shares. In case of boolean masking, the $d-1$ shares $s_1 \dots s_{d-1}$ are generated randomly and s_d is calculated according to Equation 4.1. All the calculations are performed on the d shares instead of v^*_{int}. So no correlation between the power consumption and v^*_{int} can be observed.

$$s_d = v^*_{int} \oplus s_1 \oplus \cdots \oplus s_{d-1} \tag{4.1}$$

The leakage L_i $(i = 1 \dots d)$ produced by share s_i can be modeled by Equation 4.2. It consists of a constant part δ_i, the power model (e.g. Hamming distance) of s_i and Gaussian noise B_i with zero mean and standard deviation σ_i $(B_i \sim \mathcal{N}(0, \sigma_i))$.

$$L_i = \delta_i + H(s_i) + B_i \tag{4.2}$$

The previously discussed leakage function shows that a simple DPA attack targeting a masked implementation does not lead to a successful attack. No exploitable relation between any of the shares and the secret value k exists. Higher-order DPA attacks (HO-DPA) have been shown to be a valid method to successfully attack masked implementations. The order of the attack depends on the order of the masking. A d-th order DPA attack has to be mounted to successfully attack an implementation using d shares. In a d-th order DPA attack, the leakages $L_1 \dots L_d$ are combined using a combination function C.

At this point it is important to differ between two scenarios. For the first scenario, called *sequential leakage scenario*, the leakages $L_1 \dots L_d$ appear at *different points in time*, what is mostly the case for software implementations. The first challenge in this scenario is to identify the appropriate points in time for the DPA attack where the leakages appear. Several approaches to identify this time instances have been presented in the past. For the rest of this work we assume that these time instances are already known by the attacker. The second challenge is to find an appropriate combination function for combining the samples. For 2^{nd} order DPA attacks, Prouff *et al.* [111] have analyzed several combination functions and their performance. They found that the product of the centered leakages L_1^* and L_2^* as combination function for L_1 and L_2 (C_{prod^*}) performs best in case of Hamming weight and Hamming distance leakage, see Equation 4.3. This combination function can be extended to d-th order DPA attack by including all the centered leakages $L_1^* \dots L_d^*$ into the product. For the second scenario, called the *parallel leakage scenario*, the leakages $L_1 \dots L_d$ appear at the *same instance in time*, a parallel hardware implementation is an example for this case. Here the combination function is fixed by the hardware, in many cases this is equal to the (potentially weighted) sum of the respective leakages. Raising the sum of the leakages to the power equal the order of the attack is one approach to attack such an implementation, see Equation 4.4. Waddle *et al.* [144] have shown this for 2^{nd} order DPA attacks. If all leakages appear at a single time instance, identifying the appropriate time instances where the single leakages appear (first case) becomes unnecessary.

$$L_i^* = (L_i - E(L_i)) = H(\mathbf{S}_i) + B_i - \frac{n}{2} \qquad C_{prod^*}(L_1, L_2) = L_1^* \times L_2^* \quad (4.3)$$

$$C_{sum}(L_1 \dots L_d) = (L_1^* + \dots + L_d^*)^d \quad (4.4)$$

4.3 The Attacked ASIC Zorro

This section provides an introduction to the hardware architecture of Zorro. The focus is put on information being relevant for the conducted DPA attacks. More implementation details than given in this section can be found in [97].

At the start of the Zorro project, two main motivations for the ASIC have been defined. The first motivation was to create a power-analysis attack evaluation platform. The evaluation targets are countermeasures for securing an

authenticated encryption (AE) algorithm based on the Keccak-f [1600] permutation. The second motivation was to show the applicability of a secured AE algorithm for low-resource devices. Therefore small chip size and low power consumption were the main design goals.

The ASIC contains three distinct architectures of the Keccak-f permutation with a state size of 1600 bits which differ in the applied sharing technique. The first architecture (*3-Share*) applies the three-share approach by Bertoni *et al.* [19]. Bilgin *et al.* [22] have shown that the previously proposed three-share approach by Bertoni *et al.* [19] is not provable secure against 1^{st} order DPA attacks. They provide two solutions to create implementations which are provable-secure against 1^{st} order DPA attacks. One solution is a modified three-share implementation, the second architecture on the Zorro ASIC (*3-Share**) equals this implementation. The other solution to achieve provable security against 1^{st} order DPA attacks is to apply four shares. This approach is used by the third architecture (*4-Share*) implemented on the Zorro ASIC. In addition to the masking countermeasure, the ASIC further features a hiding countermeasure. Here additional dummy rounds are executed and also the memory access is randomized. In order to ensure meaningful power measurements, the clock signal is forwarded only to the unit which is currently under investigation. This approach ensures that the two remaining units do not create any unintended noise.

In order to meet the low-area design goal, a random-access memory (RAM) macro cell is used instead of registers. Storing the 1 600 bit state has the most influence on the required area, the secret-sharing countermeasure further increases the storage requirements. In the case of the variants applying three (four) shares, the memory requirement increases to $3 \cdot 1\,600 = 4\,800$ bit ($4 \cdot 1\,600 = 6\,200$ bit). The datapath contains one 256 bit wide working register which allows to manipulate four lanes or eight slices during one RAM load/store cycle. The Keccak-f permutation consists of one lane-based transformation (ρ), the remaining transformations work on slices. Due to the architecture applying the 256 bit working register, the lane-based and slice-based transformations have to be treated separately in each round. To meet this requirement the datapath consists of a lane unit and a slice unit. The lane unit allows to modify the data in the working register according to the ρ transformation. θ, χ, ι, and π transformations are supported by the slice unit. The lane-based transformation ρ requires seven RAM load/store cycles to manipulate the whole state (the first lane is not modified by ρ). The slice-based transformations require eight RAM load/store cycles to manipulate the whole state.

In order to minimize the overall required RAM load/store cycles a slightly modified round schedule is used. This modified round schedule differs between the following three rounds:

$$R_1 = \theta \times \rho \qquad R_{2\ldots24} = \pi \times \chi \times \iota \times \theta \times \rho \qquad R_{25} = \pi \times \chi \times \iota \tag{4.5}$$

Each of the three architectures, *3-Share*, *3-Share**, and *4-Share*, support four different operation modes, depending on the configuration of the DPA countermeasures. In the following, these four operation modes are introduced:

- In ***Normal Mode (NM)***, no DPA countermeasures of the selected design are enabled. Therefore, only parts of the RAM of the activated architecture are actually used (since no shares are required) and the SpongeWrap construction works fully unprotected.

- When running in ***Hiding Mode (HM)***, the user can choose how many *dummy rounds* the enabled architecture should perform (up to 15). A single dummy round always corresponds to a full Keccak round, which significantly increases the runtime in *HM* when raising the number of dummy rounds. Simultaneously, the data transfer to and from the RAM gets shuffled using eight different possibilities. Thus, the number of time instances ti, where the leakage can appear for a configuration using j dummy rounds, is calculated according to: $ti = 8 + 8 \cdot j$ ($j \in [1 \dots 15]$) Each RAM has a couple of additional entries, which are not initialized. These words are used as inputs when executing the dummy rounds. Thereby no correlation between the actual state and the measured power traces should be observable at all. For the remainder of this work we use *HM-j* to denote Zorro running in hiding mode using j dummy rounds.

- Once the ***Masked Mode (MM)*** is selected, the activated architecture actually operates based on the shares. According to the designs' names, the *3-Share*, *3-Share**, and *4-Share* unit use a three-share approach (as proposed by Bertoni et al. [16]), a three-share approach with re-masking, and a four-share approach (as proposed by Bilgin et al. [22]), respectively. In the following, we use *MM-3* when talking about the architectures applying three shares and *MM-4* in the case of four shares, respectively.

- The most secure mode, supported by each of the three architectures, is the ***Secure Masked Mode (SMM)***, which combines the countermeasures of *HM* and *MM*. Contrary to *NM* and *HM*, where only a third/fourth of the RAM entries are actually used (as well as the uninitialized entries for the dummy rounds), in *MM* and *SMM* all entries are required for processing. We further on refer to Zorro running in *SMM* based on i shares and j dummy rounds using the following notation: *SMM-i-j*

4.4 SCA-Attack Setting

In the following paragraph, the attack scenario is defined. The initial state $\boldsymbol{S}_{init}$ of the algorithm equals the concatenation of the key K, the first message part M (which can either be a part of the associated data or the plain text), and a vector containing $c = 512$ zeros (0^c): $\boldsymbol{S}_{init} = K||M||0^c$. With $r = 1\,088$ bits and $|K| = 256$ bits, the length of the message part in $\boldsymbol{S}_{init}$ equals 832 bits ($|M| = r - |K| = 1\,088 - 256 = 832$ bits). One might argue that this example is very advantageous for an attacker due to the fact that $\boldsymbol{S}_{init}$ contains data which can be freely chosen. But a similar attack would also work if $\boldsymbol{S}_{init}$ does not contain any freely chosen data. Then, instead of recovering the secret key,

the attacker can recover the output of the first KECCAK-f permutation. With the knowledge of the intermediate state it is possible to forge messages without knowing the key. A detailed discussion about the influence of the key-length on the security of MAC-KECCAK can be found in [135]. Furthermore we want to mention that, especially for a low-resource application, what is the target application of ZORRO, it makes sense to use the remaining bits of $\boldsymbol{S}_{init}$ for the message. So the number of executions of the KECCAK-f permutation is minimized.

If no countermeasures against DPA attacks are activated (ZORRO running in *NM*), $\boldsymbol{S}_{init}$ is processed by the round transformations of KECCAK-f. Each slice S_z contains 4 unknown key bits, the remaining 21 bits are known by the attacker and $21 - \frac{c}{64} = 13$ bits, the message part M, can be freely chosen per slice.

If the masked mode using d shares is activated, the round transformations of KECCAK-f are performed on the shares $\boldsymbol{S}^1$, $\boldsymbol{S}^2$, ..., and $\boldsymbol{S}^d$ respectively. In the first execution step, the shares $\boldsymbol{S}^1 \dots \boldsymbol{S}^{d-1}$ are initialized with random values and $\boldsymbol{S}^d$ is calculated according to $\boldsymbol{S}^d = \boldsymbol{S}_{init} \oplus \boldsymbol{S}^1 \oplus \dots \oplus \boldsymbol{S}^{d-1}$. The random values required for initializing the shares have to be provided to the ZORRO ASIC from outside. The fact that $\boldsymbol{S}^1 \dots \boldsymbol{S}^d$ are processed by KECCAK-f does not allow to generate hypothetical intermediate values that are required for a DPA attack.

The θ transformation is the first linear transformation applied on the state and because of the modified round schedule this is the only slice-based transformation in the first round. To minimize the resource consumption, the architecture on the ZORRO ASIC performs the θ transformation on one slice per clock cycle, leading to a total of 64 clock cycles for NM and $d \cdot 64$ clock cycles for the d-share implementation. The RAM load/store cycles are not considered in this discussion. Note that the linear transformations can be applied on each share individually, what enables a sequential processing of the shares. On the one hand, sequential processing allows to reuse the hardware and therefore minimizes the required chip area. On the other hand, the runtime increases.

The fact that the θ transformation processes key information which is constant for every encryption and a message part, which can be chosen by the attacker, makes it a valid target for a DPA setting. According to the modified round schedule, the ZORRO ASIC leaks the Hamming distance of the slice values before and after the first θ transformation. Hamming distance values in the range of 0 and 25 are possible and the fact that in one clock cycle only one slice is modified reduces the algorithmic noise significantly compared to an implementation, where the whole state is transformed in one clock cycle. To calculate the θ transformation for S_z, the information of two neighboring slices ($(z-1)$ mod 64 and z) is required. This leads to a power model with 256 key hypotheses (2 times 4 bits) for each slice.

For *NM*, the DPA attacks could be applied on the measurements without post processing. For *MM*, the time instances, where the leakages appear, were combined using the centralized product [111]. For *HM*, windowing has been applied on the measurements to improve the results [86]. For *SMM*, DPA at-

tacks also require a combination of windowing and higher-order post processing (discussed in e.g. [140]). Possible combinations are discussed in Section 4.5.

4.4.1 Evaluation Metric

For comparing the security of the Zorro ASIC running in the different countermeasure configurations, non-profiled DPA attacks using the Pearson correlation coefficient as distinguisher have been conducted. Preliminary experiments confirmed that the ASIC leaks intermediate values according to the Hamming distance power model. Equation 4.6 is used to approximate the minimum number of required measurements N_{min} in order to distinguish the correct key hypothesis from the wrong key hypotheses. ρ_{K^*} corresponds to the correlation coefficient of the correct key. The N_{min} value is an approximation as lower bound for 100% first-order success rate.

$$N_{min} = (4/\rho_{K^*})^2 \tag{4.6}$$

As a second metric the n-th order success rate, as proposed in [134], is applied. For the investigated countermeasure configuration, N_{full} measurements have been recorded and stored. This set of measurements has been split into k groups G_i $(i = 1 \dots k)$, each containing $N_{group} = \frac{N_{full}}{k}$ measurements. A DPA attack has been conducted for the first l measurements of each group $(l \leq N_{group})$, returning the current rank of the correct key hypothesis. $k^*(l)$ is the number of groups, where the correct key hypothesis is ranked at any position between 1 and n for a specific number of measurements l. The n-th order success rate is then calculated as $\frac{k^*(l)}{k}$. A first-order success rate of 100 % indicates that the attacker already has the correct key after the DPA attack, no further brute-force attach is required. In most scenarios, the DPA attack is only the first step to reduce the search space for the correct key. The value of the correct key is then revealed by performing a brute-force attack using the reduced search space. The number of remaining key candidates k_{bf} for an n-th order success rate and $|k|$ key bytes is then calculated according to Equation 4.7. For small values of n, the number of measurements to reach the n-th order success rate of 100 % is typically higher compared to the value of N_{min} approximated with Equation 4.6 using the corresponding ρ_{K^*}. Wrong key hypotheses can often also lead to high correlation values what corrupt the rank of the correct key hypothesis.

$$k_{bf} = n^{|k|} \tag{4.7}$$

For example, if the number of measurements available for an attack is equal to the number to reach a third-order success rate ($n = 3$) of 100 %, the correct key byte value is among the first three candidates. Here the assumption is that the DPA attack can target each key byte individually. Assuming that the full key length is equal to 128 bit (16 byte, $|k| = 16$), 3^{16} key candidates remain to be tested in a subsequent brute-force attack.

4.4.2 Measurement Setup

For measuring the power consumption during the execution of Keccak-f, the voltage drop across a resistor in the core supply line of the Zorro ASIC was measured. A self-made differential amplifier with a gain of 10 has been used for the measurement. In order to maximize the voltage resolution, the resistor value has been selected accordingly. A *PicoScope 6404c* oscilloscope was used to capture the power traces which were stored on a PC for further analyses. A sampling rate of 1 GS/s and a clock frequency of the ASIC of 10 MHz, leads to 100 samples per clock cycle. The ASIC provides an 8 bit data bus for communication. An FPGA has been used between the controlling computer and the ASIC to modify the serial data received from the computer according to the bus protocol supported by the ASIC. The following data is generated on the computer and sent to the ASIC (via the FPGA): key, message, random numbers to initialize the shares, and configuration data for the countermeasures.

4.5 SCA-Attack Results

In this section the results of DPA attacks targeting Zorro with different countermeasure configurations are presented.

4.5.1 DPA Attacks Targeting Zorro in NM

Preliminary DPA experiments were performed targeting Zorro running in NM. In that mode, no countermeasures are active, so there is no protection against DPA attacks. By applying a 1^{st} order DPA attack 35 measurements are sufficient to reach a first-order success rate of 100 %. In order to reach a second-order success rate of 100 %, 20 measurements of the implementation without activated countermeasures are sufficient. 2^{32} key candidates remain that need to be tested in a subsequent brute-force attack. The DPA attack revealed $\rho_{K^*} = 0.89$ leading to $N_{min} = 20$ by applying Equation 4.6. This preliminary DPA attacks clearly indicate that the integration of countermeasures is inevitable to provide an appropriate level of security.

4.5.2 HO-DPA Attacks Targeting Zorro in MM

When Zorro is running in masked mode, 1^{st} order DPA attacks do not lead to successful results. The correct key hypothesis cannot be distinguished from wrong key hypotheses. Up to 200 000 measurements have been used for this investigation.

For MM-3, a 3^{rd} order DPA attack leads to successful key recovery. Therefore, the time instances where the three shares are processed have been combined by applying the centralized product as discussed in Section 4.2. The appropriate time instances which need to be combined were identified by a visual inspection of the recorded power trace. Figure 4.3 depicts one power trace of the first Keccak-f round of Zorro running in MM-3. The time intervals where the first θ

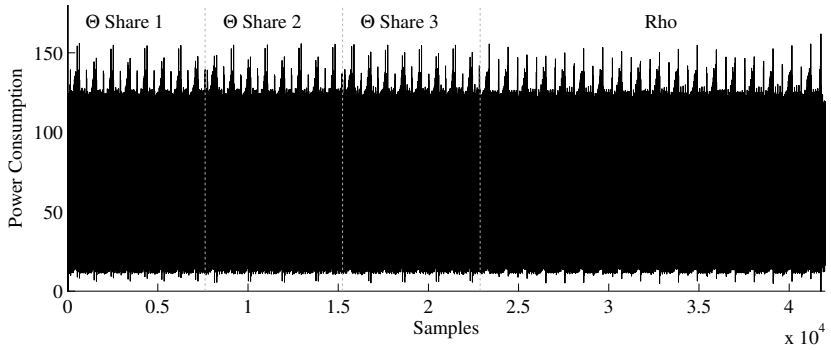


Figure 4.3: Power trace of the first KECCAK-f round of ZORRO running in MM-3.

transformation is executed on the shares are marked with the dotted, vertical lines. The same approach has been followed for the evaluation of MM-4, with the only difference that now four shares were combined, leading to a 4^{th} order DPA attack. In order to discuss the HO-DPA results targeting *MM-3* and *MM-4*, respectively, we apply two metrics. First, the success rate and second, the absolute value of the correlation coefficient corresponding to the correct key hypothesis ρ_{K^*}. Knowledge of the value ρ_{K^*} allows us to approximate the number of required measurements for a key recovery based on Equation 4.6.

For the three-share variant (*MM-3*), 10 000 measurements are sufficient to reach a first-order success rate of 100 %. With that amount of measurements no subsequent brute-force attack is required, the key search space is already reduced to 1. If a subsequent brute-force attack is considered, a lower number of measurements is sufficient in order to reveal the key. A second-order success rate of 100 % leads to a remaining key search space of 2^{32} when considering the key length of 256 bit. A second-order success rate of 100 % is reached with 3 000 measurements for *MM-3*. For *MM-3*, $\rho_{K^*} = 0.119$. By applying Equation 4.6, $N_{min} \approx 1\,130$ can be approximated as lower bound. The results of the practical DPA attacks targeting the *MM* are summarized in Table 4.1.

For the four-share variant (*MM-4*), 29 000 measurements are sufficient to reach a first-order success rate of 100 %. A DPA attack with that amount of measurements reveals the key without the need for a subsequent brute-force attack. If a subsequent brute-force attack is considered, the second-order success rate is evaluated. A second-order success rate of 100 % is reached with 10 000 measurements for *MM-4*. This leads to 2^{32} keys which need to be tested in a subsequent brute-force attack. For *MM-4*, $\rho_{K^*} = 0.06$. By applying Equation 4.6, $N_{min} \approx 4\,450$ can be approximated as lower bound.

Two things need to be clarified when talking about the result targeting *MM-3* and *MM-4*, respectively. First, the results for *3-Share* and *3-Share** are similar. This is because the performed DPA attacks target the first θ transformation while the modification for *3-Share** affects the first χ transformation. Therefore we do not differ between the two designs in the discussion dealing with *MM-3*.

Second, for the HO-DPA attacks we do not consider any overhead for searching the appropriate time samples which need to be combined.

4.5.3 DPA-Attacks Targeting Zorro in HM

In the following, the hiding countermeasure of the chip is evaluated in detail. In *HM*, additional rounds on a random state are inserted and also the load sequence from the RAM is randomized. The number of time instances ti where the targeted leakage can appear is in the range of $ti \in [16, 24, \ldots, 128]$, depending on the configuration.

Table 4.1 summarizes the results of the practical DPA attacks targeting the hiding-secured configuration. All results have been achieved by applying 1^{st} order DPA attacks after windowing has been applied on the recorded measurements. Similar to the discussions in the previous section regarding Zorro running in *MM*, we evaluate the number of traces required to reach a 1^{st} order and 2^{nd} order success rate of 100 %, respectively. Attacks targeting the configuration using one dummy round (*HM-1*) require 1 000 measurements to reach a 1^{st} order success rate of 100 % and 800 measurements to reach a 2^{nd} order success rate of 100 %. For the most secure hiding-mode configuration, *HM-15*, the required number of measurements increases to 3 300 and 2 500, respectively. For approximating N_{min}, the correlation value of the correct key hypothesis ρ_{K^*} has been used.

4.5.4 Summary of Hiding and Masking Applied Independently

Although none of the designs on Zorro is a completely countermeasures-free architecture (i.e., even though they can be disabled, the required hardware remains present), Zorro is comparable in terms of the required area with the *plain* Keccak implementation by Pessl and Hutter [107] due to its similarities. Therefore, we use the 5.5 kGEs as a reference for the required area, which could

Table 4.1: Results of the practical DPA attacks targeting Zorro running in *HM* and *MM*.

Mode	ρ_{K^*}	N_{min}	100 % success rate	
			1^{st} order	2^{nd} order
		[measurements]	[measurements]	[measurements]
Plain	0.890	20	35	20
MM-3	0.119	1 130	10 000	3 000
MM-4	0.060	4 450	29 000	10 000
HM-1	0.194	430	1 000	800
HM-2	0.170	560	1 200	900
HM-3	0.152	690	1 500	1 000
HM-4	0.139	830	1 600	1 200
...				
HM-15	0.077	2 700	3 300	2 500

Table 4.2: Performance comparison of selected countermeasure configurations of the Zorro ASIC.

Mode	Shares	Dummy R.	N_{min} [measurements]	Area [kGE]	Runtime [kCycles]
Plain	-	-	20	5.5	21.9
MM	3	-	1 130	14.0	113.2
MM	4	-	4 450	17.0	149.6
HM	-	1	430	5.5	23.2
HM	-	15	2 700	5.5	36.5
SMM	3	1	55 000	14.0	121.0
SMM	3	2	82 000	14.0	125.7
SMM	3	5	163 000	14.0	139.7
SMM	3	15	434 000	14.0	186.2
SMM	4	1	285 000	17.0	160.0
SMM	4	2	427 000	17.0	166.1
SMM	4	5	853 000	17.0	184.6
SMM	4	15	2 276 000	17.0	246.1

be reached when all the resources needed for the countermeasures, for instance, larger RAMs and intermediate registers, the non-linear processing unit for the shares as well as all the controlling overhead required to provide a variable DPA evaluation platform such as Zorro, are not present. From a security point of view the design by Pessl and Hutter is comparable with Zorro running in *NM*. Adding a hiding-only countermeasure to an unsecured (plain) Keccak design usually requires to add some minor overhead to the controlling unit. If the area needed for both this controlling and the generation/provision of the random numbers is very small, the overall area overhead for adding hiding can be seen as negligible. With regard to the runtime, an increase ranging between 6% for *HM-1* and 67% for *HM-15* must be accepted, depending on the number of dummy rounds. However, running Zorro in *HM* assures that the number of required traces increases by a factor between 22 (for *HM-1*) and 137 (for *HM-15*). The three-share design results in an area overhead of 155%, while the required traces increase by a factor of 57. Approximately 225 times more traces are required if the four-share approach is used, leading to an area increase of 209%. Also the runtime for one Keccak-f permutation increases by 417% and 584% for the three-share and the four-share architecture, respectively. The upper half of Table 4.2 summarizes the numbers of our results in *HM* and *MM*.

4.5.5 Security Approximation for Zorro in SMM

The results presented so far indicate that both countermeasures alone do not provide sufficient protection against DPA. Hence, in this section, the most secure mode of Zorro, combining hiding and masking techniques, is discussed.

Due to the increased measurement effort, results for *SMM* are approximated using an appropriate equation derived in the following. This equation takes as input the ρ_{K^*} values from the HO-DPA attacks targeting the masked implementations. For *SMM-3-1* we have also verified the correctness of the approximation

by performing a practical attack.

For attacks on the combined countermeasures (i.e., on *SMM*), we first discuss the effect of the sequence of applying windowing and higher-order post processing of the measurement data on the outcome of the DPA attack. Tillich *et al.* [140] discuss four different approaches how to attack a randomized and first-order masked AES software implementation: Biasing the mask, taking advantage of weak randomization, combining 2^{nd} order DPA and windowing, and classical 2^{nd} order DPA on windowed traces. For our scenarios, only the last two approaches are of interest which need to be modified as discussed in the following.

Combining *i*-th order DPA and windowing The *first step* of this approach consists in an i-th order post processing. The applied post-processing function takes as inputs the values at the i time samples where the targeted intermediate value is processed. Without hiding, the function is applied once. In the case of hiding, all possibilities where leakage can appear have to be evaluated by the function. For Zorro, the targeted leakage can appear at eight different time samples per share leading to 8^i evaluations of the post-processing function per round. For *SMM-i-j*, that leads to $(j+1) \cdot 8^i$ evaluations overall, where j denotes the number of dummy rounds. The *second step* of this approach consists in windowing. For this purpose, the $(j+1) \cdot 8^i$ results of the i-th order post-processing function are summed up. The resulting correlation coefficient ρ_{iW} decreases compared to the correlation coefficient of the masking-secured implementation ρ_{MMi} by a factor of $\sqrt{(j+1) \cdot 8^i}$, see Equation (4.8).

$$\rho_{iW} = \frac{\rho_{MMi}}{\sqrt{(j+1) \cdot 8^i}} = \frac{\rho_{MMi}}{\sqrt{\frac{ti}{8} \cdot 8^i}} = \frac{\rho_{MMi}}{\sqrt{ti} \cdot \sqrt{8^{i-1}}} \tag{4.8}$$

Apply *i*-th order DPA on windowed traces The *first step* of this approach consists in windowing. All the corresponding time instances per share are therefore summed up, the number of time instances ti for *SMM-i-j* is equal to $8 + 8 \cdot j$. The result of this windowing are i values, which serve as the inputs for the i-th order post-processing function applied next. The *second step* consists in executing the i-th order post-processing function using as inputs the output values of the previous windowing step. According to [86], the correlation coefficient for the hiding countermeasure compared to an unsecured implementation is reduced by the factor $\sqrt{ti}$ if windowing is applied. As a result, we come up with Equation (4.9). ρ_{MMi} is the correlation coefficient of the masking-secured implementation.

$$\rho_{Wi} = \frac{\rho_{MMi}}{\sqrt{ti \cdot i}} \tag{4.9}$$

Comparison of the two approaches In order to figure out which of the two previously discussed approaches is better suited for attacks on the combination of secret sharing and hiding countermeasures, we examine which approach leads to higher correlation values for the parameters $i \in [3, 4]$ and $ti \in [16, 24, \ldots, 128]$. By combining Equations (4.8) and (4.9), the relation $\rho_{iW} = \rho_{Wi} \cdot \sqrt{\frac{i}{8^{i-1}}}$ can be found and therefore $\rho_{iW} < \rho_{Wi}\ \forall i > 1$. Performing windowing first and

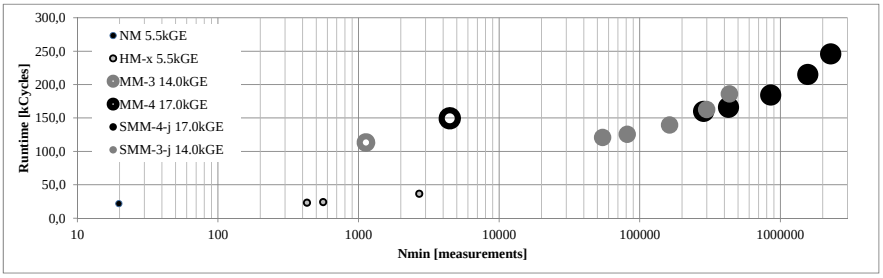


Figure 4.4: Runtime/N_{min} comparison of Zorro for all the provided modes.

afterwards executing the higher-order post processing is the better choice and will be used for the following discussions.

Security gain in SMM Equation (4.9) allows us now to approximate the security gain for the *SMM* of Zorro based on the previous results from the *practical HO-DPA attacks on the masked implementations*. Therefore we use $\rho_{MM3} = 0.119$ for *SMM-3-j* and $\rho_{MM4} = 0.060$ for *SMM-4-j*. j defines the number of dummy rounds and is used to calculate $ti = 8 + 8 \cdot j$. By performing a practical attack targeting Zorro running in *SMM-3-1* the theoretical approximation has been validated. An attack using 200 000 measurements yields a correlation value of 0.014, therefore N_{min} is about 80 000 measurements by applying Equation 4.6. This value is in line with the lower bound of 55 000 approximated with Equation 4.9.

Table 4.2 gives a summary of the security in terms of number of traces required to successfully mount a non-profiled HO-DPA attack on Zorro running in the corresponding mode. Furthermore, the runtime and area values are given for comparison with the unprotected implementation. Figure 4.4 depicts the relation between N_{min}, runtime and area. Table 4.2 and Figure 4.4 reveal some interesting facts:

- N_{min} for *HM-15* is higher compared to N_{min} for *MM-3*. Hence, if only a single countermeasure is used, it is the better choice to only use hiding as the overhead in terms of area and runtime is by far smaller but still provides a better security against non-profiled DPA attacks.

- For N_{min} values up to 300 000 measurements, *SMM-3-j* is always the better choice compared to *SMM-4-j* in terms of area and runtime overhead.

- For *SMM-3-15*, an N_{min} value of 434 000 can be provided. This is the most secure setting when three shares are used.

- For *SMM-4-15*, an N_{min} value of 2 276 000 can be achieved. From all the countermeasure settings provided by Zorro, this is the most-secure one.

4.6 Discussion

In this chapter we have investigated the vulnerability of a low-resource Keccak implementation against non-profiled DPA attacks. For the investigations a taped-out ASIC has been used. This ASIC, called Zorro, provides a flexible DPA-countermeasure evaluation platform. Masking and hiding countermeasures are supported to secure the low-resource Keccak implementation against DPA attacks. Keccak is operated in a keyed mode called SpongeWrap.

The results of the DPA attacks reveal that the masking countermeasure and the hiding countermeasure applied alone do not lead to a satisfying security gain against DPA attacks. This is especially true when considering the large area and runtime overhead introduced by the countermeasures. The main reason for the small security gain is the low-resource design of the ASIC. The sequential processing which allows to reuse many hardware blocks and therefore minimizes the occupied area, significantly increases the signal-to-noise ratio. A high signal-to-noise ratio is advantageous for DPA attacks.

In order to achieve an acceptable security gain, especially for low-resource implementations, it is required to combine both countermeasures. Security level, area overhead, and runtime overhead introduced by the countermeasures are influenced by the selection of the configuration parameters of the countermeasures. These parameters are the number of shares and the number of executed dummy operations.

Another important observation is that the hiding countermeasure applied alone achieves a higher security gain compared to the masking countermeasure using three shares. It has to be considered that the area and runtime overhead introduced by the hiding countermeasure is smaller compared to the masking countermeasure. Therefore, if only a single countermeasure is applied, hiding performs better compared to masking on low-resource devices.

All results presented in this chapter were achieved by using a low-cost power measurement setup. The prototype ASIC is intended to be used in RFID applications, making power measurements infeasible. But the results achieved in Chapter 2 let us come to the conclusion that low-cost SCA attacks in close proximity using the EM side channel would lead to comparable results. With disabled countermeasures, also remote SCA attacks should be successful. For enabled countermeasures in the remote SCA scenario, we refer to the conclusions from Chapter 2.

5 Non-Invasive Fault Attacks

The aim of this chapter is to present fault-injection setups for non-invasive fault attacks applicable for a wide range of off-the-shelf microcontrollers. In particular, the influence of tampering with the clock frequency, the power supply, and the ambient temperature on the following microcontroller platforms has been investigated: Atmel ATmega 162/v, Atmel ATxmega 256, and ARM Cortex-M0. Not only the single injection techniques have been investigated, also combinations of the techniques have been applied in order to improve the success rate of fault injections. Results of the conducted experiments approve the relevance of the presented low-cost setup. The results presented in this chapter have been published in [75] and [77], respectively. The contributions can be summarized as follows.

Contribution

- Introduction of a flexible, low-cost fault-injection setup allowing to tamper with the power supply, the clock signal and the temperature of the device under test.
- Investigation of the effects of similar fault injection methods on two different microcontroller platforms.
- Studying the influence of combining fault-injection methods on the success of the fault injection. Clock glitches during underpowering was the first investigated combination. The second combination was a clock glitch while operating the device outside the specified temperature range. Both combinations allow to increase the fault-injection success.

This chapter is structured as follows. In Section 5.1 we give an introduction to non-invasive fault attacks in the context of sensor nodes. Next, in Section 5.2 we discuss the fault injection environment and in Section 5.3 a description of the fault-injection experiments is given. The results of the experiments are discussed in Section 5.4 and Section 5.5 completes the chapter with a short conclusion.

5.1 Introduction

Fault attacks pose a serious threat for implementations of cryptographic algorithms. Their aim is to reveal secret information (e.g. a secret key) used by these algorithms by forcing faulty behavior. Examples in the past have shown that implementations that are not secured by means of countermeasures can successfully be attacked by injecting a single fault. Non-invasive fault attacks do not even require a modification of the attacked device. Popular techniques for injecting faults in a non-invasive way are tampering with the supply voltage [150], the clock signal [12], the temperature [56], or injecting EM pulses [95].

In order to conduct the aforementioned attacks, physical access to the attacked device is required. In case of non-invasive attacks, the device can be put in place after the attack, so the probability, that the attack remains undetected is high. Sensor nodes are potential targets for physical attacks. They often operate in hostile environments and are only sporadically observed by human beings. This fact makes it easy for an attacker to remove the node for the duration of the attack and put it in place afterwards. For the analyses in this work we have chosen three MCUs frequently used as central processing unit in sensor nodes, an Atmel ATmega 162/v, an Atmel ATxmega 256, and an ARM Cortex-M0.

A huge number of non-invasive fault attacks on cryptographic primitives have been reported in literature during the last years. Popular attack targets are hardware as well as software implementations of block ciphers and software implementations of the RSA algorithm. Fault attacks can be grouped into attacks that insert a fault in the datapath or the control logic.

The fault attack performed in this chapter apply clock tampering, power-supply tampering and temperature variation and their combinations. We do not target a specific cryptographic algorithm, the focus is put on the impact of the injected fault on a set of instructions executed on microcontrollers. When studying related work, there have been many fault attacks targeting a specific cryptographic algorithm using one of the previously mentioned fault-injection methods. Some examples are given in the next paragraphs.

Tampering with the power supply has been shown to be an effective, non-invasive fault-injection method. Selmane *et al.* [120] have shown, how setup-time violations due to underpowering can be used to attack the Advanced Encryption Standard (AES) block cipher. Barenghi *et al.* [13] present successful fault attacks targeting software implementations of AES and RSA. The implementations are executed on an ARM 9 general purpose CPU and underpowering is used to inject the faults. Choukri and Tunstall take advantage of power supply glitches in order to modify the round counter of a secret-key algorithm in

[33]. A similar approach to modify the AES round counter is presented in [35]. Here, the authors use electromagnetic pulses in order to enforce the faulty behavior. Schmidt *et al.* [116] use spikes in the power supply to attack an RSA implementation. Kim and Quisquater [68] show that with two precisely timed power glitches the countermeasures of a secured RSA implementation can be circumvented.

If the targeted device is supplied with an external clock signal, tampering with this clock signal has been shown to be an effective method for fault injection. Agoyan, Dutertre, and Naccache [2] apply an FPGA implementation of AES to perform fault attacks using clock glitches. With practical results they confirm their theoretical assumptions. Practical fault attacks targeting six different block ciphers implemented on an LSI have been reported by Fukunaga, Toshinori and Takahashi in [43]. The authors have used clock glitches in order to inject faults.

Not only tampering the clock signal or the power supply, also modifying the ambient temperature has been found to be a valid method forcing faulty behavior. Here, two effects can be observed, depending if the analyzed device is cooled down or heated up.

First, cooling down the device allows to increase the data-retention time. Skorobogatov [124] was one of the first performing data-retention attacks on SRAM chips in 2002. He decreased the ambient temperature of these chips by cooling the devices down to $-20\,^{\circ}\mathrm{C}$ and below. He showed that data gets somehow *frozen* and can be read out after some seconds after power down. Samyde, Skorobogatov, Anderson, and Quisquater made similar experiments published in the same year in [115]. Another similar experiment was done by Müller and Spreitzenbarth [98] in 2011. They developed a tool called FROST (forensic recovery of scrambled telephones), which allows to recover the RAM content of modern Android smart phones. The tool allows to retrieve disk encryption keys from RAM and the approach is comparable to cold boot attacks on PCs [48].

Second, increasing the ambient temperature of the device allows to change memory content. Quisquater and Samyde [112] were one of the first who observed that high temperature causes memory errors after hours of extensive heating. Govindavajhala and Appel [47] were able to induce errors into memories using a 50 watt spotlight clip-on lamp. By heating a device up to $100\,^{\circ}\mathrm{C}$, they were able to inject faults with a probability of 71.4 %. Recently, Hutter and Schmidt [56] presented heating fault-attacks on an AVR microcontroller in 2014. They operated the device above the temperature specification ($> 125\,^{\circ}\mathrm{C}$). The authors verify the efficiency of this high-temperature attack by successfully attacking an RSA implementation.

Studying the existing literature on non-invasive fault injection, we found that so far the combination of different fault-injection methodologies has not been studied in detail. Therefore we decided to close this gap by studying the effects of the combinations clock tampering and underpowering as well as clock tampering and temperature variations on the ability to inject faults. All attacks have been conducted by applying low-cost equipment consisting of off-the-shelf parts.

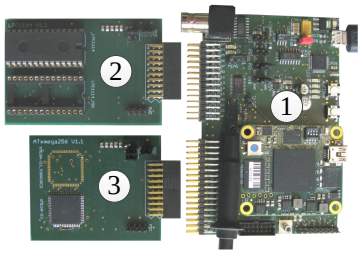


Figure 5.1: (1): Fault board; (2): ARM Cortex-M0 extension board; (3): ATxmega 256 extension board.

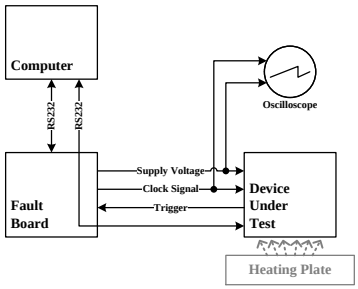


Figure 5.2: Block diagram of the fault injection environment.

5.2 Fault Injection Environment

In this section the setup for the non-invasive fault injections is introduced. This setup was designed as part of this thesis. For the design we followed the approach of using an *FPGA-based fault board* which offers a connector allowing to connect *custom extension boards*. For tampering with the ambient temperature, a heating plate has been used.

5.2.1 Custom-made Fault Board

The custom-made fault board allows to tamper with the clock signal and the power supply provided to the device under test. The main part of this fault board, which is depicted in Figure 5.1, is a *XILINX Spartan-6 XC6SLX45* FPGA. For communication with the control computer, the fault board is equipped with a USB port and the configuration is done via MATLAB® scripts on the control computer. A huge number of I/O pins for connecting a wide range of devices are available. For the experiments performed in this chapter, three pins are essential: the *clock signal*, the *supply voltage*, and the *trigger signal*. A block diagram showing the setup can be found in Figure 5.2. For visualizing the clock signal and the supply voltage, a *PicoScope 5203* from *Pico Technology*[1] has been used.

For clock-glitch generation, a similar approach like presented in [2, 38] is applied to generate the clock signal and insert glitches into it. The shape of the clock glitch can be parameterized by two values, i.e., d_1 and d_2. The first value d_1 defines the time between the positive and the negative clock edge during the clock glitch. The second value d_2 defines the time between two subsequent, positive clock edges during the clock glitch. Both values in combination define the final *shape* of the inserted clock glitch. Figure 5.3 shows the unaltered clock signal *clk* and the altered clock signal clk_{gl} for a small value of d_1 (meaning that

[1] `http://www.picotech.com/`

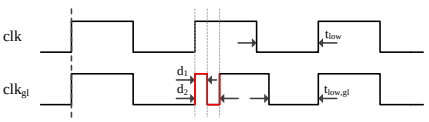


Figure 5.3: Clock-glitch generation for a small value of d_1.

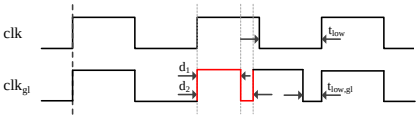


Figure 5.4: Clock-glitch generation for a large value of d_1. Note that $t_{low,gl}$ is lower compared to t_{low}.

the negative clock edge appears early after a positive clock edge). Figure 5.4 shows the signals for a bigger value of d_1 (meaning that the negative clock edge of the altered clock signal appears only slightly before the negative clock edge of the unaltered clock signal). The figures also show the low times t_{low}, defined as the time between negative and positive clock edge of the clock signal. By having a closer look at the two figures, it shows that the low times of the glitch-injected clock signal clk_{gl} is different and depends on the parameter d_1. In particular, $t_{low,gl} = t_{low} - d_1$; so the low time becomes shorter the higher the value of d_1. This means that the negative clock level becomes shorter the later the clock glitch is injected during the positive clock level. This effect appears because of the usage of the 'Digital Clock Managers' (DCMs) of the FPGA for clock-glitch generation.

When using a nominal clock frequency of 24 MHz ($T \approx 41.7$ ns), values for d_2 between 5.9 ns and 22.0 ns can be achieved with this setup, which equals a frequency range between 45 MHz and 170 MHz. Figure 5.5 depicts clock signals generated with the fault board. d_2 values in the range of 8.0 ns up to 22.0 ns, with a step size of 1.0 ns, were measured for creating this plot. The glitch is inserted after a trigger event on a predefined pin of the FPGA. Defining the number of clock cycles between the trigger event and the glitch insertion allows to precisely select the point in time for the glitch to occur.

For supply-voltage tampering, a multiplexer with different, configurable voltage values at its inputs is used. Voltage values in the range of 0 V and 5 V are supported. If the nominal supply voltage of the attacked device equals $U_1 = 3.3\,V$, $U_{Glitch} = (U_1 - U_{diff})\,V$ defines the voltage during underpowering. U_1 equals the voltage at multiplexer input 1 and U_{Glitch} equals the voltage at multiplexer input 2, respectively. U_{diff} defines the reduction of the supply voltage during underpowering. The supply voltage reduction takes place a defined number of clock cycles after a trigger event and also the duration of the underpowering can be defined precisely.

5.2.2 Extension Boards

Figure 5.1 also depicts the two extension boards for the ATxmega 256 and the ARM Cortex-M0 microcontroller respectively. For the heating experiments, conducted with the ATmega 162/v microcontroller, we had to take care that the fault board can be placed at sufficient distance to the heating plate. Otherwise

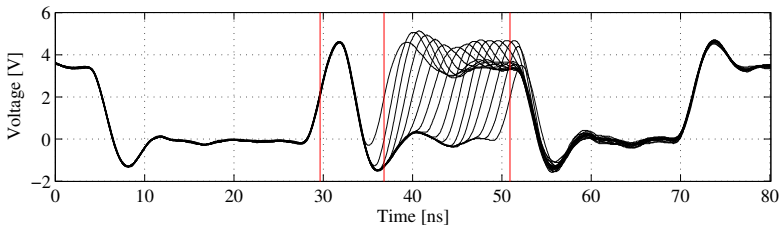


Figure 5.5: Clock signal with different settings for d_2 between 8 ns and 22 ns.

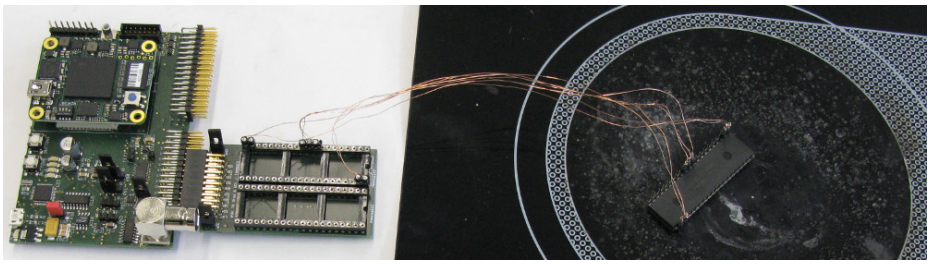


Figure 5.6: The experimental setup: The fault board is located on the left side and the microcontroller is connected to it using thin copper wires. This allows to place the microcontroller in the middle of the heating plate.

the parts on the fault board, especially the FPGA, would also heat up, what might lead to damages on the fault board. Instead of an extension board, the required pins of the ATmega 162/v microcontroller were connected with thin copper wires (0.2 mm diameter) to the fault board. This allowed us to place the custom-made fault board in appropriate distance to the heating plate, as depicted in Figure 5.6. In the following a short summary of the main features of the investigated microcontrollers is given.

ARM Cortex-M0 The ARM Cortex-M0 represents the lower end of the ARM Cortex-M family. It is a 32 bit RISC processor with Von-Neumann architecture. It supports 56 instructions, most of them belonging to the ARM *Thumb* instruction set. For instruction execution, the ARM Cortex-M0 applies a three-stage pipeline as depicted in Figure 5.7 with the three stages: *Fetch stage*, *decode stage*, and *execute stage*. Most operations of the ARM Cortex-M0 are performed with 16-bit instructions. As the CPU provides a 32-bit bus system, in every second cycle two 16-bit instructions are fetched in parallel. For the experiments performed in this work, NXP's LPC 1114 implementation of the ARM Cortex-M0 has been used [102]. This implementation supports a maximum clock frequency f_{max} of 50 MHz and a supply voltage range between 1.8 V and 3.6 V. For the performed attacks on the ARM Cortex-M0, a nominal clock frequency of 24 MHz is used, which equals a factor of 0.48 compared to f_{max}. When using clock glitches with the maximal clock glitch frequency of 170 MHz the maximal operating frequency f_{max} is exceeded by factor of 3.40. A nominal

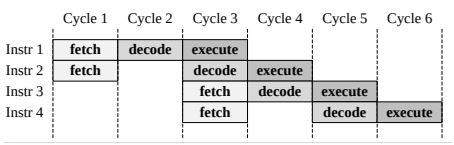


Figure 5.7: ARM Cortex-M0 instruction execution procedure.

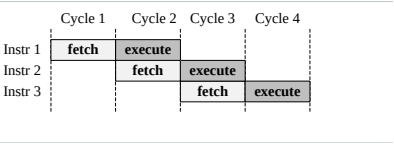


Figure 5.8: ATxmega 256 instruction execution procedure.

supply voltage of 3.3 V is used during the practical experiments on the ARM Cortex-M0.

Atmel ATxmega 256 The ATxmega 256 is a 8/16 bit microcontroller for low-power applications with a RISC architecture. The separated memories for data and program code make the ATxmega 256 a CPU with Harvard architecture. It has a two-stage instruction pipeline with one *fetch stage* and one *execute stage* as shown in Figure 5.8. During the execution of an instruction, the next instruction is simultaneously loaded from the program memory. Using a 16-bit program memory bus allows to load a 16-bit instructions in a single clock cycle. The ATxmega 256 supports 142 Atmel AVR instructions, most of them executing in a single clock cycle. For further information the authors refer to the datasheet of the microcontroller [10]. For our practical experiments we use the ATxmega 256A3 which supports a maximal operating frequency f_{max} of 32 MHz. The supply voltage range is specified between 1.6 V and 3.6 V. When using a clock frequency above 12 MHz at least 2.7 V are recommended. For the attack experiments performed in this work, the ATxmega 256 operates on a nominal clock frequency of 24 MHz, which equals a factor of 0.75 compared to f_{max}. The maximal clock glitch frequency of 170 MHz exceeds the maximal operating frequency f_{max} by factor of 5.31. For all attacks a nominal supply voltage of 3.3 V is used.

Atmel ATmega162 The ATmega 162/v is an 8-bit low-power microcontroller from Atmel. It is part of the AVR family and is based on a RISC architecture. The ATmega162 supports 131 instructions where most of them are single-cycle operations. It provides 32 internal general-purpose registers (denoted by `R0` ... `R31`) that can be used by applications. Some of them are dedicated to special functions such as the registers `R0` and `R1` which store the result of a multiplication, or the sets (`R26`,`R27`), (`R28`,`R29`), and (`R30`,`R31`) which can be used for memory addressing purposes (they are referred to registers `X`, `Y`, and `Z` in the documentation). It further has a 1 kB of internal SRAM and 16 kB of programmable flash memory. Further information about the ATmega162 can be found in the datasheet [9]. The device can be clocked up to 8 MHz with the internal clock source or up to 16 MHz using an external clock ($f_{max} = 16$ MHz). The supply voltage range is specified between 2.7 V and 5.5 V, for clock frequencies above 8 MHz a minimum supply voltage of 4.5 V is recommended. During the practical experiments, two nominal clock frequencies were examined: $f_1 = 10$ MHz and $f_2 = 20$ MHz. The factor between f_1 and f_{max} equals 0.63 and the factor between f_2 and f_{max} equals 1.25. For the second case this already

indicates a slight overclocking, but when operating the device under normal conditions no erroneous behavior was observed. The maximal clock glitch frequency of 170 MHz exceeds the maximal operating frequency f_{max} by factor of 10.63. For all attacks a nominal supply voltage of 4.5 V is used.

5.2.3 Heating Equipment

In order to tamper with the ambient temperature of the device under test, it has been placed on top of a heating plate. For that purpose, a laboratory heating plate from Schott instruments (SLK 1) was used. This equipment does not allow to accurately control the temperature, but measuring the temperature and regulating the heating power figured out to be sufficient. For temperature measurements, a PT100 sensor element has been used. This element changes the resistance according to the temperature, for 0° C the resistance equals 100 Ω. The sensor element has been placed between heating plate and chip package and the resistance of the sensor element was measured with a Fluke 111 TRUE RMS multimeter. After subtracting the resistance of the connection wires, the current temperature has been calculated with an online tool[2]. In the following it figured out that the used heating-plate model introduces electromagnetic interferences. This observation did not affect our experiments but it has to be addressed when a high signal quality is required. Power measurements in a DPA attack scenario require a high signal quality in order to succeed. Therefore we further tested a resistor-based heating element instead of the heating plate. Due to this modification the electromagnetic interferences disappeared.

5.3 Experiment Description

In this section we describe the experiments which were preformed for the investigations of non-invasive fault injections targeting microcontrollers. First, the influence of fault injections using clock glitches targeting the ARM Cortex-M0 microcontroller and the ATxmega 256 microcontroller is investigated. Here the goal is to evaluate the vulnerability of similar instructions executed on two different microcontroller platforms to similar fault injections, especially clock glitches. Second, we study the combination of different fault injection methods to increase the reliability of the fault injections. We study the combination of clock glitches and underpowering targeting the ARM Cortex-M0 microcontroller. The impact of the ambient temperature on the success of clock-glitch attacks targeting specific instructions executed on the ATmega 162/v microcontroller is further analyzed.

[2] `http://www.thermibel.be/documents/pt100/conv-rtd.xml`

5.3.1 Fault-Injection Impact on Different Microcontroller Platforms

When performing fault injections targeting a microcontroller by applying clock glitches, two important characteristics have to be considered. First, an instruction is executed in several CPU stages according to the underlying instruction pipeline. Second, the results may vary depending on the used glitch parameters. It turned out that the main influencing parameter for the experiments discussed in this section is d_2. Therefore we will refer to this value when discussing the results. A fault during the fetch stage can lead to an execution of a wrong instruction due to reading from a wrong address or due to an alteration of the instruction. If the instruction is altered during the fetch stage, an exception might occur due to an invalid instruction. A fault during the decode stage can lead to a misinterpreted instruction. The effect of a fault during the execution stage highly depends on the executed instruction. Either the calculation can be influenced, registers are updated with wrong values, or the register update is skipped at all.

In order to draw meaningful comparisons between the two microcontroller platforms, we target single pipeline stages with the clock glitch. Furthermore, the following three different classes of instructions are evaluated separately: arithmetical/logical instructions, branch instructions, and memory instructions. An overview of the actually examined instructions is given in Table 5.1. For arithmetical/logical instructions, addition, multiplication and a logical left-shift have been evaluated. `Rd` and `Rn` denote the registers for the two operands and the result is stored to `Rd`. Conditional branches have been chosen to evaluate branch instructions and load/store operations have been chosen to evaluate memory instructions.

Studying the impact of the fault injections on the previously defined instructions is the main goal of the experiments. To allow a fair comparison between both microcontroller platforms, a similar test procedure has been used for all instructions, independent of the device executing it. In order to avoid undesirable side-effects, the targeted instruction has been placed between two blocks of `nops`. The procedure consists of the following five steps:

1. **Initialization of the microcontroller:** During the initialization phase, the configuration of the microcontroller is performed. The clock system

Table 5.1: Examined instructions.

Instruction class	ATxmega 256	ARM Cortex-M0
Arithmetical/Logical	`add Rd,Rn`	`adds Rd,Rn`
	`mul Rd,Rn`	`muls Rd,Rn`
		`lsls Rd,#imm`
Branch	`breq label`	`beq label`
Memory	`ld Rd,X`	`ldr Rd,[Rn]`
	`str X,Rn`	`str Rd,[Rn]`

is configured for using an external clock provided by the fault board, I/O pins and the UART interface are configured appropriately. Furthermore, the SRAM and all the registers are initialized with defined values allowing to detect eventual impacts of the fault on memory content.

2. **Synchronization with the control computer:** Via a UART message the microcontroller signalizes the control computer the completion of the initialization phase and waits for further commands. The reception of this message triggers the configuration of the fault board by the control computer. This configuration includes the length, duration, and position of the clock glitch and the event triggering the clock glitch. Finally, the control computer resumes the program execution on the microcontroller.

3. **Rise trigger pin:** In order to allow a precise insertion of the clock glitch, a defined number of clock cycles before the execution of the targeted instruction a trigger signal is generated.

4. **Execution of the targeted instruction:** The targeted instruction gets executed a fixed number of clock cycles after the trigger signal. This instruction is surrounded by a number of `nops` in order to prevent any side-effects due to the clock glitch.

5. **Result communication:** In the last step, the values of the CPU and general purpose registers are transferred to the control computer where they get compared to values of a fault-free reference execution. The result of this comparison together with the fault-injection parameters are stored for later evaluation. The clock cycle when the fault was injected allows to draw conclusions about the affected pipeline stage of the target instructions execution procedure.

5.3.2 Combination of Fault Injection Methods

In the following paragraphs the experiments for evaluating combinations of fault-injection methods are introduced.

Clock Glitch and Underpowering

This combination has been studied targeting the ARM Cortex-M0 microcontroller and is based on the fact that the propagation delay of CMOS logic increases with a lower supply voltage. Combining underpowering with clock-glitch insertion should make the device more sensitive to fault injections. The ARM Cortex-M0 MCU has been chosen to prove this assumption because this MCU showed a low sensitivity on clock glitches applied alone. The supply voltage of the ARM Cortex-M0 is reduced to 1.2 V during the clock glitch, this value is beyond the minimum operating voltage of 1.8 V given in the datasheet [102]. It has to be mentioned that for the experiments, where underpowering is applied, the decoupling capacitors have been removed. This step was necessary

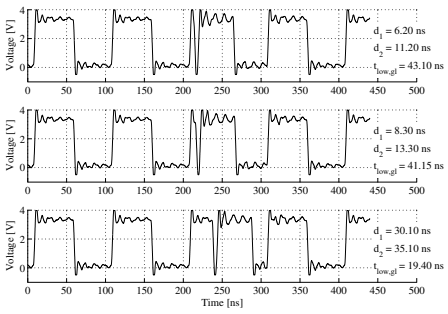


Figure 5.9: Different clock-glitch shapes for 10 MHz.

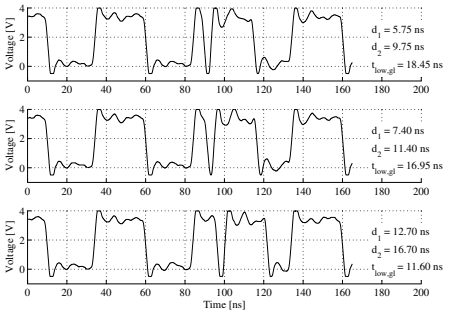


Figure 5.10: Different clock-glitch shapes for 20 MHz.

because due to the discharging of the capacitors, it is very hard to synchronize the underpowering and the clock glitch.

Clock Glitch and Temperature Variation

Preliminary experiments revealed that when combining clock glitches with temperature variation, all clock glitch parameters (d_1, d_2, and $t_{low,gl}$) influence the results. Figure 5.9 and Figure 5.10 show measured clock signals provided to the ATmega 162/v MCU for three different $[d_1, d_2]$ settings, once for a reference clock frequency of 10 MHz and once for a reference clock frequency of 20 MHz, respectively. These figures also show the relationship between d_1 and $t_{low,gl}$. The corresponding settings for $[d_1, d_2]$ are shown in the legends of these plots. Values for d_2 in the range of 7.0 ns and 50.0 ns were used for the setting of $f_{clk} = 10$ MHz. This equals glitch frequencies between 20 MHz and 142 MHz. For the setting of $f_{clk} = 20$ MHz, values for d_2 between 7.0 ns and 25.0 ns were used.

In the following, we describe the evaluation process for studying the influence of ambient temperature on clock-glitch fault injections in detail. The microcontroller program executed on the ATMEGA MCU is similar to the microcontroller program introduced in Section 5.3.1. All experiments are performed for ambient temperatures of 25 °C (room temperature) and 100 °C, respectively. Note that the maximum temperature rating is specified to 125 °C in the datasheet, so we do not operate the device beyond its specifications. The whole procedure was automated using a MATLAB® script in order to maximize the performance and minimize human interaction. The values given in Table 5.2 had to be defined before the script was started. Then, for each parameter set $[d_1, d_2, cmd]$, the following procedure is performed N times:

1. Configure the fault board with the current clock glitch parameters $[d_1, d_2]$.

2. Arm the clock glitch function on the fault board to insert the clock glitch with the defined shape after a trigger event.

3. Send the command *cmd* to the microcontroller.

4. Wait for the response of the microcontroller.

5. Compare the received register contents with the reference register contents Reg_{ref} of the reference execution without clock glitch and store the values if there are deviations.

5.4 Experiment Results

In this section the results of the experiments defined in Section 5.3 are presented. We start with the discussion of the fault-injection impact on two different microcontroller platforms followed by the discussion about the combination of fault-injection methods.

5.4.1 Fault-Injection Impact on Different Microcontroller Platforms

On both microcontrollers (ARM Cortex-M0 and ATxmega 256), all the investigated instructions are vulnerable to clock glitches. For the arithmetical/logical instructions, similar effects were observed on both microcontroller platforms. These effects were skipping or repeating of instructions and modifying the results of a calculation. One difference was that on the ARM Cortex-M0 MCU two instructions can be influenced in parallel because two instructions are fetched in parallel. Clock glitches targeting the branch instructions allow to prevent the branch from being taken. This result was achieved on both microcontrollers, only the parameters for the glitch were different. For the memory instructions, clock glitches allow loading constant values instead of the expected values and skipping of load/store instructions. On the ARM Cortex-M0 the value to be stored could also be modified. Summing up the achieved results it can be concluded that the impact of clock glitches on both microcontroller platforms is very similar. The main differences are the clock glitch parameters to cause specific faults. Figure 5.11 summarizes all the results in terms of d_2 values which allow to cause faulty behavior for the targeted instruction in the corresponding

Table 5.2: Parameter set for fault injection.

Parameter	Description
$d_{1,start}$	Start value for d_1
$d_{1,end}$	End value for d_1
$d_2 - d_1$	Shape of the inserted glitch (glitch duration)
Δd_1	Step size for increasing d_1
N	Number of repetitions for same glitch shape
cmd	Command defining the targeted instruction
Reg_{ref}	Reference register values for the current command
f_{clk}	Clock frequency for the microcontroller

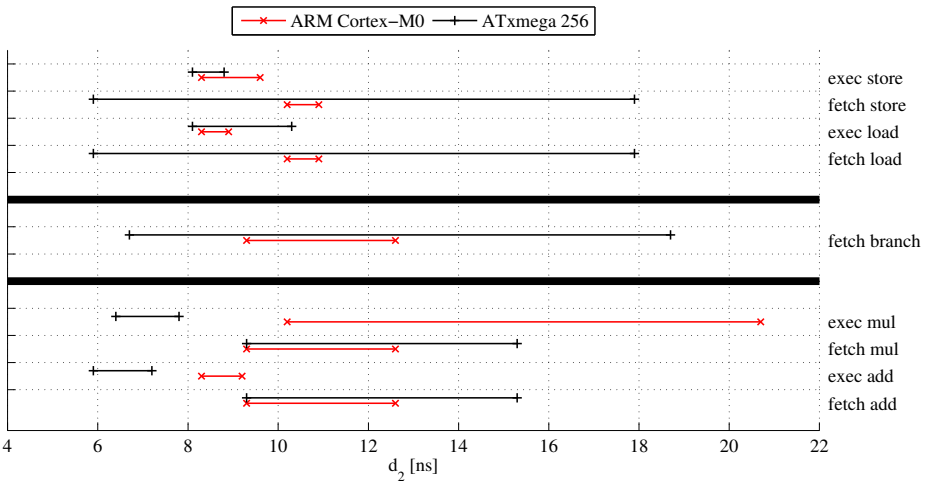


Figure 5.11: d_2 values leading to successful fault injections in the corresponding pipeline stages for the examined instructions.

pipeline stage. In the following paragraphs a detailed summary of all observed fault effects due to clock glitches is given.

ARM Cortex-M0, Arithmetical/Logical Instructions

When targeting the fetch stage of the `adds Rd,Rn`/`muls Rd,Rn`/`lsls Rd,Rn` instructions, d_2 values between 9.3 ns and 12.6 ns cause the fetch buffer not to be updated. As a consequence the previously fetched instructions stay in the fetch buffer and get executed again.

When targeting the execution stage of the `adds Rd,Rn` instruction, erroneous calculation results for d_2 values between 8.3 ns and 9.2 ns were observed. The erroneous value written to `Rd` depends on the currently used d_2 value as well as the terms of the sum stored in `Rd` and `Rn`. For the execution stage of the `muls Rd,Rn` instruction, erroneous calculation results for d_2 values between 10.2 ns and 20.7 ns were observed. The erroneous value written to `Rd` increases with the currently used d_2 value and is also dependent on the operands stored in `Rd` and `Rn`. Finally, when targeting the execution stage of the `lsls Rd,#imm` instruction, the value of the destination register is set to zero instead of the left-shifted value for d_2 values between 8.3 ns and 10.7 ns.

Targeting the decode stage of the arithmetical instructions did not yield to any faulty behavior.

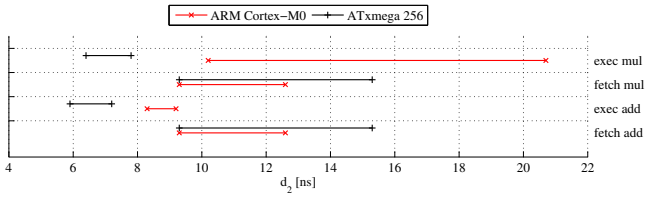
One important observation is that the d_2 intervals for affecting the fetch and execution stage do not completely overlap. This allows selecting the preferred pipeline stage in each clock cycle by carefully choosing the appropriate d_2 value.

ATxmega 256, Arithmetical/Logical Instructions

When targeting the fetch stage of the `add Rd,Rn` instruction, d_2 values between 5.9 ns and 17.9 ns lead to a non-execution of the addition. The assumption was that the previously fetched `nop` instruction stays in the fetch buffer, i.e. the fetch buffer is not updated due to the clock glitch (remember that the targeted `add` instruction is surrounded by several `nop` instructions). This assumption was verified by targeting the next fetch stage. Now the addition was executed twice for d_2 values between 9.3 ns and 15.3 ns. When targeting the fetch stage of the `mul Rd,Rn`, the same d_2 values lead to a non-execution of the multiplication. Compared to the addition, it was not possible to force a second execution of the multiplication by targeting the next fetch stage. The upper results show that clock glitches allow to prevent the fetch buffer from being updated. The required d_2 values to force this behavior depend on the current instruction located in the fetch buffer.

For inserting a clock glitch during the execution stage of the `add Rd,Rn` instruction, d_2 values in the range between 5.9 ns and 7.2 ns lead to a constant value in the destination register `Rd`. No relation between the constant value and other register values was observed. Further, the same value could be observed for all d_2 values in the interval [5.9 ns, 7.2 ns]. Executing the multiplication requires two clock cycles. A fault injection using clock glitches was only possible during the second execution cycle. Here, d_2 values in the interval [6.4 ns, 7.8 ns] led to erroneous values in the high byte of the multiplication result.

Comparison: Arithmetical/Logical Instructions



Attacks on arithmetical/logical instructions led to similar behavior on both MCU platforms. Due to the parallel fetch stage of the ARM Cortex-M0, two instruction fetches can be influenced with a single clock glitch. Furthermore, the d_2 values causing erroneous behavior vary for the two platforms.

Skipping an instruction allows an attacker to output an internal state before a last modification, e.g. the AES state before the last key addition. With a pair of ciphertexts, once without and once with skipped last key addition, it is possible to calculate bytes of the last round key. Repeating one instruction can render several operations ineffective. If e.g., an internal value a is XOR'ed with a key byte k twice, this results in the original value a again ($a \oplus k \oplus k = a$). Setting the result of an arithmetical operation to a constant, known value (when attacking the execution stage) also poses a serious threat for attacks on cryptographic primitives implemented in software.

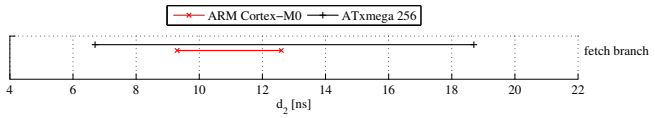
ARM Cortex-M0, Branch Instruction

When targeting the fetch stage of the `beq` *`label`* instruction following a `cmp Ra,Rb` instruction it was possible to prevent the branch from being taken independent of the result of the previous comparison. d_2 values between 9.3 ns and 12.6 ns led to this behavior. Note that the results are exactly the same as for the evaluation targeting the fetch stage of the `adds Rd,Rn` instruction.

ATxmega 256, Branch Instruction

Clock glitches during the fetch stage with a length of d_2 between 6.7 ns and 18.2 ns allowed to skip the branch instruction `breq` *`label`* independent of the result of the previously executed comparison `cp Ra,Rb`. This behavior can be explained by using the results of the attacks on the fetch stage of the arithmetical/logical instructions. The fetch buffer is not updated, the previously fetched instruction is executed again (in the current case the comparison).

Comparison: Branch Instruction



With similar effects, both MCU platforms were vulnerable to attacks in the fetch stage of *branch instructions*. The only difference when obtaining this behavior between the ARM Cortex-M0 and the ATxmega 256 was the glitch period d_2.

Preventing a branch from being executed enables, e.g. the option of skipping security-relevant code segments, loop iterations or complete function calls. In this context, a manipulation of algorithms implemented in software dealing with sensitive data might lead to a leakage of valuable or secret information processed on an MCU. A particularly interesting fact is, that branches are usually executed directly after a compare instruction. Assuming that the compare instruction remains in the instruction fetch buffer, the compare instruction is executed twice instead of the branch instruction. Thus, it is possible to change the program flow without any undesirable side effects on data or other parts of the code.

ARM Cortex-M0, Memory Instructions

For evaluating memory instructions, `ldr Rd,[Rn]` (load data from SRAM address stored in `Rn`) and `str Rd,[Rn]` (write data to SRAM address stored in `Rn`) were investigated. Load and store instructions both require two execution cycles.

When targeting the fetch stage, d_2 values between 10.2 ns and 10.9 ns lead to wrong values written to and read from memory, respectively. In particular, the wrong value written or read was always zero.

The first observation when targeting the two execution stages of the memory instructions was that only the second execution stage is vulnerable to clock-glitch attacks performed with our setup. We observed that for d_2 values between 8.3 ns and 8.9 ns the SRAM address stored at `Rn` was loaded to `Rd` in the case of the load instruction. In the case of the store instruction, the register value at `Rn` instead of the register value at `Rd` was stored to the appropriate SRAM address. Furthermore, d_2 values between 9.0 ns and 9.6 ns during the second execution stage prevented the execution of the load and store operation. This observation allows to prevent any load/store operations from being executed in an implementation.

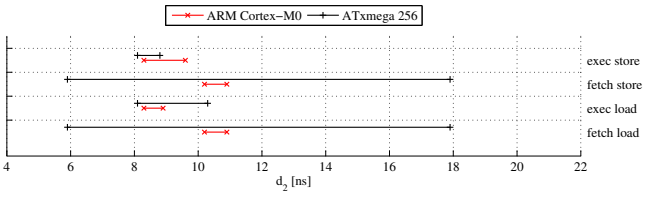
ATxmega 256, Memory Instructions

For the ATxmega 256 we target the memory instructions `ld Rd,X` and `st X,Rn` where `X` holds the 16 bit SRAM address as a combination of register `R27` and `R28`. It is notable that the `ld` instruction requires two execution cycles, whereas the execution of the `st` instruction is performed within one clock cycle.

When targeting the fetch stage, instructions `ld Rd,X` and `st X,Rn` were not executed when using a glitch period d_2 between 5.9 ns and 17.9 ns. This behavior can be explained due to the fact that the injected fault prevents the instruction fetch buffer from being updated as we already observed in our previous experiments concerning the fetch stage.

When `ld Rd,X` is attacked in the first execution cycle wrong data is written to the destination register `Rd`. In this context, no stable relations between the resulting wrong value of `Rd` and any value at a different memory location are given. The second execution cycle of `ld Rd,X` exposed to be resistant against clock glitch attacks using our setup. When attacking `st X,Rn` in its execution stage wrong data is written to the memory address, which is defined by the address pointer `X`. For both instructions, the erroneous behavior is achieved when using a clock glitch period d_2 between 8.1 ns and 8.8 ns. Increasing d_2 to a value between 9.6 ns and 10.3 ns, register `Rd` is set to zero in case of attacking `ld Rd,X` in the first execution cycle, while for `st X,Rn` no further effects were observed.

Comparison: Memory Instructions



Injecting clock glitches targeting memory instructions led for both MCU platforms to the following behavior: loading constant values or preventing a

load or store instruction from being executed. Additionally, constant values, namely zero or the memory address are stored instead of the desired register entry in case of the ARM Cortex-M0.

In this context, loading a known and constant value instead of true memory entries enables an attack scenario on algorithms which are based on retrieving secret information from memory. These attacks present a threat to cryptographic implementations where memory instructions are, e.g. used to load keys or initial values for random number generators. Again, influencing these values before they are applied to cryptographic primitives might establish access to sensitive data.

5.4.2 Combination of Fault Injection Methods

In this section the results of the experiments for combined fault injections are presented. When combining underpowering with clock glitches, for specific glitch shapes reproducibility rates of 100% for fault injections could be achieved. This was not possible with our setup when only applying clock glitches. When clock glitches are combined with high ambient temperature the parameters for injecting similar faults as for room temperature change. This appears because of the *temperature derating factor*. We could also observe that the number of parameters where faults occur becomes larger for the higher temperature. Also, faults which were not observed at room temperature appear at the high temperature. The following paragraphs provide a detailed discussion of the results.

Clock Glitch and Underpowering

Preliminary clock-glitch attacks targeting the ARM Cortex-M0 MCU figured out that this MCU type is quite insensitive to this kind of fault injection. Regardless of the used system clock frequency and the glitch period d_2, only a negligible number of attacks led to mainly non-reproducible results. Combining clock glitches and underpowering targeting the ARM Cortex-M0 MCU allows to increase the reproducibility of the injected faults and the interval for d_2 values allowing to inject faults. The supply voltage has therefore been decreased to 1.2 V during the clock glitch as depicted in Figure 5.12. For the evaluated instructions defined in Section 5.3.1 the combination of the fault injection methods allowed to achieve 100 % reproducibility rates for specific d_2 ranges, which was not possible when applying clock glitches alone. These d_2 ranges for the studied instructions are summarized in Figure 5.11.

Clock Glitch and Temperature Variation

For the discussion of the results the focus is put on the `add R16,R5` instruction. By means of clock glitches the following fault behavior was observed: The same instruction is repeated, a modified instruction is repeated, the result of the addition is corrupted, `mov` is executed instead of `add`, and inconsistent faults without a fixed pattern. The actual type of injected fault depends on the current

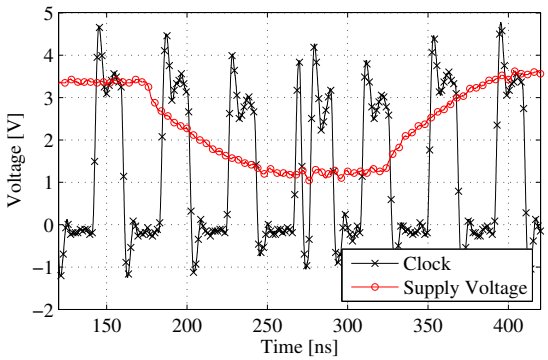


Figure 5.12: Clock signal and supply voltage during an attack ($d_2 = 10.0\,\text{ns}$, $U_{Glitch} = 1.2\,\text{V}$).

setting of the parameters d_1 and $(d_2 - d_1)$. The difference $(d_2 - d_1)$ describes the time interval the clock signal is low before the additionally inserted positive clock edge.

Figure 5.13 and Figure 5.14 depict the results of the experiments when the MCU is clocked with 10 MHz and 20 MHz, respectively. Results for ambient temperature of 25°C are marked with a circle (○) and results for 100°C are marked with a star (⋆). Parameter d_1 is plotted on the horizontal axis and the offset on the vertical axis corresponds to the type of fault. Four different $(d_2 - d_1)$ values have been considered. The following observations can be derived from the figures:

- There is no negative impact of increasing the temperature on the type of fault. The same faults can be injected for both investigated temperatures, room temperature and 100°C.
- By increasing the temperature, the clock-glitch injection time is shifted to the right. The reason for that relates to the *temperature derating factor* and is discussed later in this section.
- For specific clock-glitch shapes (e.g. $(d_2 - d_1) = 5\,\text{ns}$) the time interval where a specific fault occurs becomes larger for higher temperatures.
- For the setting $(d_2 - d_1) = 5\,\text{ns}$ some faults only occur when the device is running under high ambient temperature. Therefore, the success rate for a fault injection on that device gets higher with increasing temperature.
- When the device is clocked with 20 MHz, the high ambient temperature increases the variety of faults in the cases $(d_2 - d_1) = 6\,\text{ns}$ and $(d_2 - d_1) = 7\,\text{ns}$ respectively. Also the number of values for d_1, where fault injections are successful, increase.

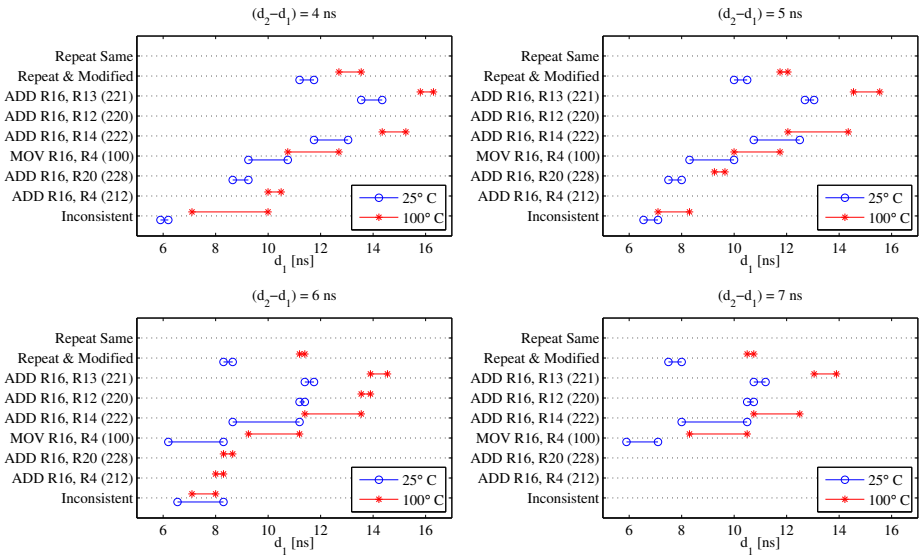


Figure 5.13: Types of faults generated targeting the instruction `add R16,R5` for ambient temperature and 100°C for different clock glitch settings ($f_{clk} = 10MHz$).

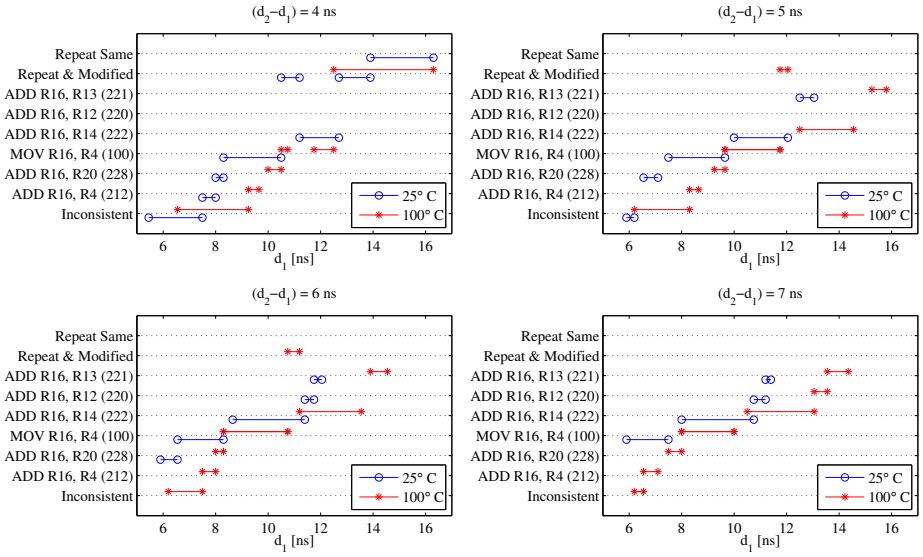


Figure 5.14: Types of faults generated targeting the instruction `add R16,R5` for ambient temperature and 100°C for different clock glitch settings ($f_{clk} = 20MHz$).

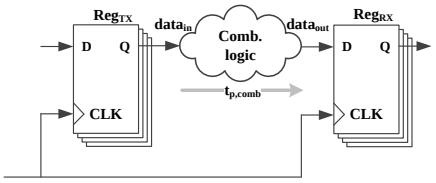


Figure 5.15: A simple synchronous circuit.

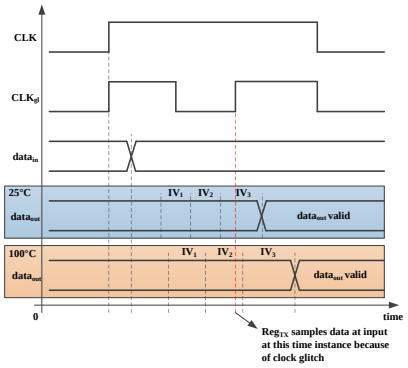


Figure 5.16: Timing diagram showing the influence of temperature on the result of a clock glitch insertion.

Temperature derating factor By analyzing Figures 5.13 and 5.14, it is clearly visible that the sensitivity window for inducing the faults is shifted to the right when the temperature is higher. We are now going to explain this effect with a simple example. Let us assume the simple synchronous circuit shown in Figure 5.15. The registers sample the data input D at the positive clock edge and the same clock signal CLK is provided to all registers. A combinational logic block is located between the transmitting registers Reg_{TX} and the receiving registers Reg_{RX}. This combinational logic block has a propagation delay $t_{p,comb}$. This propagation delay defines the time after which the output has settled to a stable value in the worst case after an input change. A proportional relationship between $t_{p,comb}$ and the junction temperature exists, i.e., the higher the junction temperature is, the longer is the propagation delay of a combinational circuit. In industry, the *derating factor* K_Θ is used to describe the influence of the temperature on the speed of a circuit. In order to fully describe the impact of PTV (process, temperature, and voltage) variation on the speed of a circuit, derating factors describing the process (K_P) as well as the supply voltage (K_V) also exist. The nominal timing is multiplied with the product of K_Θ, K_P, and K_V to get the timing for a specific condition. More detailed information about the derating factors can be found, for example, in Chapter 12 (p. 590) in the book *Digital Integrated Circuit Design* by H. Kaeslin [64].

Several intermediate values (IV_1, IV_2, IV_3, ...) appear at the output of the combinational logic block before it settles to the stable value. If the receiving registers sample their input before the combinational block provides a stable value (due to a too high clock frequency or the insertion of a clock glitch to perform a fault attack), this consequently leads to wrong results. The intermediate values, which can be observed at the output of the combinational logic block depend on the previous input value $data_{in}(t-1)$ and the new input value

$data_{in}(t)$. Each intermediate value can be observed for a specific time interval. With rising temperature, the speed of the combinational logic slows down as discussed above, so the temperature influences the signal-propagation time and therefore the fault-injection window when a glitch is effective or not. This fact is shown in the timing diagram in Figure 5.16. The proportional relationship between temperature and speed of the circuit increase the size of the signal-propagation intervals as well as shifts their position to the right. If a similar clock glitch is inserted at two different temperatures, the type of the fault is different if the receiving registers sample different intermediate values. This fact is also illustrated in Figure 5.16. By applying the modified clock signal CLK_{gl}, the receiving registers are forced to sample the output of the combinational logic block $data_{out}$ before it has settled to a stable value. For a temperature of $25°C$, IV_3 is sampled while for $100°C$, IV_2 is sampled.

5.5 Discussion

In this chapter, non-invasive fault injection methods have been investigated by targeting three microcontroller units.

First, the influence of clock glitches on two different microcontroller units have been investigated. Although they differ in their architecture and also apply different instruction execution pipelines, similar effects to clock glitches have been observed.

Second, the combination of fault injection methods in order to improve the success rate of the fault injections have been studied. The results show that the combination of underpowering and clock glitches significantly increase the reproducibility of the injected faults and increase the overall sensitivity of the attacked device to clock tampering. By combining clock glitches with heating, the number of different faults could be increased for specific clock-glitch shapes. Furthermore, a temperature-dependent shift of the parameters for successful fault injections was observed. This shift can be explained with the temperature derating factor.

For the investigated microcontrollers, our low-cost fault injection platform shows very good performance. But it has to be mentioned that this setup is not intended to attack high-performance microcontrollers supporting maximum clock frequencies beyond 170 MHz. This is because the maximum glitch frequency supported by the fault board equals 170 MHz. Also the underpowering approach is likely to be less efficient when targeting microcontroller platforms running at lower supply voltages like e.g. 1.2 V. This is because the low-power architecture also limits the voltage during underpowering.

6
Semi-Invasive Fault Attacks

The aim of this chapter is to present fault-injection setups for semi-invasive fault attacks. In a first step, we discuss the equipment required for performing optical fault injections using a laser beam. In a second step, we perform fault-injection attacks targeting two microcontrollers. The aim of the attacks is to study several parameters influencing the success of optical fault injections. The content of this chapter has been published in [72] and the contributions can be summarized as follows.

Contribution

- In this work we have improved an existing fault-injection setup. The main improvements compared to the existing setup are as follows. Increasing the laser output power by using specialized, pulsed laser diodes. Application of specialized lenses in order to minimize the loss due to reflection. Enabling rear-side fault injections by using laser diodes with 1 064 nm wavelength.
- Verification of the correct functionality of the setup by performing preliminary attacks.
- An identification of and discussion about the parameters which mainly influence the capability to inject optical faults: *laser output power, laser pulse length, and correct focus of the laser beam.*
- The application of the *light-induced voltage alteration (LIVA)* technique in order to identify laser-sensitive spots on the chip.

This chapter is organized as follows. In Section 6.1 an introduction to semi-invasive fault attacks is given followed by some preliminary information in Sec-

tion 6.2. Our low-cost fault injection environment is presented in Section 6.3. The conducted experiments are introduced in Section 6.4 followed by the results of the experiments given in Section 6.5. Section 6.6 terminates the chapter with a discussion.

6.1 Introduction

In contrast to non-invasive fault injections introduced in the previous chapter, semi-invasive fault injections require a modification of the attacked device. In case of optical fault injection, the package material of the targeted chip has to be removed in order to allow a direct illumination of the silicon. Due to its small spot size, the different available wavelengths, and the easily adjustable optical power output, lasers have become the preferred choice for optical fault injection. Several publications in the past have shown that especially content stored in SRAM or EEPROM is vulnerable to optical fault attacks, i.e. the content can be modified by illumination. Corrupting the data stored in non-volatile and/or volatile memory can further lead to an exposure of secret information. This threat was already uncovered in the late nineties by Anderson *et al.* [6] as well as Boneh *et al.* [26]. In [5] it is shown how to block the EEPROM write operation. Schmidt *et al.* [118] use ultraviolet (UV) irradiation in order to modify memory content. By applying a laser beam, Skorobogatov [128] shows how to inject more fine-grained faults. In fact he is able to flip single bits in registers. A laser beam with a certain power can also be used to disable the write operation on SRAM memory cells. By disabling the write operation the content of the cells always stays the same. This fact is used in order to perform optically enhanced power analysis attacks, introduced by Skorobogatov [127]. Optical fault attacks targeting a CRT-based RSA implementation are reported in the work of Schmidt *et al.* [117]. Memory write and erase operations are the target of the attacks presented in [121]. Summing up the related work it figures out that memory (volatile and non-volatile) of microcontrollers is a common target for optical fault attacks.

The previously mentioned attacks assume the availability of a working optical fault-injection environment. In the following sections the main parts which need to be included in such an environment are discussed. Therefore an existing setup is improved to enable next to front-side attacks also rear-side attacks. The application of new laser diodes and specialized lenses further improve the setup. Furthermore the important parameters *laser focus, laser pulse length, and laser output power* are examined in detail. Their influence on successful fault injections from the front side and the rear side is discussed. Experiments already documented in literature targeting two different microcontrollers, one PIC 16F84 and one ATmega 162/v, are repeated with our setup. The objective of repeating this experiments is to prove the proper function of the environment as well as to evaluate the influence of the selected parameters in an efficient way. The results of these experiments can also be used for evaluating other devices. An approach in order to find laser sensitive spots on the chip surface based on

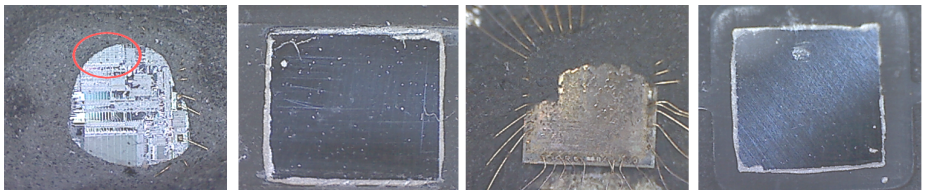


Figure 6.1: Exposed microcontrollers. From left to right: PIC 16F84 front side (circle indicates the location of the SRAM); PIC 16F84 rear side; ATmega 162/v front side; ATmega 162/v rear side.

light-induced voltage alteration (LIVA, [3]) is also presented and verified for the two microcontroller platforms. By identifying laser sensitive spots on the chip the efficiency of optical fault attacks can be increased significantly because the search space for efficient injection locations can be severely restricted. Even from the rear side, where no chip structure is visible, e.g. the location of the SRAM can be found. In addition, there is no need for modifying the fault-injection setup to perform further attacks taking advantage of the gathered location information.

6.2 Optical Fault-Injection Attacks

In this section general information about optical fault attacks is given. Optical fault attacks can be performed e.g. with flash lights or laser beams. The advantage of using a laser beam is that it can be focused to a small spot, so the targeted spot on the chip surface can be selected with high accuracy. In order to enable optical fault attacks the chip has to be exposed in a first step. Exposing means gaining access to the chip surface by removing the package. Two possibilities exist, opening the chip from the front side or from the rear side. Gaining access to the front side of the chip requires the usage of toxic acids (as presented in e.g., [126]) in order to remove the package material. This technique should only be applied by chemists in an adequate environment. Furthermore great care has to be taken not to damage the sensitive bonding wires. Rear side opening can be performed using a mill in order to remove the package and then removing the heat-sink metal plate e.g. with small pliers. As opening a chip from the rear side does not require any expensive or dangerous equipment it can be conducted with comparable small effort. Figure 6.1 depicts the exposed microcontroller chips we have used for our experiments. The two leftmost pictures correspond to the PIC 16F84 microcontroller and the two rightmost pictures correspond to the ATmega 162/v microcontroller. Potential targets of optical fault attacks on microcontrollers are the SRAM or the EEPROM memory. The reason for the sensitivity of these parts on laser irradiation is outlined in the following. Furthermore an introduction and explanation of the proposed method for finding optical sensitive spots on the chip surface is given in this section. This method is based on light-induced voltage alteration (LIVA, [3])

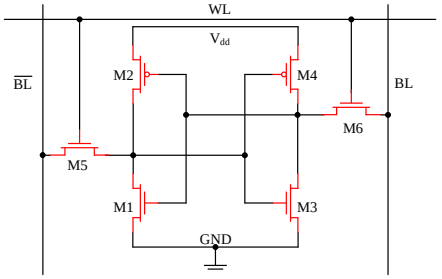


Figure 6.2: Architecture of an SRAM cell.

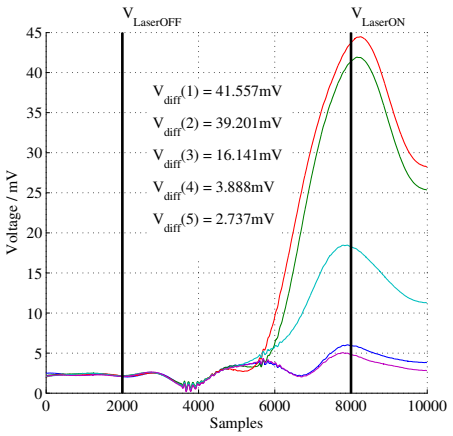


Figure 6.3: Voltage drop across the resistor in the ground line while irradiating different spots on the chip (laser active from 5 000 to 10 000 samples).

6.2.1 Influence of Light Irradiation on Memory Cells

A detailed description on how light irradiation can be used to manipulate EEPROM or Flash memory on microcontrollers as well as on dedicated storage chips can be found in [121]. As we do not target EEPROM nor Flash memory in this work no detailed description is given here.

The main target of the optical fault attacks carried out in this work is the SRAM of microcontrollers. A detailed description of the functionality of SRAM is following. SRAM is the short form for *static random-access memory* and it is a volatile memory. Volatile memories loose their stored value when the supply voltage is switched off. The architecture of an SRAM cell is depicted in Figure 6.2. The cell typically consists of six transistors overall where four transistors form a flip-flop (M1 ... M4) and the remaining two (M5, M6) are used for write and read access. When a cell stores a logical zero M2 and M3 are conducting and M1 and M4 are non-conducting. On the other hand, when the cell stores a logical one M2 and M3 are non-conducting and M1 and M4 are conducting. A laser beam with sufficient power targeting one specific transistor can force this transistor to change the state from non-conducting to conducting. If e.g., the targeted cell stores a logical one and the laser beam is focused on M3 the state of M3 changes from non-conducting to conducting if the power of the laser beam is sufficient. As a result the flip-flop changes the state then from logical one to logical zero. The same effect can also be achieved if a cell stores a logical zero and M1 is targeted with the laser beam. Then the stored value of the cell is flipped to logical zero. The newer the production technology of the attacked chip is, the more difficult is it to target a single transistor with the laser

beam. The size of the transistors decreases on the one hand but the minimum size of the laser spot is limited on the other hand. So newer technologies make it harder to induce predictable faults as the laser beam influences several transistors at the same time.

6.2.2 Finding Optical Sensitive Spots

One of the first steps (after the decapsulation step) when launching an optical fault attack is to identify the area of interest. If the attack e.g. targets intermediate values during a computation stored in SRAM, the area of interest equals the location of the SRAM. For older microcontrollers (e.g., PIC 16F84) this is a quite easy task if the front side of the chip is opened. Here the location of the SRAM can be found by visual inspection with a microscope. Things change when the attack is performed from the rear side or when a microcontroller produced with a newer fabrication technology is attacked. From the rear side no structure is visible with the human eye using a microscope. Chips produced with a newer fabrication technology also have several metal layers on top. This metal layers make front-side attacks very difficult because the metal layers shield the underlying transistors from the laser irradiation. In the following a method is presented which should assist in finding such sensitive spots. It is based on "light-induced voltage alteration" (LIVA), introduced in [3] and can be applied for front-side attacks and rear-side attacks. In order to conduct the method only the power (V_{dd}) and ground (GND) pins of the analyzed chip need to be connected to a power supply. The chip has to be powered with a constant voltage and a resistor has to be added into the ground line like in a standard power-analysis scenario. The voltage drop across the resistor is measured using an oscilloscope. Several spots on the chip surface are illuminated by short laser pulses and the voltage drop across the resistor is measured shortly before and during the laser pulse. Each recorded trace is stored together with some location information corresponding to the illuminated spot on the chip. In an analysis step the difference of the voltage values during the laser pulse ($V_{LaserON}$) and before the laser pulse ($V_{LaserOFF}$) for every spot is calculated: $V_{diff} = V_{LaserON} - V_{LaserOFF}$. If the analyzed spot is not sensitive to laser irradiation $V_{diff} \approx 0$. On the other hand $V_{diff} > 0$ if the spot is sensitive to laser irradiation. Figure 6.3 shows the recorded traces for five different spots. The laser was active between 5 000 and 10 000 samples. Two spots are highly sensitive to the laser irradiation ($V_{diff}(1), V_{diff}(2)$), one is medium sensitive ($V_{diff}(3)$) and two spots are less sensitive ($V_{diff}(4), V_{diff}(5)$). Combining the V_{diff} values and the corresponding location information allows to create a sensitivity plot of the chip.

6.3 Fault Injection Environment

In this section the optical fault-injection environment used to perform the optical fault attacks is presented. This optical fault-injection environment can be split into five main components which are discussed one after the other in the

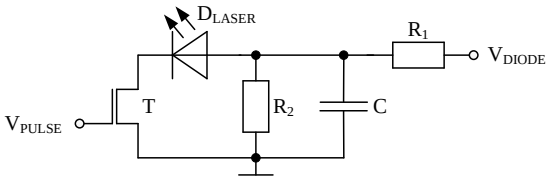


Figure 6.4: Schematic of the laser diode driver circuit.

following: the optical equipment, the stepper table, the oscilloscope, the control computer, and the custom-made fault board. Oscilloscope, control computer, and stepper table are standard equipment without the need for any modification, so they will only be introduced shortly. These devices were already applied in the existing fault-injection setup. For the optical equipment and the custom-made fault board, several modifications were necessary to improve the existing fault-injection setup. These modifications are discussed in the corresponding sections. Figure 6.5 depicts the interaction between all the mentioned components. In the end of this section also some information about the devices under test (DUT) used for the experiments, namely the PIC 16F84 and the ATmega 162/v microcontrollers, can be found.

6.3.1 Optical Equipment

In order to enable front-side attacks as well as rear-side attacks different types of laser diodes have to be used. The existing setup which only allowed front-side attacks used a laser diode with a wavelength of 780 nm and an optical output power of 90 mW cw (continuous wave) and 200 mW pulsed (max. pulse length: 500 ns, duty cycle: 50 %) respectively. In order to increase the pool of attackable devices using front-side attacks the existing laser diode was exchanged by a laser diode providing a higher output power. The new laser diode has a wavelength of 808 nm and a maximum optical output power of 16 W. This diode can only be operated in pulsed mode with a maximum pulse length of 200 ns and a duty cycle of 0.1 %. For rear-side attacks a laser diode with a wave length of 1 064 nm and a maximum optical output power of 16 W was used. Also this diode can only be operated in pulsed mode with a maximum pulse length of 200 ns and a duty cycle of 0.1 %. In order to generate the high-energy pulses for the pulsed laser diodes, it was necessary to create a laser-diode driver. Figure 6.4 depicts the schematic of this circuit. A capacitor is charged to the maximum voltage of the laser diode. A field-effect transistor (FET) used as a switch allows to discharge the capacitor across the laser diode producing the optical laser pulse. Also the laser mount has been modified in order to allow easy access and replacement of the laser diode. This laser mount integrates a collimation lens in order to collimate the laser beam. The distance between laser source and collimation lens can be precisely adjusted with a screw. This adjustment allows to exactly set the diameter of the collimated laser beam. Two different collimation lenses had to be used, one for the 780/808 nm diodes and the other one for the 1 064 nm diode.

Each lens has a high transmission factor for the specific wavelength range. So a well-collimated laser beam as well as low losses due to reflection are guaranteed. The laser mount is attached on a fixture of the microscope, originally designed to mount a camera. The collimated laser beam is focused with a standard 50 x lens for the 780 nm and 808 nm wavelength diodes and with a 20 x lens for the 1064 nm wavelength diode. The 20 x lens has an improved transmission factor for wavelengths in the range of 1 064 nm. Experiments showed that the lens used for the front-side attacks does not work for rear-side attacks due to the higher wavelength of the laser source required for the rear side.

6.3.2 Stepper Table

In order to enable automated scanning of the chip surface a stepper table which can be controlled via MATLAB® commands is used. The stepper table can be moved in x, y and z direction with an accuracy of 0.05 μm. With the x and y coordinate the spot where the laser beam hits the chip is selected and with the z coordinate the focus of the laser beam is set. Our results show that the z coordinate is a crucial factor when performing optical fault attacks. For the remainder of this work, whenever a variation of the z coordinate is mentioned, this corresponds to setting the correct focus of the laser beam.

6.3.3 Oscilloscope and Control Computer

A *'LeCroy WP 725 Zi'* oscilloscope is used in order to record the required signals. These signals involve trigger signals from the device under test for the laser pulse on the one hand and power traces on the other hand. Power traces were recorded in order to measure the influence of the laser beam on the power consumption of the device. A sampling rate of 10 GS/s was used in order to record the power traces and the build-in 20 MHz lowpass filter was enabled to cut off high-frequency noise.

The *'LeCroy WP 725 Zi'* oscilloscope is also capable of running MATLAB® scripts and so it was also used as the control computer. Communication with the device under test, the custom-made fault board as well as with the stepper table can therefore be realized with a single device.

6.3.4 Custom-made Fault Board

In order to generate laser pulses of exact length and at exact time instances, the same custom-made fault board as presented in Section 5.2 has been used. The laser pulses can be triggered by two events. The first event for triggering a laser pulse is a MATLAB® command (*shoot_laser_man*). Using this command, the laser pulse is triggered manually. The second event is an external trigger event occurring at the trigger input pin of the fault board (e.g., by applying a trigger pin on the device under test). The length of the laser pulse as well as the delay after the trigger event (for the latter case) can be configured. The clock signal and the power supply for the DUT are also provided by the fault board.

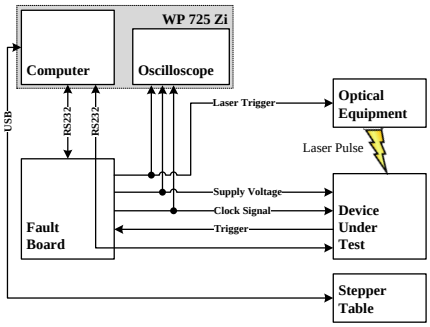


Figure 6.5: Block diagram of the setup for optical fault injections.

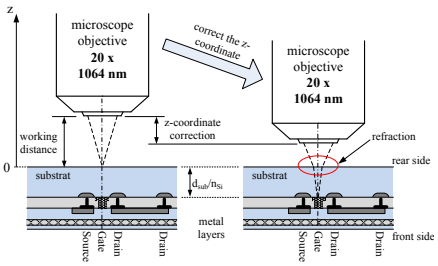


Figure 6.6: Influence of the z coordinate on the laser focus.

In order to pause the DUT for a specific time interval, the fault board provides the functionality to stop the clock signal manually.

6.3.5 Devices Under Test (DUT)

For the performed experiments two different 8 bit microcontrollers were used, one PIC 16F84 [92] (1.2 μm process technology) and one ATmega 162/v [9] (0.35 μm process technology). Several optical fault attacks on the PIC 16F84 microcontroller have been reported in literature so far (e.g. [121], [128]) mainly targeting the SRAM. We were able to repeat the reported experiments successfully with our optical fault-injection environment and these results serve as the base for our work. For the ATmega 162/v microcontroller on the other hand hardly any reported practical optical fault attacks could be found in literature. We used this type of microcontroller to confirm that the results achieved with the PIC 16F84 can be transformed on other microcontroller platforms.

6.4 Experiment Description

In this section we briefly introduce the performed optical fault-injection experiments. Due to several differences, we split the discussion into front-side experiments and rear-side experiments.

6.4.1 Front-side Experiments

As preliminary experiment, optical fault injection attacks targeting the SRAM of the PIC 16F84 microcontroller have been performed. The location of the SRAM can be found by visual inspection with the microscope, this makes a time-consuming stepping over the whole die area unnecessary. The distance between lens and chip die (z-coordinate) was set according to the working distance of the lens. The working distance of the applied 50 x lens is 9 mm. Additionally, the z-coordinate was varied in a range of 70 μm in every location to find the best

focus for the laser beam. For detecting faults, the SRAM of the microcontroller was initialized with known values in a first step. Setting a general-purpose IO pin on the microcontroller signalizes the end of the initialization and this event triggers a laser pulse with a length of 200 ns. After the laser irradiation, the SRAM content was verified and wrong values were reported. This procedure was repeated on different locations on the SRAM by modifying the x coordinate and the y coordinate. In addition, at every location also the z coordinate was modified in a range of 70 μm. There are two main reasons why a variation of the z coordinate is performed. First, it is hardly possible to precisely set the correct working distance which directly influences the focus of the laser beam. Second, it cannot be guaranteed that the chip die is perfectly aligned horizontally. This fact makes it impossible to have a well-focused laser beam in every location without z-coordinate variation.

Based on the preliminary experiment, the influence of the parameters *laser pulse length, laser power, and z-coordinate* on the ability to inject a fault has been studied. Therefore, one location where a fault could be injected was selected and the three parameters were varied one after the other leading to two sets of parameters, depending if a fault was detected or not. In order to approximate the laser power $P_{diode,el}$, the voltage drop across the laser diode, U_{diode} and the current through the laser diode I_{diode} was measured to calculate $P_{diode,el} = U_{diode} \cdot I_{diode}$. Due to several losses it can be assumed that $P_{diode,el} > P_{diode,opt}$, with $P_{diode,opt}$ being the optical laser output power.

Next the LIVA approach was applied in order to find laser-sensitive spots on the PIC 16F84 and the ATmega 162/v respectively. Therefore, only the V_{dd} and GND pins of the microcontroller were connected to a power supply. A 47 Ω resistor was placed into the ground line to measure the voltage drop with an oscilloscope. A predefined area on the chip surface was scanned by modifying the x and y coordinates and the values of the voltage drop across the resistor shortly before and during the laser illumination were recorded and stored together with the location information.

6.4.2 Rear-side Experiments

Compared to the front-side experiments, we have switched the order of the experiments when targeting the rear-side of the chips. First, laser sensitive locations were identified. Based on the location information, fault-injection attacks targeting the SRAM were performed and afterwards the parameter influence was investigated. It was also necessary to change the laser diode to the model with 1 064 nm wavelength and the lens (20 x, optimized transmission for 1 064 nm).

Finding laser-sensitive locations on the PIC 16F84 served as the preliminary experiment. The distance between lens and the chip die was set to approximately 6 mm, what equals the specified working distance of the lens. Due to the optimization for 1 064 nm wavelength, no visible picture for the human eye can be created with this lens. A focused picture is an indicator for the correct working distance, so this approach was not applicable for setting the correct working distance. This fact, the varying thickness of the substrate, and the small horizontal

misalignment of the die makes a variation of the z coordinate in every location necessary. Figure 6.6 depicts the influence of the substrate thickness on the z coordinate. The transistor, which is located a specific distance d_{sub} below the substrate, poses the main target for the laser beam. So the initial set working distance has to be decreased by d_{sub}/n_{Si} by modifying the z coordinate. n_{Si} equals the refractive index of silicon ($n_{Si} \approx 3.5$, [84]).

By using the location information of the laser-sensitive spots, fault injections targeting the SRAM were performed again. Therefore the same approach as for the front-side experiments was applied. The length of the laser pulse was set to 200 ns and occurring faults were recorded.

In a further experiment, the influence of the parameters *laser pulse length, laser power, and z-coordinate* on the ability to inject a fault from the rear-side has been studied. For this investigation, one position where a fault injection was successful according to the previous experiment, has been selected.

6.5 Experiment Results

In the current section the results of the front-side and rear-side experiments introduced in Section 6.4 are presented. Fault injections targeting the SRAM of the PIC 16F84 microcontroller as already documented in [128], were performed as preliminary experiments. The success of these preliminary experiments confirmed the correct functionality of the fault-injection setup.

6.5.1 Front-side Experiments

Investigating one location where a fault injection is successful allows to discuss the influence of the parameters *z coordinate, laser power, and laser pulse length* on the ability to induce a fault. The result of this experiment is depicted in Figure 6.7. Light gray points indicate that the fault for the current setting (*pulse length, laser power, z coordinate*) could be injected and dark gray means that no fault could be injected. The *electrical laser power* has been increased in five steps (150 mW, 170 mW, 190 mW, 200 mW, 220 mW) and four different *pulse length values* have been examined (100 ns, 200 ns, 500 ns, 1000 ns). 150 mW *electrical laser power* is not sufficient in order to induce a fault even with the largest *pulse length* of 1000 ns. Result achieved with higher laser power can be interpreted as follows. Increasing the *laser output power* as well as increasing the *laser pulse length* make the setup less sensitive on the *z coordinate*. The z coordinate is directly related to the focus of the laser beam. That means, a well focused laser beam requires less energy for a successful fault injection. This observation is important if the laser diode is already operated at its limits in terms of power and pulse length. Additionally, by using the minimum required energy for a successful fault injection in combination with a well-focused laser beam the number of affected transistors can be minimized. For fine-grained fault injections minimizing the number of affected transistors by the laser beam is a desirable goal.

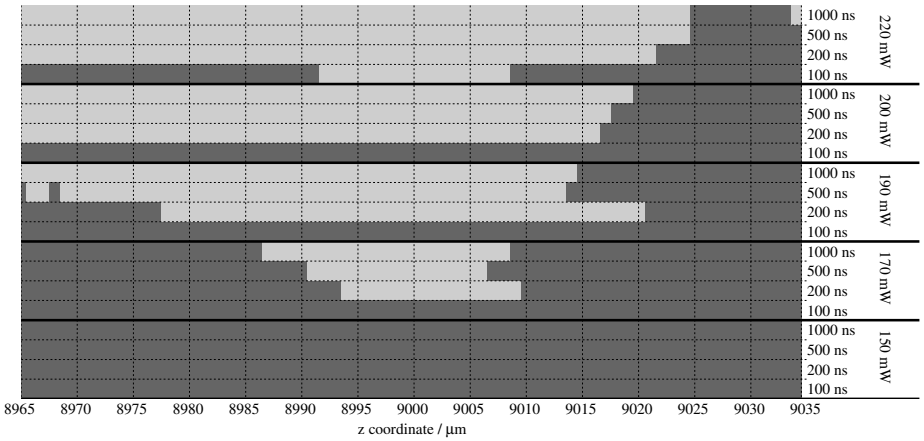


Figure 6.7: Influence of the parameters *laser output power, laser pulse length, and z coordinate* on the fault-injection success, PIC 16F84 front side (light-gray: success, dark-gray: no success).

Next the results of the application of the method to find laser sensitive spots on the chip is presented. Figure 6.8 depicts the result of a scan of one part of the SRAM area of the PIC 16F84 microcontroller. Laser-sensitive spots are bright while spots which did not show sensitivity on laser irradiation are dark. The bright locations correlate with the locations where fault injections were successful in the preliminary experiment. The plot shows two lines of 8 bit values of the SRAM. Figure 6.9 depicts the result of the evaluation focusing on the influence of the z coordinate on V_{diff}. The spot with a high sensitivity with the coordinates -0.10 / -0.21 (marked with the circle in Figure 6.8) has been selected for that purpose and V_{diff} has been measured for different z coordinates. The z coordinate (equals the distance between lens and chip surface, i.e. the focus of the laser beam) has been modified in the range of $9000\,\mu\text{m} \pm 35\,\mu\text{m}$ with a step size of $1\,\mu$m. $9000\,\mu$m equals the specified working distance of the used lens. As a result of this evaluation the optimal distance for conducting optical fault attacks is between $8980\,\mu$m and $8990\,\mu$m because the maximum V_{diff} values are achieved in that interval.

Figure 6.13 depicts the result of a scan of a randomly chosen area of the ATmega 162/v microcontroller. Only a small number of laser-sensitive spots in the upper part of the plot can be observed. Further analyses of these spots did not yield any results, i.e. laser irradiation targeting these spots during the execution of a program did not produce any faults. The reasons could be the higher density and number of metal layers on top as well as the smaller process technology compared to the PIC 16F84 microcontroller. Rear-side attacks seemed to be more promising so no further evaluations targeting the front side of the ATmega 162/v have been performed.

With the PIC 16F84 microcontroller it could further be proven that an ar-

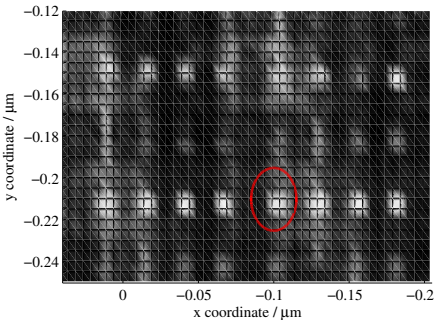


Figure 6.8: Laser sensitivity scan (PIC 16F84 front side).

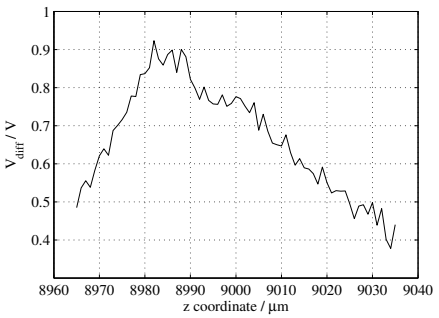


Figure 6.9: Z-coordinate influence on V_{diff} (PIC 16F84 front side).

bitrary number of faults can be induced within a single clock cycle. Therefore the clock-stop feature of the fault board was used. The clock was stopped after the initialization of the SRAM was finished. Stopping the clock allows to target several spots on the chip by moving the stepper table. Each spot was illuminated with a laser pulse. Then the clock was started again and the verification routine reported several faults. This approach equals an attack using N laser beams in parallel, where N is the number of faults which should be induced at the same time. The higher N is the more difficult is the attack with N laser beams. Each beam needs to be put in place with a high accuracy. The size of the lenses focusing the beams limits the minimum distance between two spots. These problems can be solved by stopping the clock and relocating a single laser beam. Prerequisites for this approach are that the attacked device supports an external clock and does not detect the corruption of the clock. The clock-stop approach can also be applied if two consecutive values in the same register need to be manipulated but the cool-down time of the diode is too long. This cool-down time equals an interval between two consecutive laser pulses which must not be undercut. If undercut, it can lead to a damage of the laser diode. Here the clock can be stopped or slowed down to allow the laser to cool down. One attack targeting consecutive register values is presented by Trichina *et al.* [141]. The authors target a protected CRT-RSA implementation and apply a multi-fault laser attack.

6.5.2 Rear-side Experiments

The first experiment targeting the rear side was finding laser sensitive spots on the PIC 16F84 microcontroller. The main focus of the experiments is put on the SRAM so the goal was to create a rear-side sensibility plot of the SRAM. The result of this experiment is depicted in Figure 6.10. The location of the SRAM on the front-side opened chip is known. This information was used in order to decrease the search area on the rear-side opened chip. The area inside the box

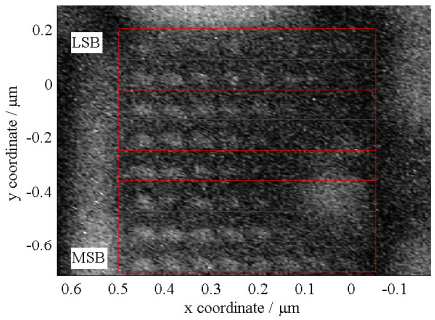


Figure 6.10: PIC 16F84 rear side laser sensitivity plot.

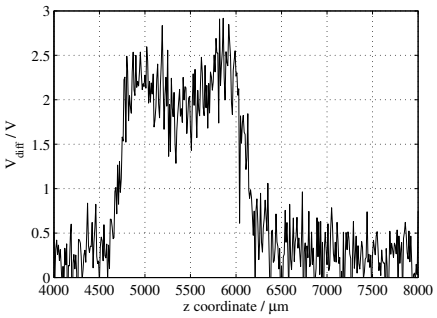


Figure 6.11: Z-coordinate influence on V_{diff} (PIC 16F84 rear side).

in the figure equals the whole SRAM of the PIC 16F84 microcontroller. Due to the fact that we used a 20 x lens for the rear-side evaluations compared to a 50 x lens for the front-side evaluations the achieved resolution compared to the front-side evaluations is significantly lower. In order to show the influence of the z coordinate for rear-side evaluations we have chosen one spot with a high sensitivity on the laser irradiation. At this spot the z coordinate was modified in the range of $6000\,\mu\text{m} \pm 2000\,\mu\text{m}$ with a step size of $10\,\mu\text{m}$. For every step V_{diff} was calculated and the result is plotted in Figure 6.11. The result is comparable to the result achieved during the front-side evaluation (Figure 6.9). Decreasing the distance between lens and chip surface until a specific value is reached increases the sensibility to laser irradiation.

In the second experiment we used the location information of the laser sensitive spots from the previous experiment. Each spot with a V_{diff} value above a given threshold was illuminated with a laser pulse with a length of 200 ns. The same algorithm as for the corresponding front-side experiment was executed on the PIC 16F84. The analysis of the results showed that with the given setup it is possible to influence each single bit of every SRAM register similar to the corresponding front-side experiment. Targeting the lower part of the SRAM area influences the *most significant bit (MSB)* while targeting the upper part influences the *least significant bit (LSB)*. Some spots lead to bit faults in more than one register. This is due to the fact that the achievable spot size with the 20 x lens is limited. So the laser spot on the chip influences neighboring transistors at the same time.

Figure 6.12 depicts the result of the investigation of the parameters *laser pulse length, laser power and z coordinate* on the fault-injection success. Light-gray parts indicate that for the corresponding parameters a fault injection is successful. The following *laser pulse lengths* were verified: 75 ns, 100 ns, 125 ns, 150 ns, 175 ns and 200 ns. In order to give values for the *laser power* we used the electrical power ($P_{diode,el} = U_{diode} \cdot I_{diode}$) and the following *power values*

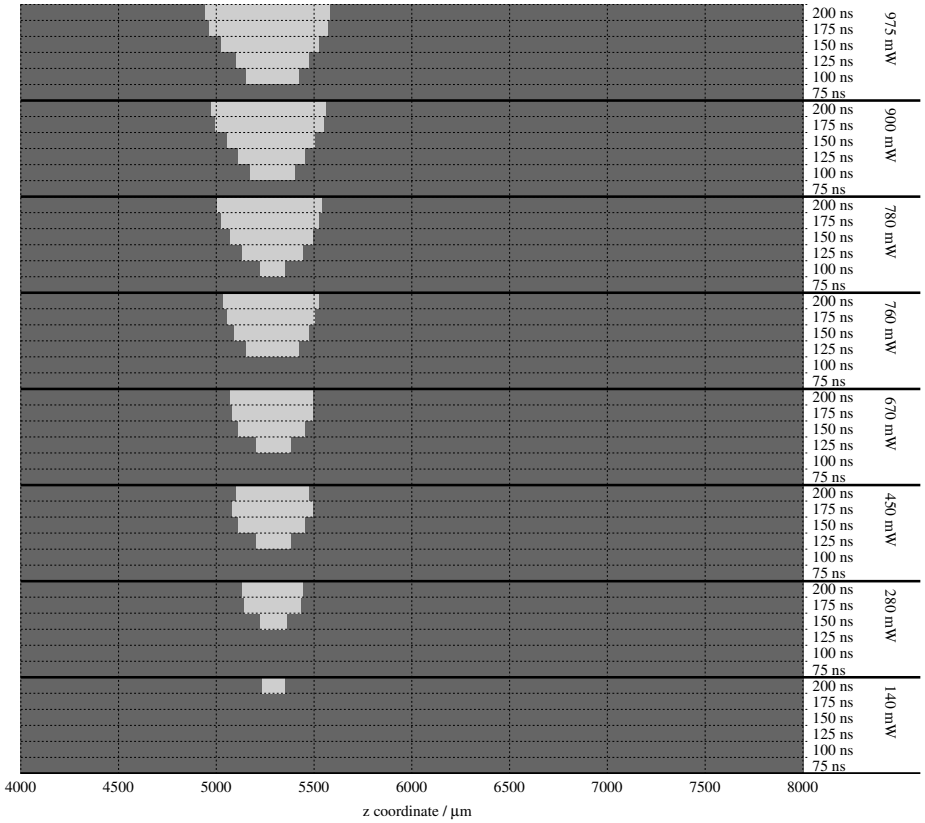


Figure 6.12: Influence of the parameters *laser output power, laser pulse length, and z coordinate* on the fault-injection success, PIC 16F84 rear side (light-gray: success, dark-gray: no success).

were used for the experiment: 140 mW, 280 mW, 450 mW, 670 mW, 760 mW, 780 mW, 900 mW, 975 mW. For each combination of *laser power and pulse length* the *z coordinate* was varied in the range of $6000\,\mu\text{m} \pm 2000\,\mu\text{m}$ with a step size of $10\,\mu$m. It can be observed that a minimum *pulse length* of 100 ns is required in order to induce a fault. Similar to the front-side experiments, the influence of the z coordinate decreases with increasing laser pulse length and laser power. Again a well-focused laser beam as well as the minimum laser energy allows to minimize the number of affected transistors by the laser beam. Another observation is that the *z coordinate* where faults can be induced is significantly smaller than the working distance of the lens. That is due to the fact depicted in Figure 6.6 and correlates well with the result depicted in Figure 6.11.

Figure 6.14 depicts the result of a sensitivity scan of the whole chip area of the ATmega 162/v microcontroller from the rear side. The size of the whole scan area is 3 mm x 3 mm and the step size was $30\,\mu$m. Brighter spots indicate a

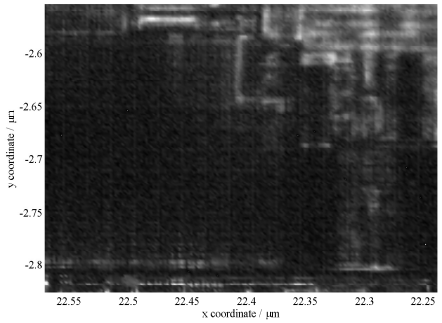


Figure 6.13: ATmega 162/v front side laser sensitivity plot.

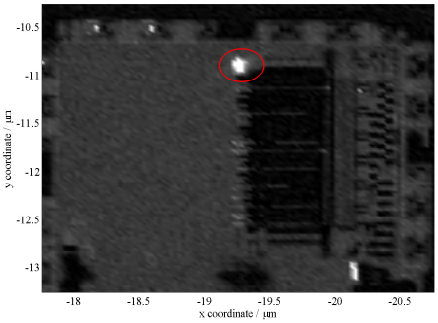


Figure 6.14: ATmega 162/v rear side laser sensitivity plot.

higher sensitivity to laser irradiation and darker spots are less sensitive on laser irradiation. The area inside the circle is most sensitive to laser irradiation so we examined this area in further experiments.

For the investigation of the area marked with the circle a similar SRAM-verification algorithm like for the PIC 16F84 was executed on the DUT. After the initialization of the SRAM the spots inside the mentioned area were irradiated with laser pulses with a length of 200 ns. For several spots inside this area the RS 232 communication between control computer and DUT stopped working properly and a reset of the DUT was required. One explanation for that behavior is that the SRAM is located in this area and the laser pulse influences several registers. Some of the influenced registers might hold relevant information required for the RS 232 communication between control computer and DUT. By modifying these values no proper communication can be achieved any more. One of these values could e.g. be a timer reload value which is required to achieve the correct baud rate. If this value is corrupted the communication does not work properly any more. Investigations of sensitive spots outside of the circle did not produce any faulty behavior. This results shows that the method for finding laser sensitive spots on the chip works also for the ATmega 162/v microcontroller very well. The main limiting factor is the 20 x lens that limits the minimum size of the laser spot.

6.6 Discussion

In this chapter several parameters which influence the success of optical fault attacks, have been examined in detail.

In the first part the required components for a fault-injection setup using laser diodes are discussed. We further present our fault-injection setup consisting of the following parts: microscope with laser mount, lenses, custom-made fault board, stepper table, control computer, oscilloscope, and laser diodes with different wavelengths. Different wavelengths are required for injecting faults from

the front side (780 nm and 808 nm) as well as from the rear side (1 064 nm) of the chip. The fault-injection setup is assembled with off-the-shelf parts accessible without any restrictions and at a fairly low price of a few thousand dollars.

Preliminary experiments targeting a PIC 16F84 microcontroller were performed to prove the correct functionality of the setup. Results of these preliminary experiments served as basis for the verification of the parameters influencing the success of optical fault injections. These examined parameters are *the laser power, the laser pulse length and the focus of the laser spot (z coordinate).* As the results show, by increasing the laser power and the laser pulse length one can achieve a higher probability for successfully injecting a fault. Nevertheless, both aforementioned parameters are limited by e.g. the type of used laser diode. The laser diodes used for our experiments, e.g. allow a maximal pulse length of 200 ns. Due to this limitation, the third parameter, the correct focus of the laser beam becomes crucial. The working distance of the applied microscope lens serves as a good starting point, but it figured out that for improving the fault-injection success, some fine-tuning by varying the z coordinate is necessary.

Furthermore, our setup allows to create so-called laser-sensitivity plots of the chip surface. One can make use of these plots to find laser sensitive spots on the chip surface. The approach of light-induced voltage alteration (LIVA) has been applied therefore. Laser-sensitivity plots are well-suited for decreasing the effort for fault-injection attacks targeting a specific implementation or algorithm. Plots for the PIC 16F84 microcontroller and the ATmega 162/v microcontroller from both, front and rear side, have been generated but the approach is applicable for any device.

The success of optical fault injections not only depends on the used equipment but also on the properties of the attacked device. Here the manufacturing process is the main factor. The smaller the manufacturing process, the harder it is to inject precise and reproducible faults. The manufacturing process of the PIC 16F84 is 1.2 μm and the manufacturing process of the ATmega 162/v microcontroller is 0.35 μm. For both devices we were able to create laser sensitivity plots with a high resolution, reproducible faults could only be produced targeting the PIC 16F84 microcontroller. Reasons are the bigger feature size of the PIC 16F84 microcontroller and also the limited capabilities of our fault injection setup. This let us come to the conclusion that precise and reproducible optical fault injections targeting devices below 0.35 μm require a more-sophisticated setup which might exceed our defined cost limit.

7 Active Relay Attacks

In this chapter we present relay attacks targeting contactless systems based on the RFID and NFC technology. The aim of these relay attacks is not to reveal key information or other secret data but to increase the communication distance between reader and tag for malicious purposes. This setup allows to initiate a communication between reader and tag although the tag is not in close proximity of the reader. We show that by applying two off-the-shelf, NFC-enabled smart phones a relay attack can be mounted without the requirement of a public network. We use the bluetooth communication channel or an ad-hoc wireless network as relay channel. During the practical experiments we uncovered several limitations introduced by the smart phones. By replacing one smart phone with a custom-made proxy device most of these limitations can be eliminated. The conducted experiments clearly show that a wide range of contactless systems is vulnerable to relay attacks that only apply low-cost, off-the-shelf equipment. Results presented in this chapter have been presented at IEEE-RFID 2014 [76] and the contributions can be summarized as follows.

Contribution

- We first pinpoint the advantages and disadvantages of NFC-based relay proxies compared to custom-made hardware. Custom proxies solve many relay-attack restrictions, e.g., cloning of the victim's UID, adaption of low-level ISO/IEC protocol parameters, direct request for Waiting Time Extensions, or modifications in the lower-level RFID protocols.

- We first present "three-phones-in-the-middle" attacks where we use one NFC phone to act as an access point for two other phones. Using this

setup, we demonstrate a successful relay attack over a distance of more than 110 meters.

- We compare the most relevant relay channels used for smart phones, i.e., the *Internet* (*WLAN*) and *Bluetooth* relay channels, and evaluate their performance regarding relay distance and speed. Practical results of performed relay attacks are given.
- We introduce an ISO/IEC compliant way to extend the relay time during the anticollision loop of ISO/IEC 14443 A. By injecting bit collisions in the tree walking algorithm, an adversary is able to extend the relay time up to several seconds if needed. The proposed method is useful in cases where an unknown but constant UID of a card has to be relayed in a first pass or if an interleaved (challenge response) protocol is used in practice, as proposed by Feldhofer [39].
- Compared to the recent work of Francis *et al.* [41] we present a more effective relay attack by applying a custom-made proxy that is highly flexible. It allows more sophisticated relay attacks by custom parameterizations and optimizations on different communication layers, e.g., increasing the relay distance.

This chapter is organized as follows. Section 7.1 gives an introduction to relay attacks. The preliminaries to relay attacks are given in Section 7.2 followed by a discussion of the attack scenarios and the used setup in Section 7.3. Experimental results are given in Section 7.4 followed by conclusions in Section 7.5.

7.1 Introduction

Contactless smart cards based on the Radio Frequency Identification (RFID) or Near-Field Communication (NFC) technology are frequently used in applications like ticketing, access control, or cashless payment. The popularity of these systems can be described by easy handling and the increased comfort compared to contact-based systems. The majority of contactless smart cards operate up to a distance of ten centimeters according to the used ISO/IEC 14443 smart card standard. This relatively small communication range often gives a misleading impression of security. In Chapter 2 we already discussed RFID and NFC tags in the context of side-channel analysis attacks. In this chapter we put the focus on another class of attacks which exploit the contactless communication of RFID/NFC systems, known as relay attacks. The main idea is to place a proxy device (or often referred to as *ghost*), which impersonates a victim's card, in close proximity to the reader. The proxy then forwards all messages to a mole (or often referred to as *leech*) that fakes an authentic reader to a victim's card. The distance between the proxy and the mole can thereby be as large as possible and as far as the response time is sufficiently short, e.g., Sportiello and Ciardulli recently demonstrated a successful relay attack over more than 300 miles [132]. In contrast to our work they require a public network. Eddie Lee [83], as another

example, successfully relayed the communication of contactless payment systems using two (low-cost) NFC smart phones over a WiFi relay channel. However, NFC-mobile phones as proxies or moles, as reported and used also in [42, 83, 132], lack in low relay times typically much larger than 10 milliseconds. Custom-made proxies using analog RFID-relay circuits, in contrast, are very fast (less than a few microseconds) but they are passive and do not allow modifications of the relay content.

7.2 Preliminaries

In this section we first give an introduction to important parameters which have to be considered for relay attacks. A majority of these parameters is defined in the ISO/IEC 14443 standard. Next, two setups for relay attacks used for our studies are discussed. First, attacks applying NFC-enabled smart phones and second, attacks applying custom-made devices. Afterwards, we discuss different setups which have been used in related work.

7.2.1 The ISO/IEC 14443 Timing Constraints

The ISO/IEC 14443 [61] is an international standard that defines all necessary parts to allow a contactless communication with identification cards. These cards can be smart cards, RFID transponders, or any other integrated circuit that is attached to an antenna. These devices are referred to as Proximity Integrated Circuit Cards (PICCs)[1] that communicate with a reader—the Proximity Coupling Device (PCD). The physical characteristics, power and signal interfaces, and communication protocols for both PICCs and PCDs are defined in four parts. Part one and two define the mechanical properties, dimensions, and modulation/demodulation types and the necessary coding methods. Part three, in particular important for relay attacks and the following terminology, specifies the initialization and anticollision process between the PICCs and the PCD. During this phase, all PICCs in the reading field of the PCD are getting selected by challenging them with a request command (defined as `REQA` or `REQB` in the standard types A and B, respectively). If there are more than one PICCs in the field, an anticollision loop is initiated that is used to detect and correctly select all PICCs in the proximity. We now list the most important parameters of the standard that are required for the attacks described in the following sections. The numbers given are valid for a bit rate of 106 kbps, for higher bit rates the numbers are slightly different.

The Frame Delay Time (FDT). The FDT is the time between two frames in opposite direction. During initialization and anticollision, the FDT from PCD to PICC has to be 91.15 μs (if the last bit of the PCD frame equals 1) or 86.43 μs

[1] Proximity-coupled cards operate in the electromagnetic near-field of a reader device that emits either a 125 kHz (LF) or 13.56 MHz (HF) carrier signal. The typical reading range of these systems is 10 cm.

(if the last bit of the PCD frame equals 0) in case of short frames[2]. For standard frames (higher-level commands), the FDT is calculated according to Equation 7.1 (last bit 1) and Equation 7.2 (last bit 0), respectively. f_c equals to the carrier frequency, i.e., 13.56 MHz.

The FDT from PICC to PCD has to be at least 86.4 μs, i.e.,

$$FDT = \frac{n \cdot 128 + 84}{f_c} \tag{7.1}$$

and

$$FDT = \frac{n \cdot 128 + 20}{f_c}, \tag{7.2}$$

where $n \geq 9$ (in case of a 106 kbit/s data rate).

The Frame Waiting Time (FWT). The FWT equals the maximum time the PCD has to wait for an answer of the PICC. This value can be set by the PICC in the Answer to Select (`ATS`) command using the according Frame Waiting Integer (FWI) value. FWI values in the range from 0 to 14 are supported which lead to FWT values between 302 μs and 4 989 ms (c.f. Equation 7.3).

$$FWT = \left(256 \cdot \frac{16}{f_c}\right) \cdot 2^{FWI} \tag{7.3}$$

Waiting Time Extension (WTX). If the PICC requires more time than the current FWT in order to process the received command from the PCD, it can request an extension of the time by sending a Waiting Time Extension (`WTX`) command to the PCD. The PCD has to answer with the same WTX command for confirmation. 4 949 ms is the maximum value the waiting time can be extended to. The WTX concept is useful for cryptographic calculations as they are typically relatively time consuming.

7.2.2 Relay using NFC-Enabled Smart Phones

Amongst the most straight-forward solutions for performing relay attacks is to use an NFC-enabled smart phone. The phone can act as a proxy device (by receiving and forwarding data from a PCD) and/or as a mole (acting as a PCD to a target PICC). According to NFC World [123], more than 320 smart phones that can be bought today support NFC (September 2015).

Next to NFC, common smart phones also support the use of other communication technologies such as WLAN, GSM, or Bluetooth. These technologies can be also used as relay channels as we will describe in the next section. Before this, we list the main advantages of these phones for performing relay attacks and also highlight their disadvantages and limitations.

Advantages. One of the most obvious advantages of smart phones is the *ease of use*. There exist many tools and software development environments to write own software to communicate with other devices in the proximity or even

[2]Short frames are used to initiate a communication, e.g., `REQA` or `WUPA` commands.

around the world (using WLAN or GSM/UMTS/LTE). The phones usually integrate hardware support for any communication technology that can be accessed by a specified API. That means that users can simply transfer messages over a communication channel (e.g., Bluetooth or WLAN) by calling a simple API function that is executed transparently to the application. Furthermore, smart phones are widely used and integrated in our community so that they do not attract much attention when they are applied in a relay attack.

Disadvantages. What is described as a big advantage of NFC-enabled smart phones on the one hand, is one of the largest limitations on the other hand. Almost all smart phones integrate special hardware ICs for the individual communication technologies. These hardware modules can be used and accessed only by a set of given interfaces. The use and even the modification of certain lower-level commands is highly limited and often not possible in practice. For instance, for most of the available smart phones, the Unique Identifier (UID) is constant (unalterable) or random (except of some constant manufacture-dependent bytes). Practical relay attacks using mobile phones therefore often assume that the UID is not going to be checked by the RFID system (because the UID can actually not be cloned and relayed by the proxy). For example, Eddie Lee has shown that mobile payment systems like Google wallet do not check UIDs of credit cards which allows relaying of payment transactions using NFC smart phones [83]. Next to this, it is often not possible to define or modify low-level protocol parameters, for example, the Select Acknowledge (`SAK`) information that defines the individual card type or the Frame Waiting Time (`FWT`) in the Answer to Select (`ATS`) command in ISO/IEC 14443-4. Also especially when using the smart phone as a proxy device, it is often not possible to initiate a Waiting Time Extension (`WTX`) that is necessary to request a potentially necessary amount of additional (relay) time. Summing up, doing anything that deviates from the ISO standard is not possible when using a smart phone as proxy.

7.2.3 Relay using Custom Proxy/Mole

Another solution for performing relay attacks is to use custom hardware. Custom-made proxies and/or moles have the advantage that they are fast and/or very flexible depending on whether they use dedicated controllers or not. We therefore distinguish between *analog* and *digital* custom relay devices. Analog devices make use of analog circuits not involving digital-signal conversions which makes the relay communication very fast (far below milliseconds as, for example, shown by Thevenon *et al.* in [138]). They are typically passive and do not modify the content of the relay messages. Digital devices, in contrast, involve an RF-to-digital conversion including a microcontroller or processor. They can be active and allow the modification of all parameters of the RFID-network stack. In particular, they allow cloning of UIDs and the modifications of all ISO/IEC commands. For example, it is possible, as highlighted by Issovits and Hutter [62], to intentionally cause an `FWT` timeout to force the PCD to send an `R(NAK)` command which allows the proxy to re-send the message (and thus gain additional time for the relay process). Another way to extend the time, as it will be shown

later in this chapter, is to modify individual bits of the message such that, e.g., the CRC is incorrect or by causing bit collisions. This also forces the PCD to send an `R(NAK)` command and allows re-sending of messages according to the specification in ISO/IEC 14443-4 [61]. Note that these bit modifications can be made using custom proxies to fool standard compliant PCDs and PICCs without further interventions. Furthermore, different data rates between PCD-proxy and mole-PICC can be specified by custom devices in order to gain additional time for the relay attack. Custom-made moles also allow to generate a HF field with a frequency higher than 13.56 MHz to increase the data processing speed of PICCs, which leads to shorter relay times. Related work also shows that custom devices allow to significantly increase communication ranges, e.g., Oren *et al.* [104] extended the communication range between a standard PCD and their custom-made proxy to 3.8 feet (1.15 m). Kirschenbaum *et al.* [69] presented an approach to extend the range between mole and PICC. As mole they used a custom-made reader constructed with low-cost electronic equipment. This setup allows to extend the communication range up to 25 cm.

As disadvantage, however, custom-made devices are usually more expensive and time and experience is needed for the design and implementation of the setup. Details about costs for custom-made proxies/moles are given in Section 7.3.3.

7.2.4 Related Work

Kfir *et al.* [67] were one of the first who pointed out the vulnerability of authentication schemes using contactless smartcards regarding relay attacks in 2005. Hancke *et al.* [50] presented attacks against RFID systems such as eavesdropping as well as simple relay attacks (relaying the UID of a card). For relaying the data, they used a custom communication channel between proxy and mole (FSK RF link). The distance between proxy and mole was up to 50 meters. Issovits *et al.* [62] used an NFC enabled mobile phone as mole and a special programmable RFID tag as proxy in order to perform a relay attack. As relay channel they used Bluetooth. Note that in their work, only the higher-layer communication (ISO/IEC 14443-4) was relayed. The lower-layer communication has not been relayed because of the strict timing constraints set by the standard (a few microseconds instead of a few milliseconds). In order to relay also lower-layer communication, Thevenon et al. [138] presented two different setups for relay attacks with delay times lower than 2 μs. The first setup uses a coaxial cable between two antennas. A passive matching circuit is sufficient to enable relay distances of up to 20 meters. A wireless link is used in the second setup. In order to allow a far-field communication with very low delay, the reader signal is modulated on a carrier. At the receiver side, the signal is demodulated. The demodulated signal equals the reader signal with a low delay.

A practical long-distance relay attack was presented by Sportiello *et al.* [132]. They relayed the communication of an ePassport using the Internet as a relay channel. Francis *et al.* [42] have used NFC smart phones for proxy and mole and Bluetooth as relay channel for their attack in order to relay unencrypted data. As

Table 7.1: Overview of different relay-attack setups (types and channels used).

Proxy / **Mole**	Custom (digital) / NFC Phone	NFC Phone / NFC Phone	Custom (digital) / Custom (analog)	Custom (analog) / Custom (analog)
Bluetooth	our work [62], [122]	our work [41], [42]		
Internet		[132]		
Custom			[50]	[138]
WLAN		our work [83]		

they have relayed a peer-to-peer communication, where both parties are active, they did not investigate connection parameters like `FWI` or `WTX`. Vulnerabilities introduced by the usage of smart phones emulating contactless smart cards (e.g., as contactless bank cards) have been discovered by Anderson [4] in 2007. Lee [83] successfully relayed the communication of a contactless credit card transaction by using two smart phones. The fact that the system does not validate the UID enables this attack. Timing constraints are not evaluated in this work, the performance of a WLAN relay channel is sufficient in order to succeed with the attack.

Table 7.1 gives an overview of related work and our work and their used setup and relay channel. For our evaluations we use a modified version of the digital custom-made proxy of Issovits and Hutter [62] and Silberschneider *et al.* [122], respectively. In contrast to [62] we relay an encrypted communication to point out that encryption alone is not a valid measure to counter relay attacks. Furthermore we discuss the advantages of custom-made proxies compared to NFC-enabled smart phones. We extend [122] by doing detailed timing analyses and also compare the Bluetooth relay channel with the WLAN relay channel. Francis *et al.* [41, 42] used a Bluetooth relay channel between two NFC phones but they did not discuss the limitations of this setup compared to custom devices that is a main contribution of this work. Finally, Lee *et al.* [83] also used WLAN as relay channel between two NFC phones but we extend this setup in this work by presenting a "three-phones-in-the-middle" scenario which leads to an improved relay distance.

7.2.5 Terminology

Figure 7.1 shows the communication flow for a standard smart card (upper plot) and a relay-attack scenario (lower plot). Throughout the rest of the chapter, we use the following terminology:

- t_{PP} represents the duration from proxy request to mole response,
- t_{RR} represents the duration from PCD/reader request to proxy response, and

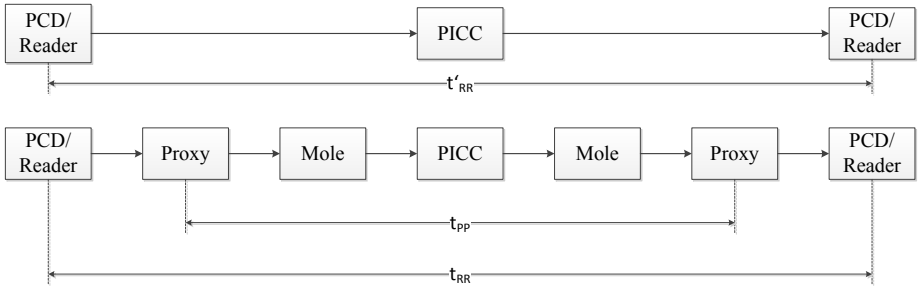


Figure 7.1: The upper part shows a scenario without relaying and the lower part shows a scenario performing a relay attack. Note that the reader on the left and right side represent the same device.

- t'_{RR} denotes the time from PCD/reader request to PICC response (no relay).

The most important parameter is t_{PP}, the time between proxy request and mole response. This time covers the entire relay chain. This value varies for different relay channel types and proxy/mole devices.

7.3 Attack Scenarios and Used Setups

For the relay attacks described in the following, we focus on relaying an AES authentication process between a PCD and an ISO/IEC 14443 Type A PICC. First, we describe the authentication process in a detail. Second, we describe our used setups using two and three NFC smart phones. Finally, we introduce our custom proxy device.

7.3.1 Relaying an AES Authentication Process

Authentication is required for many applications such as access control, ticketing, or e-passports which makes it an attractive target for relay attacks. Most of the commercially available smart cards that include security features, e.g., the Mifare DESFire card family, implement the authentication process according to ISO/IEC 9798-2 [58]. This standard specifies a challenge-response protocol where the PCD sends a random challenge to the PICC that performs an encryption operation on the challenge (and possibly optional data in addition). The PICC sends the answer back to the PCD which can then decide if the PICC is authentic or not by decrypting and evaluating the answer of the PICC.

Figure 7.2 shows the commands exchanged during our relay attacks. In a first step, the PCD starts the initialization and anticollision phase and sends a `REQA` and `SELECT` command to the proxy device. The proxy answers with its UID and changes into *ACTIVE* state. After that, ISO/IEC 14443-4 commands

are exchanged using the block format of the transmission protocol. The encapsulated messages are formatted according to ISO/IEC 7816-4 APDUs [57]. The PCD sends the challenge to the proxy within an *INTERNAL_AUTHENTICATE* command. The proxy forwards the challenge to the mole over either Bluetooth or WLAN, which then forwards the challenge to the PICC using NFC. Meanwhile, the (custom digital) proxy might send a Waiting Time Extension (WTX) in cases where the relay communication is slow in order to request an additional response time for the PICC. The answer of the PICC is then sent back to the PCD over the NFC and Bluetooth/WLAN relay link.

7.3.2 The "Phones-in-the-middle" Attack

First, we use two NFC-enabled smart phones and perform a relay attack using Bluetooth (as similarly done by Francis et al. [42]) as a reference attack. Second, we performed the same attack over a WLAN link and compare the results in terms of performance and relay range. Finally, we add another NFC phone that acts as a WLAN access point to extend the relay communication link. The same evaluation is done in this case.

The Proxy. As a proxy, we used a Google Nexus S smart phone. This smart phone runs the Android operating system and it has an NFC chip integrated which enables it to act as an NFC reader. In order to allow emulation of a PICC, we made use of a modified operating system kernel called CyanogenMod 9.1. Card emulation is per se only supported by the Android operating system Version 4.4 (*KitKat*). The Google Nexus S smart phone does not receive an OS update to Version 4.4, so we decided to install and use CyanogenMod 9.1 for that purpose. A detailed description for that process as well as how to use the card emulation can be found in [37]. Note that the proxy device is under control of the attacker so the required modifications regarding the operating system do not influence the applicability of the attack. On the other hand, newer devices running Android 4.4 or higher can make use of the host card emulation (HCE) feature in order to act as a proxy in relay attacks[3].

As already described in Section 7.2.2, the use of NFC phones comes along with certain restrictions. One of these restrictions is that the UID of the smart phone is generated randomly for every new `SELECT` command of the PCD. In our scenario, the UID is four bytes long where the first byte is fixed to the manufactory value 0x08. We therefore assume an RFID system that does not check the UID but only verifies higher-level protocol commands. In particular, the used smart phone only allows to send ISO/IEC 14443-4 commands using I-blocks. As an additional drawback, it is not able to send Waiting Time Extensions (WTX) to the PCD since this is automatically handled by the device in hardware. Our smart phone (Google Nexus S with Cyanogenmod 9.1 OS) further sets the FWT to 77.3 ms. It turned out, however, that this value is sufficient for our relay attacks as demonstrated in the next section.

[3] More information about the HCE feature can be found at `https://developer.android.com/guide/topics/connectivity/nfc/hce.html`.

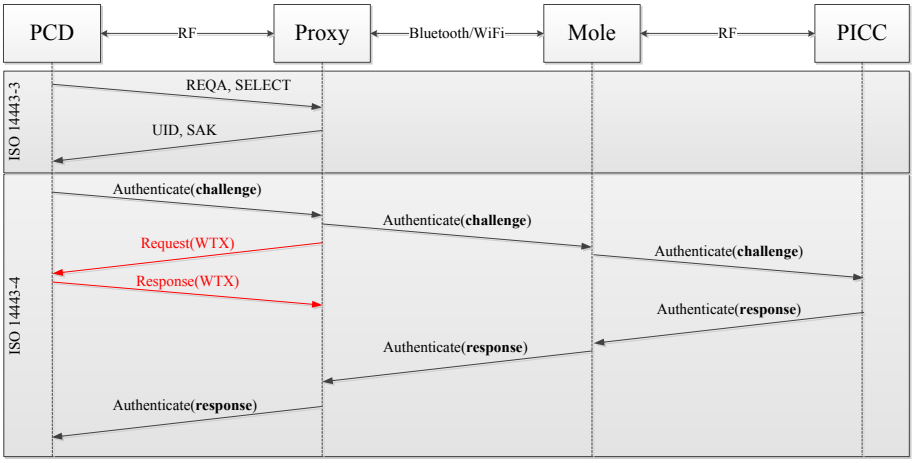


Figure 7.2: Relay communication flow during an AES authentication process.

The software application that runs on our proxy works as follows. As soon as the proxy is in the reading range of the PCD and after it has been successfully selected, it listens to the commands of the PCD. If the device receives an *INTERNAL_AUTHENTICATE* command, the challenge is forwarded to the mole. At this point the relay channel can be set to either WLAN or Bluetooth. The received response from the mole is then forwarded to the PCD using the NFC interface of the smart phone. The time required for the relay communication (t_{PP}) is measured using the time measurement ability of Java (`System.currentTimeMillis()`).

The Mole. As a mole, we also used a Google Nexus S smart phone (running Android Version 4.1.2). Note that in case of the mole, no modifications of the operating system are required because no card emulation feature is required. Instead, the mole needs to act as a conventional PCD.

The application on the mole works as follows. First, the PICC is selected and as soon as an *INTERNAL_AUTHENTICATE* command is received, the message is forwarded to the PICC via the NFC/RFID channel. After reception of the response from the PICC, the data is simply forwarded to the proxy on the selected relay channel.

Three Phones in the Middle. In this scenario, we added another smart phone between the proxy and the mole to act as a WLAN access point (AP). Note that this third smart phone does not necessarily has to support NFC functionality. We used a third Google Nexus S smart phone and enabled the "Mobile WLAN-Hotspot" feature on that device. If enabled, the proxy as well as the mole can connect to this AP and therefore extend the relay channel by a factor of two. Note that in this scenario, no Internet connection is required. Instead, the communication is routed over the AP device under control of the attacker.

This scenario increases the applicability of the relay attack due to a larger relay distance.

7.3.3 A Custom Relay Proxy

For circumventing the limitations introduced by using an NFC phone as a proxy, we also used a custom-made proxy device which is similar to the OpenPICC [23] or Proxmark [146] devices. The design of the analogue part of our custom-made proxy is similar to the IAIK demotag introduced in Section 2.2 but instead of the FPGA a microcontroller is used to implement the digital part. Our custom proxy as well as the Proxmark RFID simulator can be assembled for less than 400 EUR. The development of our proxy was an iterative process lasting several years, including hardware improvements (analog front-end, interfaces, etc.) and software improvements (increasing the number of supported protocols and applications).

Figure 7.3 shows our custom-made proxy. The main parts of the proxy are a programmable Atmel AVR microcontroller—therefore analog signals are converted to digital signals and vice versa during the communication. The left board in Figure 7.3 shows the microcontroller board, which implements the ISO/IEC 14443 protocol (type A) and allows sending and receiving of higher-level APDUs. The analog part consists of an HF antenna and a simple modulation/demodulation circuit that is connected to the I/O of the AVR microcontroller. The analog part is placed on the main board shown on the right side in Figure 7.3. Furthermore, we connected a self-made Bluetooth module to the microcontroller (using an available RS232 interface of the AVR) to allow communication with a mole.

Since the microcontroller is freely programmable, we are able to set custom UIDs (4, 7, or 10 bytes). Thus, we are able to completely clone the UID of an existing PICC. Furthermore, we have the possibility to make any modifications in the lower-protocol level, i.e., setting the same `SAK` or `FWT` values as the victim's PICC. For example, setting the Frame Waiting Integer (`FWI`) to the maximum value corresponds to a FWT of nearly 5 seconds which is far enough to relay a communication around the world using the Internet as a relay channel [132], for instance. The proxy also allows sending of Waiting Time Extensions (`WTX`) in order to increase the `FWT` for the subsequent command.

Using the custom-made proxy, the UID of the PICC can be first challenged by the mole which sends the UID to the proxy. The proxy can then simply clone the UID before the actual initialization and anticollision of the PCD, in contrast to the usage of a smart phone as proxy. This cloning can be done if the UID is considered as a known constant. If it is not a known constant and if the UID will be checked by the RFID system (especially in cases where only a single pass is possible), we propose the following anticollision-time extension. The proposed extension allows to gain additional time during the anticollision of PICCs to successfully relay the (unknown) UID between PICC and proxy before answering to the PCD with the correct UID of the victim. The proposal is fully standard compliant and does not require any modifications on the PCD

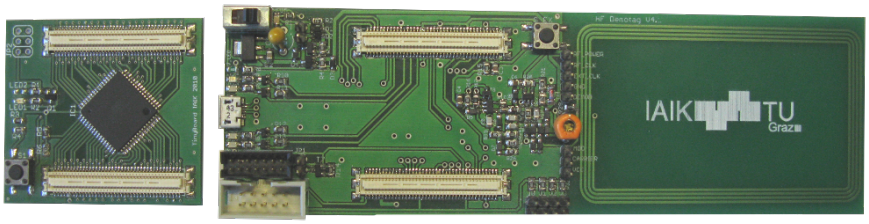


Figure 7.3: The custom made proxy. Left: microcontroller board, Right: main board including analog part.

or PICC side.

ISO/IEC-Compliant Anticollision Time Extension. In order to increase the response time during PICC anticollision, we propose to induce single bit faults during the anticollision loop. The idea is very similar to the *blocker tag* proposal of Juels, Rivest, and Szydlo [63] that is used to successfully block individual UIDs for privacy-preserving reasons. In our case, however, we do not entirely block the cards but we rather exploit the fact that the time for the anticollision loop can be extended if collisions occur which can be accomplished using our custom-made proxy.

In the following, we briefly introduce the anticollision loop (binary search tree algorithm or often referred to as "tree walking" protocol) as specified in ISO/IEC 14443-A [61]. The idea, however, can be also applied for slotted ALOHA (type B) or other (proprietary) singulation protocols (e.g., UHF EPC Gen2). The following algorithm is a recursive depth-first search in a binary tree and works as follows.

1. First, the PCD sends a SELECT command (`SEL`, i.e., $0x93$ in case of a 4-byte UID) followed by the actual Number of Valid Bits (NVB)[4], which is per default $0x20$.

2. After that, all PICCs in the field of the PCD answer with their UIDs. If there are several PICCs in the field of the PCD, collisions might occur (can be recognized by incorrect field modulations). In this case, the PCD identifies the bit position of the collision and re-sends all UID bits up to the position where the collision occurred and adds a 0 or 1 (the PCD sets the NVB accordingly).

3. Now, only PICCs that start with the same UID bit prefix answer with the remaining bits. This is repeated recursively until all PICCs are identified.

Let's assume a victim's PICC with a 4-byte UID. Then, there are $2^k = 2^{32}$ possible UIDs that can be identified by the given algorithm, where k denotes the binary tree depth. Now, further assume that our custom-made proxy device

[4]The Number of Valid Bits (NVB) defines the number of already correctly received UID bits of a PICC.

is able to intentionally cause bit collisions by sending a 0 or 1 simultaneously. Then, a proxy can cause bit collisions for the first x bits of all possible UIDs to gain additional relay time, where x represents the required tree-search depth to succeed a relay attack in a given time.

An example. Let's estimate the runtime needed to send one anticollision command. According to the standard, this needs $FDT_1 + T_{Anticollision\_frame} + FDT_2$ microseconds. The FDT_1 represents the FDT between the PCD and PICC, i.e., 91.15 μs or 86.43 μs dependent on the last bit transmitted by the PCD. The $T_{Anticollision\_frame}$ represents the time needed for transmitting the anticollision frame, i.e., $9.4 \cdot (20 + NVB)$ μs. Finally, FDT_2 represents the FDT between the PICC and PCD, i.e., 86.43 μs. So the amount of time needed to transmit and receive an anticollision command (without considering the PCD processing time to prepare the anticollision commands) is between 361 μs ($= 2 \cdot 86.43 + 9.4 \cdot (20 + 0)$) and 554 μs ($= 91.15 + 9.4 \cdot (20 + 20) + 86.43$).

Now, if a proxy would like to extend the anticollision time to, for example, 100 milliseconds, and if we assume 450 μs for a single anticollision command, the proxy has to cause bit collisions for the first x bits of the UID, i.e.,

$$2^x \cdot 450 = 100\,000 \rightarrow x = log_2 \left(\frac{100\,000}{450} \right) = 7.796. \tag{7.4}$$

7.4 Results

In this section, the results of the performed relay attacks are presented. First, we present the results obtained by using NFC smart phones. Here we were able to reach relay distances of 40 meters when using Bluetooth as relay channel and 60 meters when using WLAN as relay channel. In the WLAN scenario the relay distance can be increased to 110 meters when using a third smart phone. t_{pp} mean values of 67 ms/68 ms can be achieved, depending on the setup. Second, results of our custom proxy are discussed. Here, only Bluetooth can be used as relay channel. The maximum relay distance was found to be 40 meters, what is in-line with the smart-phone experiments using Bluetooth. The mean t_{pp} value is 162 ms what is approximately 95 ms longer than in the smart-phone scenarios. So we can conclude that for scenarios where the timing is critical, the relay attacks applying smart phones outperform our custom device. On the other hand, the custom device provides more flexibility and the timing penalty can be reduced by hardware improvements. The interface between Bluetooth module and microcontroller turned out to be the main reason for the larger t_{pp} values.

7.4.1 "Phones-in-the-middle"

In a first experiment, we used a Bluetooth connection as a relay channel between proxy and mole. 500 authentication runs have been performed using this setup. We measured the time between receiving the request of the reader and receiving the response of the mole, i.e., t_{PP}, on our proxy device. Figure 7.4 shows the result of this experiment. The distance between proxy and the mole was about

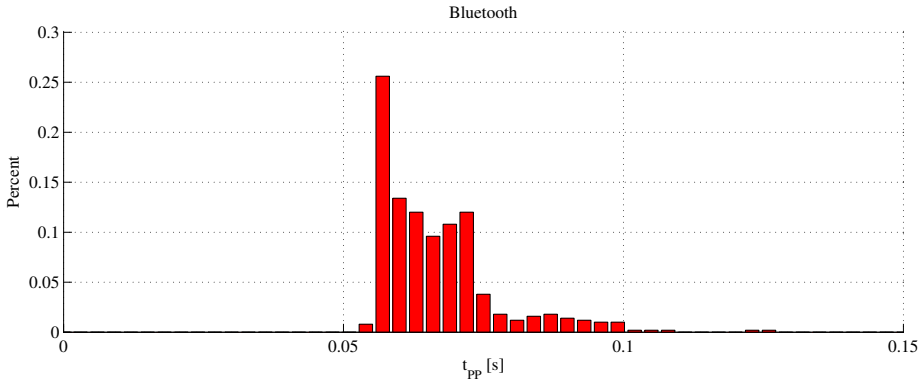


Figure 7.4: Result of 500 measurements of t_{PP} with constant distance between proxy and mole using Bluetooth as relay channel.

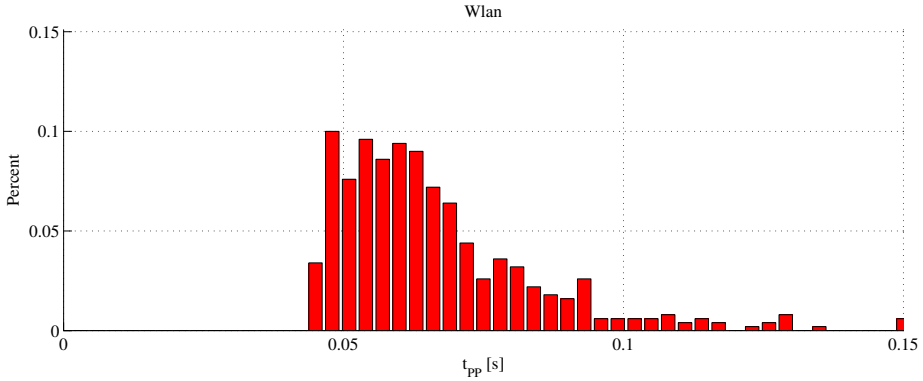


Figure 7.5: Result of 500 measurements of t_{PP} with constant distance between proxy and mole using WiFi as relay channel.

2 meters. The mean value for t_{PP} for the 500 measurements is 67 ms and the minimum value is 55 ms. 90 % of the values are in the range between 55 ms and 80 ms and 99 % of the values are in the range between 55 ms and 100 ms.

After that, we increased the distance between proxy and mole until the connection got lost. It was possible to increase the distance to about 40 meters without loosing the Bluetooth connection for our setup. It showed that the distance between proxy and mole does not significantly has an influence on t_{PP}.

In a second experiment, we relayed the communication over WLAN instead of Bluetooth. We enabled WLAN on both proxy and mole, at which one device acts as a WLAN access point and the other one connects to it. Again t_{PP} was measured for 500 authentication runs while the distance between proxy and mole was set to 2 meters. The result of this experiment is shown in Figure 7.5. The mean value for t_{PP} for the 500 measurements is 67 ms and the minimum value

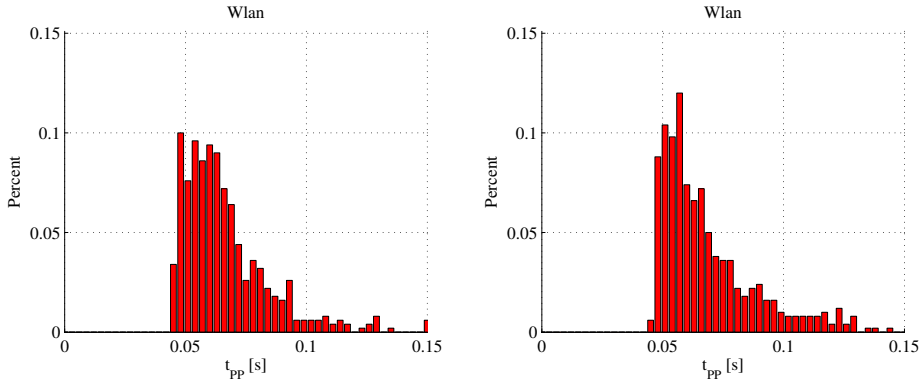


Figure 7.6: Comparison of t_{PP} for the scenario with only two smart phones (left) and three smart phones (right).

is 46 ms. 90 % of the values are in the range between 46 ms and 80 ms and 99 % of the values are in the range between 46 ms and 129 ms.

After this experiment, we also increased the distance between proxy and mole to make a comparison to Bluetooth. Using WLAN, a relay distance of up to 60 meters was possible. The distance does not significantly influence t_{PP}.

"Three Phones-in-the-middle". We placed a third NFC phone between proxy and mole and enabled a WLAN access point. It showed that this additional phone theoretically doubles the relay distance and does not significantly increase t_{PP}. With this setup, we achieved a maximum relay distance of over 110 meters. A comparison between two and three "phones-in-the-middle" is shown in Figure 7.6. The mean value for t_{PP} for the 500 measurements using three smart phones is 68 ms and the minimum value is 46 ms. 90 % of the values are in the range between 46 ms and 90 ms and 99 % of the values are in the range between 46 ms and 130 ms.

7.4.2 Bluetooth vs. WLAN

With WLAN we were able to achieve lower t_{PP} values compared to Bluetooth (46 ms vs. 55 ms). When keeping in mind the maximum delay time of about 5 seconds allowed by the ISO/IEC standard, the difference of 9 ms between the

Table 7.2: Comparison of results between Bluetooth and WLAN using NFC phones.

		Bluetooth	WLAN, 2 phones	WLAN, 3 phones
$t_{\mathbf{PP\ min}}$	[ms]	55	46	46
$t_{\mathbf{PP\ mean}}$	[ms]	67	67	68
Interval 90 %	[ms]	55 - 80	46 - 80	46 - 90
Interval 99 %	[ms]	55 - 100	46 - 129	46 - 130
Max. distance	[m]	40	60	110

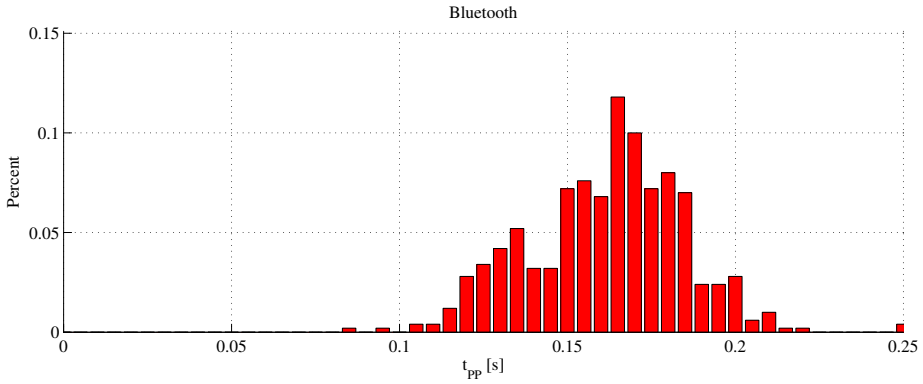


Figure 7.7: Result of 500 measurements of t_{PP} with constant distance between proxy and mole using Bluetooth as relay channel and a programmable transponder as proxy.

two technologies does not affect the application of the relay attack. The fact that the FDT value cannot be influenced and is fixed to 77.3 ms for the card emulation on the smart phone shows that in this scenario a fast connection is still advantageous. With the WLAN connection, 76 % of the t_{PP} values were below FDT and for the Bluetooth connection 88.4 % of the t_{PP} values were below FDT. The variation of the t_{PP} values for subsequent authentication runs is higher in the WLAN scenario but even in the worst case t_{PP} did not exceed 130 ms. With WLAN and two smart phones, distances of up to 60 meters were achieved in our experiments which is a bit more compared to the 40 meters achieved with Bluetooth. By introducing a third smart phone acting as an access point, the distance can nearly be doubled to 110 meters. In Table 7.2, the achieved results are summarized.

7.4.3 Custom Relay Proxy

We evaluated the performance of our custom proxy by measuring the relay timings for 500 authentication runs. The result is depicted in Figure 7.7. The distance between proxy and mole was set to 2 meters as in the previous experiments and we did not change the distance during this experiment. The mean value for t_{PP} is 162 ms and the minimum value is 86 ms. 90 % of the values are in the range between 86 ms and 187 ms and 99 % of the values are in the range between 86 ms and 212 ms. In order to find the maximum distance for relaying the communication, the distance between proxy and mole has been increased step by step. The mole lost the Bluetooth connection to the proxy at a distance of approximately 40 meters in our experiment. Table 7.3 lists the exact relay timings.

Table 7.3: Results for our custom-made proxy using a Bluetooth channel.

		Bluetooth
$t_{\mathbf{PP\ min}}$	[ms]	86
$t_{\mathbf{PP\ mean}}$	[ms]	162
Interval 90 %	[ms]	86 - 187
Interval 99 %	[ms]	86 - 212
Max. distance	[m]	40

7.5 Conclusion

In this chapter, we pointed out how practical relay attacks can be improved when using a custom-made proxy compared to NFC-enabled smart phones. We started with a discussion of the limitations which arise when using a smart phone as a proxy device. Some of these limitations are: the UID cannot be set to a fixed, predefined value; adaption of low-level ISO/IEC protocol parameters is not possible; no active/direct request for Waiting Time Extensions; no way to modify lower-level RFID protocol commands. We emphasized that these limitations can be circumvented by using custom-made proxies and presented practical results of attacks using a microcontroller-based (low cost) device.

The results show that practical relay attacks performed with two NFC smart phones pose a real threat due to the fact that the introduced delay time is below the maximum, tolerated delay time for both evaluated relay channels (Bluetooth, WLAN). Furthermore, the "three-phones-in-the-middle" approach allows to nearly double the relay distance without the need for a public network. Delay times are not (noticeable) affected by this modification. Our custom-made proxy is highly flexible and allows more sophisticated attacks than using NFC-enabled smart phones.

8 Conclusions

In the course of this thesis we have highlighted the power of low-cost setups for performing physical attacks such as side-channel analysis (SCA) and fault analysis (FA). All the presented setups are built-up with off-the-shelf equipment. The relatively easy access to such low-cost setups make them a serious threat for real-world devices. This is because physical attacks requiring such a setup can be performed by a high number of attackers and are therefore likely to lead to a real-world exploit.

For SCA attacks targeting RFID systems, we present a novel measurement approach for EM measurements called resolution optimization. This approach does not require any pre-processing circuitry. It also performs better at higher distances compared to an approach based on analogue demodulation presented in related work. We further take advantage of the resolution optimization to conduct SCA attacks at distances up to one meter between attacked device and measurement antenna. At this distance, we are able to recover an AES key used for authenticating a prototype RFID-tag chip. This results clearly highlight that exploitable side-channel information can be measured at multiples of the communication range of a few centimeters. Based on this, the assumption that the security of such RFID systems is defined by the short communication range, is not valid. Relay attacks denote another attack that allows to circumvent the limitation of a short communication range for malicious intentions. In this context we present a low-cost relay setup. This setup only requires two NFC-enabled smart phones and allows to relay encrypted authentications between reader and tag at distances up to 60 meters. For improving the attack we suggest to replace one smart phone by a custom-made device. This custom-made device can be assembled at low-costs and allows to circumvent several limitations which arise when using the smart phone.

Next to RFID systems, microcontrollers for sensor nodes have also been shown to be vulnerable to SCA attacks based on EM measurements. Exemplary attacks targeting AES implementations on an off-the-shelf microcontroller show the importance of the measurement location when using the EM side channel. By selection of the correct measurement location, the number of required measurements for key recovery can be minimized. Next to an off-the-shelf microcontroller, we further analyzed a prototype chip specialized for sensor-node applications. The setup of the chip allowed us to perform high-resolution, semi-invasive EM measurements. The integrated cryptographic hardware modules were evaluated and the results show the power of the high-resolution EM measurements compared to conventional power measurements. We can conclude that in the case of EM side-channel scenarios, the measurement effort for key recovery can be minimized by carefully selecting the correct measurement parameters. This is especially valid for the application of low-cost equipment.

Next to EM measurements, we further present a low-cost, power-measurement setup for evaluating a prototype ASIC. The ASIC implements an SCA-protected, authenticated encryption algorithm based on KECCAK, with the design goals low area and low power consumption. This allows the ASIC to be applied in future RFID systems. We are among the first to perform DPA attacks targeting a keyed KECCAK instance implemented on a taped-out ASIC. The results of the DPA attacks reveal that because of the resource-efficient implementation the proposed secret-sharing countermeasure does not lead to the expected security gain. Here we take advantage of the area and runtime figures of the chip to bring the introduced overhead of the countermeasures in relation with the security gain. We can conclude that only a combination of secret sharing with a second countermeasure, i.e. hiding, leads to a satisfying security level.

Next to passive attacks, we have also studied the applicability of the low-cost approach for active attacks. We cover setups for non-invasive and semi-invasive fault injection. For the non-invasive case, tampering with the clock signal, the supply voltage, and the ambient temperature turned out to be valid injection methods when targeting microcontrollers. Our setup also allows to combine any two of the aforementioned fault injection methods. This leads to an increased efficiency in terms of occurrence and reproducibility of the injected fault. For the semi-invasive case we present a low-cost, optical fault-injection setup. We apply pulsed laser diodes as light source and the laser beam is focused by using the lenses of a microscope. Diodes with different wavelengths (808 nm and 1 064 nm) can be used to allow front-side and rear-side fault injections. Our evaluations focus on the setup parameters required for a successful fault injection. We can conclude that the laser pulse length, the laser power, and the laser focus have to be chosen carefully for successfully injecting a fault. Also the type of fault is influenced by these parameters.

Summing up the results of this thesis we can conclude that a wide range of electronic devices are vulnerable to low-cost SCA and FA attacks. Of course, the success of such attacks highly depends on the properties of the attacked device and the selection of the countermeasures. But, as shown in Chapter 2 and in

Chapter 4, the selection and implementation of the countermeasures has to be done with great care. Wrong assumptions or flaws in the implementation are likely to lead to security degradations. In addition to the equipment costs it is also conceivable to add the measurement effort and the computational effort as additional parameters for classifying the complexity of the attack. In this work we consider only the parameter measurement effort, which we discuss in the context of the implementations with countermeasures. During our investigations, the computational effort for conducting the attacks was negligible so we did not further discuss it.

9 About the Author

Author information as of October 2015.

Personal Information

- **Name:** Thomas Korak
- **Date of birth:** December 12th, 1985
- **Place of Birth:** Klagenfurt, Carinthia, Austria.

Education

- **06/2009 - 11/2011:** Graz University of Technology, Austria: Master of Science in Computer Engineering (Telematik).
- **10/2006 - 06/2009:** Graz University of Technology, Austria: Bachelor of Science in Computer Engineering (Telematik).

Professional and Academic Experience

- **2011/11 - present:** PhD student, Institute for Applied Information Processing and Communications (IAIK), Graz University of Technology.
- **2011/08 - 2011/11:** Student researcher, Institute for Applied Information Processing and Communications (IAIK), Graz University of Technology.

Author's Publications

Author's publications as of October 2015 mapped according to the corresponding chapters

Chapter 2

- T. Korak and T. Plos. Applying Remote Side-Channel Analysis Attacks on a Security-enabled NFC Tag. In *Topics in Cryptology - CT-RSA 2013, San Francisco, USA, February 25 - March 1, 2013, Proceedings*, LNCS, pages 207 – 222. Springer-Verlag, 2013

- T. Korak, T. Plos, and M. Hutter. Attacking an AES-enabled NFC Tag - Implications from Design to a Real-World Scenario. In Springer-Verlag, editor, *3rd International Workshop on Constructive Side-Channel Analysis and Secure Design, Darmstadt, Germany, 3. - 4. May, 2012, Proceedings.*, pages 17 – 32, 2012

- T. Korak and T. Plos. EM Leakage of RFID Devices - Comparison of Two Measurement Approaches. In *The 9th International Conference on Availability, Reliability and Security (ARES 2014), Fribourg, Swizerland, September, 2014, Proceedings*, pages 120–125. IEEE, 2014

- T. Korak, T. Plos, and A. Zankl. Minimizing the Costs of Side-Channel Analysis Resistance Evaluations in Early Design Steps. In *The 8th International Conference on Availability, Reliability and Security (ARES 2014), Regensburg, Germany, September 2-6, 2013, Proceedings*, pages 169–177. IEEE, 2013

Chapter 3

- T. Korak. Location-dependent EM Leakage of the ATxmega Microcontroller. In M. Debbabi, editor, *The 7th International Symposium on Foundations and Practice of Security FPS'2014, Montreal, Canada, November 03-05, 2014, Proceedings*, LNCS, pages 17 – 32. Springer-Verlag, 2014

Chapter 4

- M. Muehlberghuber, T. Korak, M. Hutter, and P. Dunst. Towards Evaluating DPA Countermeasures for Keccak on a Real ASIC. In *Constructive Side-Channel Analysis and Secure Design, The sixth International Workshop on*. Springer, 2015. in press

Chapter 5

- T. Korak and M. Höfler. On the Effects of Clock and Power Supply Tampering on Two Microcontroller Platforms. In *Workshop on Fault Diagnosis*

and Tolerance in Cryptography - FDTC 2014, Busan, Korea, September 23, 2014, Proceedings, pages 8 – 17, 2014

- T. Korak, M. Hutter, B. Ege, and L. Batina. Clock Glitch Attacks in the Presence of Heating. In *Workshop on Fault Diagnosis and Tolerance in Cryptography - FDTC 2014, Busan, Korea, September 23, 2014, Proceedings*, pages 104 – 114, 2014

Chapter 6

- T. Korak. Investigation of Parameters Influencing the Success of Optical Fault Attacks. In J. L. Danger, M. Debbabi, J.-Y. Marion, J. Garcia-Alfaro, and N. Zincir Heywood, editors, *The 6th International Symposium on Foundations and Practice of Security FPS'2013, La Rochelle, France, October 21-22, 2013, Proceedings*, LNCS, pages 140–157. Springer-Verlag, 2013

Chapter 7

- T. Korak and M. Hutter. On the Power of Active Relay Attacks using Custom-Made Proxies. In IEEE, editor, *2014 IEEE International Conference on RFID (IEEE RFID 2014), Orlando, Florida, USA, April 8-10, 2014, Proceedings*, pages 126–133, 2014

- R. Silberschneider, T. Korak, and M. Hutter. Access Without Permission: A Practical RFID Relay Attack . In *Austrochip 2013, 21st Austrian Workshop on Microelectronics, Linz, Austria, October 10, 2013, Proceedings*, pages 59 – 64, 2013

Further Contributions

- M. Muehlberghuber, F. K. Gurkaynak, T. Korak, P. Dunst, and M. Hutter. Red Team vs. Blue Team Hardware Trojan Analysis: Detection of a Hardware Trojan on an Actual ASIC. In ACM, editor, *Hardware and Architectural Support for Security and Privacy - HASP 2013, Second Workshop, Tel-Aviv, Israel, June 23, 2013, Proceedings.*, pages 1 – 8. ACM, 2013

- O. Söll, T. Korak, M. Muehlberghuber, and M. Hutter. EM-Based Detection of Hardware Trojans on FPGAs. In IEEE, editor, *IEEE International Symposium on Hardware-Oriented Security and Trust – HOST 2014, Arlington, Virginia, USA, May 6-7, 2014, Proceedings*, pages 84 – 87. IEEE, 2014

- T. Korak and L. Wilfinger. Handling the NDEF Signature Record Type in a Secure Manner. In *IEEE International Conference on RFID-Technologies and Applications (RFID-TA), Nice, France, November 05-07, 2012, Proceedings*, pages 107–112, November 2012

- E. Wenger, T. Korak, and M. Kirschbaum. Analyzing Side-Channel Leakage of RFID-Suitable Lightweight ECC Hardware. In *RFID Security - RFIDSec 2013, 9th Workshop, Graz, Austria, July 9-11, 2013, Proceedings*, Lecture Notes in Computer Science, pages 128 – 144. Springer, 2013

- H. Martin, T. Korak, E. S. Millan, and M. Hutter. Fault Attacks on STRNGs: Impact of Glitches, Temperature, and Underpowering on Randomness. *IEEE Transactions on Information Forensics and Security*, 10:266 – 277, 2014

- T. Plos, M. Aigner, T. Baier, M. Feldhofer, M. Hutter, T. Korak, and E. Wenger. Semi-Passive RFID Development Platform for Implementing and Attacking Security Tags. *International Journal of RFID Security and Cryptography [Elektronische Ressource]*, 1:16 – 24, 2012

Bibliography

[1] CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness. `http://competitions.cr.yp.to/caesar.html`, March 2013.

[2] M. Agoyan, J.-M. Dutertre, D. Naccache, B. Robisson, and A. Tria. When Clocks Fail: On Critical Paths and Clock Faults. In *Smart Card Research and Advanced Application*, pages 182–193. Springer, 2010.

[3] C. Ajluni. Two New Imaging Techniques Promise to Improve IC Defect Identification. *Electronic Design*, 43(14):37–38, July 1995.

[4] R. Anderson. RFID and the Middleman. In *Financial Cryptography and Data Security*, pages 46–49. Springer Berlin Heidelberg, 2007.

[5] R. J. Anderson and M. G. Kuhn. Tamper Resistance - A Cautionary Note. In *Proceedings of the 2nd USENIX Workshop on Electronic Commerce, Oakland, California, November 18-21, 1996*, pages 1–11. USENIX Association, November 1996. ISBN 1-880446-83-9.

[6] R. J. Anderson and M. G. Kuhn. Low Cost Attacks on Tamper Resistant Devices. In B. Christianson, B. Crispo, M. Lomas, and M. Roe, editors, *Security Protocols, 5th International Workshop*, volume 1361 of *Lecture Notes in Computer Science*, pages 125–136. Springer, 1997.

[7] Atmel. AVR1318: Using the XMEGA built-in AES accelerator, 2008. [Online; accessed 5-November-2013].

[8] Atmel. 8/16-bit AVR XMEGA A3 Microcontroller, 2013. [Online; accessed 5-November-2013].

[9] Atmel Corporation. ATmega 162/v Datasheet, 2003.

[10] Atmel Corporation. 8/16-bit Atmel XMEGA A3U Microcontrollers. Available online at `http://www.atmel.com/Images/Atmel-8386-8-and-16-bit-AVR-Microcontroller-ATxmega64A3U-128A3U_192A3U-256A3U_datasheet.pdf`, 2013.

[11] A. Auer. Scaling Hardware for Electronic Signatures to a Minimum. Master thesis, University of Technology Graz, October 2008.

[12] J. Balasch, B. Gierlichs, and I. Verbauwhede. An In-depth and Black-box Characterization of the Effects of Clock Glitches on 8-bit MCUs. In *FDTC, 2011*, pages 105–114. IEEE, 2011.

[13] A. Barenghi, G. M. Bertoni, L. Breveglieri, and G. Pelosi. A Fault Induction Technique Based on Voltage Underfeeding with Application to Attacks Against AES and RSA. *Journal of Systems and Software*, 86(7):1864–1878, 2013.

[14] L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede. Public-Key Cryptography for RFID-Tags. In *Workshop on RFID Security 2006 (RFIDSec06), July 12-14, Graz, Austria*, 2006.

[15] S. Belaïd, F. De Santis, J. Heyszl, S. Mangard, M. Medwed, J.-M. Schmidt, F.-X. Standaert, and S. Tillich. Towards Fresh Re-Keying with Leakage-Resilient PRFs: Cipher Design Principles and Analysis. *Journal of Cryptographic Engineering*, 4(3):157–171, 2014.

[16] G. Bertoni, J. Daemen, N. Debande, T.-H. Le, M. Peeters, and G. Van Assche. Power Analysis of Hardware Implementations Protected with Secret Sharing. Cryptology ePrint Archive: Report 2013/067, February 2013.

[17] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. In *SAC 2011, Revised Selected Papers*, volume 7118 of *LNCS*, pages 320–337. Springer, 2011.

[18] G. Bertoni, J. Daemen, M. Peeters, G. V. Assche, and R. V. Keer. Keccak Implementation Overview, May 2012. Version 3.2.

[19] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Building Power Analysis Resistant Implementations of Keccak. In *2nd SHA-3 Conference, August 2010*.

[20] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. KECCAK specifications, Version 2 – September 10, 2009. Available online at `http://keccak.noekeon.org/Keccak-specifications-2.pdf`, September 2009.

[21] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. The Keccak Reference. Submission to NIST (Round 3), January 2011. Available online: `http://csrc.nist.gov/groups/ST/hash/sha-3/Round3/submissions_rnd3.html`.

[22] B. Bilgin, S. Nikova, V. Rijmen, V. Nikov, J. Daemen, and G. V. Assche. Efficient and First-Order DPA Resistant Implementations of KECCAK. In *CARDIS 2013*, volume 8419 of *LNCS*. Springer, 2013.

[23] Bitmanufactur GmbH. OpenPICC. Available online at http://www.openpcd.org, 2013.

[24] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '07, pages 450–466, Berlin, Heidelberg, 2007. Springer-Verlag.

[25] M. K. Bond. *Understanding Security APIs.* PhD thesis, University of Cambridge, 2004.

[26] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults (Extended Abstract). In W. Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceedings*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51. Springer, 1997.

[27] S. Bono, M. Green, A. Stubblefield, A. Juels, A. Rubin, and M. Szydlo. Security Analysis of a Cryptographically-Enabled RFID Device. In *USENIX Security Symposium, Baltimore, Maryland, USA, July-August, 2005, Proceedings*, pages 1–16. USENIX, 2005.

[28] N. Borisov, I. Goldberg, and D. Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. In M. Naghshineh and M. Zorzi, editors, *International Conference on Mobile Computing and Networking - MobiCom, July 16-21, Rome, Italy, 2001*, pages 180–189. ACM, 2001.

[29] D. Bosomworth. Mobile Marketing Statistics 2015. `http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/`. Accessed: 2015-08-27.

[30] M. Botta, M. Simek, and N. Mitton. Comparison of Hardware and Software Based Encryption for Secure Communication in Wireless Sensor Networks. In *Telecommunications and Signal Processing (TSP), 2013*, pages 6–10. IEEE, 2013.

[31] B. Canvel, A. P. Hiltgen, S. Vaudenay, and M. Vuagnoux. Password Interception in a SSL/TLS Channel. In *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *LNCS*, pages 583–599. Springer, 2003.

[32] D. Carluccio, K. Lemke, and C. Paar. Electromagnetic Side Channel Analysis of a Contactless Smart Card: First Results. In E. Oswald, editor, *Workshop on RFID and Lightweight Crypto (RFIDSec05), July 13-15, Graz, Austria*, pages 44–51, 2005.

[33] H. Choukri and M. Tunstall. Round Reduction Using Faults. *FDTC*, 5:13–24, 2005.

[34] N. T. Courtois, S. O'Neil, and J.-J. Quisquater. Practical Algebraic Attacks on the Hitag2 Stream Cipher. In *Information Security Conference – ISC'09*, 2009.

[35] A. Dehbaoui, A.-P. Mirbaha, N. Moro, J.-M. Dutertre, and A. Tria. Electromagnetic Glitch on the AES Round Counter. In *COSADE, 2013*, pages 17–31. Springer, 2013.

[36] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani. On the Power of Power Analysis in the Real World: A Complete Break of the KEELOQ Code Hopping Scheme. In *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008, Proceedings*, 2008.

[37] N. Elenkov. Emulating a PKI Smart Card with CyanogenMod 9.1. http://nelenkov.blogspot.it/2012/10/emulating-pki-smart-card-with-cm91.html, October 2012.

[38] S. Endo, T. Sugawara, N. Homma, T. Aoki, and A. Satoh. An On-Chip Glitchy-Clock Generator and its Applicataion to Safe-Error Attack. In *COSADE, 2011*, Workshop Proceedings COSADE 2011, pages 175–182, 2011.

[39] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong Authentication for RFID Systems using the AES Algorithm. In *Cryptographic Hardware and Embedded Systems – CHES 2004, 6th International Workshop, Cambridge, MA, USA, August 11-13, 2004, Proceedings*, 2004.

[40] K. Finkenzeller. *RFID-Handbook*. Carl Hanser Verlag, 2nd edition, April 2003. ISBN 0-470-84402-7.

[41] L. Francis, G. Hancke, and K. Mayes. A Practical Generic Relay Attack on Contactless Transactions by Using NFC Mobile Phones. *International Journal of RFID Security and Cryptography (IJRFIDSC)*, 2:92–106, 2013.

[42] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis. Practical NFC Peer-to-Peer Relay Attack Using Mobile Phones. In *Radio Frequency Identification: Security and Privacy Issues*, pages 35–49. Springer, 2010.

[43] T. Fukunaga and J. Takahashi. Practical Fault Attack on a Cryptographic LSI with ISO/IEC 18033-3 Block Ciphers. In *FDTC, 2009*, pages 84–92. IEEE, 2009.

[44] K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic Analysis: Concrete Results. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, volume 2162 of *LNCS*, pages 251–261. Springer, 2001.

[45] D. Genkin, I. Pipman, and E. Tromer. Get Your Hands Off My Laptop: Physical Side-Channel Key-Extraction Attacks on PCs. Cryptology ePrint Archive, Report 2014/626, 2014. `http://eprint.iacr.org/`.

[46] C. P. Gouvêa and J. López. High Speed Implementation of Authenticated Encryption for the MSP430X Microcontroller. In *LATINCRYPT 2012*, pages 288–304. Springer, 2012.

[47] S. Govindavajhala and A. W. Appel. Using Memory Errors to Attack a Virtual Machine. In *IEEE Symposium on Security and Privacy, Proceedings of the 2003*, pages 154–165, 2003.

[48] J. Halderman, S. D.Schoen, N. Heninger, W. Clarkson, W. Paul, J. A.Calandrino, A. J.Feldman, J. Appelbaum, and E. W.Felten. Lest We Remember: Cold Boot Attacks on Encryption Keys. In *17th USENIX Security Symposium, San Jose, CA, July 2008*, pages 45–60, 2008.

[49] P. Hämäläinen, T. Alho, M. Hännikäinen, and T. D. Hämäläinen. Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core. In *9th EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools (DSD 2006), Dubrovnik, Croatia, 30. August-1 September, 2006. Proceedings*, pages 577–583. IEEE Computer Society, September 2006.

[50] G. P. Hancke. Practical Attacks on Proximity Identification Systems. In *IEEE Symposium on Security and Privacy (S&P 2006), Berkeley/Oakland, California, USA, 21-24 May*, pages 328–333, 2006.

[51] D. Hein, J. Wolkerstorfer, and N. Felber. ECC is Ready for RFID - A Proof in Silicon. In *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, Canada, August 14-15, 2008, Revised Selected Papers*, Lecture Notes in Computer Science (LNCS), September 2008.

[52] J. Heyszl, D. Merli, B. Heinz, F. De Santis, and G. Sigl. Strengths and Limitations of High-Resolution Electromagnetic Field Measurements for Side-Channel Analysis. In S. Mangard, editor, *CARDIS 2013*, volume 7771 of *LNCS*, pages 248–262. Springer Berlin Heidelberg, 2013.

[53] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. In *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, 1998.

[54] M. Hutter, S. Mangard, and M. Feldhofer. Power and EM Attacks on Passive 13.56 MHz RFID Devices. In *Cryptographic Hardware and Embedded Systems – CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, 2007.

[55] M. Hutter, M. Medwed, D. Hein, and J. Wolkerstorfer. Attacking ECDSA-Enabled RFID Devices. In *Applied Cryptography and Network Security – ACNS 2009, 7th International Conference, Paris-Rocquencourt, France, June 2-5, 2009, Proceedings*, 2009.

[56] M. Hutter and J.-M. Schmidt. The Temperature Side-Channel and Heating Fault Attacks. In *CARDIS 2014*, LNCS, pages 219–235, 2014.

[57] International Organisation for Standardization (ISO). ISO/IEC 7816-4: Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 4: Interindustry commands for interchange. Available online at `http://www.iso.org`, 1995.

[58] International Organisation for Standardization (ISO). ISO/IEC 9798-2: Information technology – Security techniques – Entity authentication – Mechanisms using symmetric encipherment algorithms, 1999.

[59] International Organisation for Standardization (ISO). ISO/IEC 15693-3: Identification cards - Contactless integrated circuit(s) cards - Vicinity cards – Part 3: Anticollision and transmission protocol, 2001.

[60] International Organization for Standardization (ISO). ISO/IEC 14443: Identification Cards - Contactless Integrated Circuit(s) Cards - Proximity Cards, 2000.

[61] International Organization for Standardization (ISO). ISO/IEC 14443: Identification Cards - Contactless Integrated Circuit(s) Cards - Proximity Cards, 2009.

[62] W. Issovits and M. Hutter. Weaknesses of the ISO/IEC 14443 Protocol Regarding Relay Attacks. In A. Collado and M. Bozzi, editors, *Conference on RFID-Technologies and Applications - IEEE RFID-TA, Barcelona, Spain, September 15-16*, pages 335–342, 2011.

[63] A. Juels, R. L. Rivest, and M. Szydlo. The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy. In *10th ACM Conference on Computer and Communication Security, Washington, DC, USA, October 27-30*, pages 103–111. ACM Press, 2003.

[64] H. Kaeslin. *Digital Integrated Circuit Design – From VLSI Architectures to CMOS Fabrication*. Cambridge University Press, 2008. ISBN 978-0-521-88267-5.

[65] T. Kasper, D. Oswald, and C. Paar. EM Side-Channel Attacks on Commercial Contactless Smartcards Using Low-Cost Equipment. In *Information Security Applications - WISA 2009, 10th International Workshop, Busan, Korea, August 25-27, 2009, Revised Selected Papers*, 2009.

[66] E. B. Kavun and T. Yalcin. A Lightweight Implementation of Keccak Hash Function for Radio-Frequency Identification Applications. In S. B. O. Yalcin, editor, *Workshop on RFID Security – RFIDsec 2010, 6th Workshop, Istanbul, Turkey, June 7-9, 2010, Proceedings*, volume 6370 of *Lecture Notes in Computer Science*, pages 258–269. Springer, 2010.

[67] Z. Kfir and A. Wool. Picking Virtual Pockets using Relay Attacks on Contactless Smartcard Systems. In *SecureComm 2005, Athens, Greece, 5-9 September 2005, Proceedings*, pages 47–58. IEEE Computer Society, September 2005.

[68] C. H. Kim and J.-J. Quisquater. Fault Attacks for CRT Based RSA: New Attacks, New Results, and New Countermeasures. In *Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems*, pages 215–228. Springer, 2007.

[69] I. Kirschenbaum and A. Wool. How to Build a Low-Cost, Extended-Range RFID Skimmer. *IACR Cryptology ePrint Archive*, 2006:54, 2006.

[70] I. Kizhvatov. Side-Channel Analysis of AVR XMEGA Crypto Engine. In *Proceedings of the 4th Workshop on Embedded Systems Security*, page 8. ACM, 2009.

[71] P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In M. Wiener, editor, *CRYPTO 1999*, volume 1666 of *LNCS*, pages 388–397. Springer, 1999.

[72] T. Korak. Investigation of Parameters Influencing the Success of Optical Fault Attacks. In J. L. Danger, M. Debbabi, J.-Y. Marion, J. Garcia-Alfaro, and N. Zincir Heywood, editors, *The 6th International Symposium on Foundations and Practice of Security FPS'2013, La Rochelle, France, October 21-22, 2013, Proceedings*, LNCS, pages 140–157. Springer-Verlag, 2013.

[73] T. Korak. Location-dependent EM Leakage of the ATxmega Microcontroller. In M. Debbabi, editor, *The 7th International Symposium on Foundations and Practice of Security FPS'2014, Montreal, Canada, November 03-05, 2014, Proceedings*, LNCS, pages 17 – 32. Springer-Verlag, 2014.

[74] T. Korak et al. TAMPRES Deliverable D5.4 Sensor Node Measurements, 2014. Available online at `http://www.tampres.eu/`.

[75] T. Korak and M. Höfler. On the Effects of Clock and Power Supply Tampering on Two Microcontroller Platforms. In *Workshop on Fault Diagnosis and Tolerance in Cryptography - FDTC 2014, Busan, Korea, September 23, 2014, Proceedings*, pages 8 – 17, 2014.

[76] T. Korak and M. Hutter. On the Power of Active Relay Attacks using Custom-Made Proxies. In IEEE, editor, *2014 IEEE International Conference on RFID (IEEE RFID 2014), Orlando, Florida, USA, April 8-10, 2014, Proceedings*, pages 126–133, 2014.

[77] T. Korak, M. Hutter, B. Ege, and L. Batina. Clock Glitch Attacks in the Presence of Heating. In *Workshop on Fault Diagnosis and Tolerance in Cryptography - FDTC 2014, Busan, Korea, September 23, 2014, Proceedings*, pages 104 – 114, 2014.

[78] T. Korak and T. Plos. Applying Remote Side-Channel Analysis Attacks on a Security-enabled NFC Tag. In *Topics in Cryptology - CT-RSA 2013, San Francisco, USA, February 25 - March 1, 2013, Proceedings*, LNCS, pages 207 – 222. Springer-Verlag, 2013.

[79] T. Korak and T. Plos. EM Leakage of RFID Devices - Comparison of Two Measurement Approaches. In *The 9th International Conference on Availability, Reliability and Security (ARES 2014), Fribourg, Swizerland, September, 2014, Proceedings*, pages 120–125. IEEE, 2014.

[80] T. Korak, T. Plos, and M. Hutter. Attacking an AES-enabled NFC Tag - Implications from Design to a Real-World Scenario. In Springer-Verlag, editor, *3rd International Workshop on Constructive Side-Channel Analysis and Secure Design, Darmstadt, Germany, 3. - 4. May, 2012, Proceedings.*, pages 17 – 32, 2012.

[81] T. Korak, T. Plos, and A. Zankl. Minimizing the Costs of Side-Channel Analysis Resistance Evaluations in Early Design Steps. In *The 8th International Conference on Availability, Reliability and Security (ARES 2014), Regensburg, Germany, September 2-6, 2013, Proceedings*, pages 169–177. IEEE, 2013.

[82] T. Korak and L. Wilfinger. Handling the NDEF Signature Record Type in a Secure Manner. In *IEEE International Conference on RFID-Technologies and Applications (RFID-TA), Nice, France, November 05-07, 2012, Proceedings*, pages 107–112, November 2012.

[83] E. Lee. NFC Proxy. Available online at http://sourceforge.net/p/nfcproxy/wiki/Home/, 2012.

[84] H. Li. Refractive Index of Silicon and Germanium and its Wavelength and Temperature Derivatives. *ICON*, 5:9, 1979.

[85] V. Lomné, E. Prouff, M. Rivain, T. Roche, and A. Thillard. How to estimate the success rate of higher-order side-channel attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, pages 35–54, 2014.

[86] S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks – Revealing the Secrets of Smart Cards*. Springer, 2007. ISBN 978-0-387-30857-9.

[87] H. Martin, T. Korak, E. S. Millan, and M. Hutter. Fault Attacks on STRNGs: Impact of Glitches, Temperature, and Underpowering on Randomness. *IEEE Transactions on Information Forensics and Security*, 10:266 – 277, 2014.

[88] G. McGraw. *Software Security: Building Security In*. Addison-Wesley Professional, 2006. ISBN 9780321356703.

[89] M. Medwed, F.-X. Standaert, J. Großschädl, and F. Regazzoni. Fresh Re-Keying: Security Against Side-Channel and Fault Attacks for Low-Cost Devices. In *Progress in Cryptology–AFRICACRYPT 2010*, pages 279–296. Springer, 2010.

[90] M. Medwed, F.-X. Standaert, and A. Joux. Towards Super-Exponential Side-Channel Security with Efficient Leakage-Resilient PRFs. In E. Prouff and P. Schaumont, editors, *Cryptographic Hardware and Embedded Systems CHES 2012*, volume 7428 of *Lecture Notes in Computer Science*, pages 193–212. Springer Berlin Heidelberg, 2012.

[91] T. S. Messerges, E. Dabbish, R. H. Sloan, et al. Examining Smart-Card Security under the Threat of Power Analysis Attacks. *Computers, IEEE Transactions on*, 51(5):541–552, 2002.

[92] Microchip Technology Inc. PIC16F84 Data Sheet, 2001.

[93] A. Moradi and G. Hinterwälder. Side-Channel Security Analysis of Ultra-Low-Power FRAM-based MCUs. In *COSADE*, pages 175–190. Springer, April 2014.

[94] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang. Pushing the Limits: A Very Compact and a Threshold Implementation of AES. In *Advances in Cryptology - EUROCRYPT 2011, 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, 2011.

[95] N. Moro, A. Dehbaoui, K. Heydemann, B. Robisson, and E. Encrenaz. Electromagnetic Fault Injection: Towards a Fault Model on a 32-bit Microcontroller. In *FDTC, 2013*, pages 77–88. IEEE, 2013.

[96] M. Muehlberghuber, F. K. Gurkaynak, T. Korak, P. Dunst, and M. Hutter. Red Team vs. Blue Team Hardware Trojan Analysis: Detection of a Hardware Trojan on an Actual ASIC. In ACM, editor, *Hardware and Architectural Support for Security and Privacy - HASP 2013, Second Workshop, Tel-Aviv, Israel, June 23, 2013, Proceedings.*, pages 1 – 8. ACM, 2013.

[97] M. Muehlberghuber, T. Korak, M. Hutter, and P. Dunst. Towards Evaluating DPA Countermeasures for Keccak on a Real ASIC. In *Constructive Side-Channel Analysis and Secure Design, The sixth International Workshop on*. Springer, 2015. in press.

[98] T. Müller and M. Spreitzenbarth. FROST - Forensic Recovery of Scrambled Telephones. In M. Jacobson, M. Locasto, P. Mohassel, and R. Safavi-Naini, editors, *ACNS 2013, Banff, AB, Canada.*, volume 7954, pages 373–388, 2011.

[99] National Institute of Standards and Technology (NIST). Cryptographic Hash Algorithm Competition Website. `http://csrc.nist.gov/groups/ST/hash/sha-3`.

[100] S. Nikova, V. Rijmen, and M. Schläffer. Secure Hardware Implementation of Non-linear Functions in the Presence of Glitches. In *Information Security and Cryptology - ICISC, Seoul, Korea, December 3-5*, pages 218–234, 2008.

[101] K. Nohl. Cryptanalysis of Crypto-1. Computer Science Department University of Virginia, White Paper, 2008.

[102] NXP. LPC1110/11/12/13/14/15 Product Data Sheet. Available online at `http://www.nxp.com/documents/data_sheet/LPC111X.pdf`, December 2013.

[103] OpenCores.org. OpenMSP430. Available online at `http://opencores.org/project,openmsp430`, June 2009.

[104] Y. Oren, D. Schirman, and A. Wool. Range Extension Attacks on Contactless Smart Cards. In *Computer Security–ESORICS 2013*, pages 646–663. Springer, 2013.

[105] Y. Oren and A. Shamir. Remote Password Extraction from RFID Tags. *IEEE Transactions on Computers*, 56(9):1292–1296, September 2007.

[106] D. Oswald and C. Paar. Breaking Mifare DESFire MF3ICD40: Power Analysis and Templates in the Real World. In *CHES 2011*, volume 6917/2011 of *LNCS*, 2011.

[107] P. Pessl and M. Hutter. Pushing the Limits of SHA-3 Hardware Implementations to Fit on RFID. In *CHES 2013*, volume 8086, pages 126–141. Springer, 2013.

[108] T. Plos, M. Aigner, T. Baier, M. Feldhofer, M. Hutter, T. Korak, and E. Wenger. Semi-Passive RFID Development Platform for Implementing and Attacking Security Tags. *International Journal of RFID Security and Cryptography [Elektronische Ressource]*, 1:16 – 24, 2012.

[109] T. Plos and C. Maierhofer. On Measuring the Parasitic Backscatter of Sensor-enabled UHF RFID Tags. In *ARES 2012, Prague, Czech Republic, August 20-24, 2012, Proceedings*, pages 38–46. IEEE, August 2012.

[110] A. Y. Poschmann. *Lightweight Cryptography - Cryptographic Engineering for a Pervasive World.* PhD thesis, Faculty of Electrical Engineering and Information Technology, Ruhr-University Bochum,Germany, February 2009.

[111] E. Prouff, M. Rivain, and R. Bévan. Statistical Analysis of Second Order Differential Power Analysis. *Computers, IEEE Transactions on*, 58(6):799–811, 2009.

[112] J.-J. Quisquater and D. Samyde. Eddy Current for Magnetic Analysis with Active Sensor. In *Conference on Research in SmartCards (E-Smart'02), Nice, France.*, pages 185–194. UCL, September 2002.

[113] S. U. Rehman, M. Bilal, B. Ahmad, K. M. Yahya, A. Ullah, and O. U. Rehman. Comparison Based Analysis of Different Cryptographic and Encryption Techniques Using Message Authentication Code (MAC) in Wireless Sensor Networks (WSN). *arXiv preprint arXiv:1203.3103*, 2012.

[114] S. Rinne, T. Eisenbarth, and C. Paar. Performance Analysis of Contemporary Light-Weight Block Ciphers on 8-bit Microcontrollers. Available online at `http://www.crypto.ruhr-uni-bochum.de/imperia/md/content/texte/publications/conferences/lw_speed2007.pdf`, June 2007.

[115] D. Samyde, S. P. Skorobogatov, R. J. Anderson, and J.-J. Quisquater. On a New Way to Read Data from Memory. In *IEEE Security in Storage Workshop (SISW02)*, pages 65–69. IEEE Computer Society, 2002.

[116] J.-M. Schmidt and C. Herbst. A Practical Fault Attack on Square and Multiply. In *FDTC, 2008*, pages 53–58. IEEE, 2008.

[117] J.-M. Schmidt and M. Hutter. Optical and EM Fault-Attacks on CRT-based RSA: Concrete Results. In K. C. Posch and J. Wolkerstorfer, editors, *Proceedings of Austrochip 2007, October 11, 2007, Graz, Austria*, pages 61–67. Verlag der Technischen Universität Graz, October 2007. ISBN 978-3-902465-87-0.

[118] J.-M. Schmidt, M. Hutter, and T. Plos. Optical Fault Attacks on AES: A Threat in Violet. In D. Naccache and E. Oswald, editors, *Fault Diagnosis and Tolerance in Cryptography, Sixth International Workshop, FDTC 2009, Lausanne, Switzerland ? September 6, 2009, Procceedings*, pages 13–22. IEEE-CS Press, September 2009.

[119] K. Schramm and C. Paar. Higher Order Masking of the AES. In *CT-RSA 2006*, volume 3860 of *LNCS*, pages 208–225. Springer, 2006.

[120] N. Selmane, S. Guilley, and J.-L. Danger. Practical Setup Time Violation Attacks on AES. In *Dependable Computing Conference, 2008. EDCC 2008. Seventh European*, pages 91–96. IEEE, 2008.

[121] Sergei Skorobogatov. Optical Fault Masking Attacks. In *Fault Diagnosis and Tolerance in Cryptography*, 2010.

[122] R. Silberschneider, T. Korak, and M. Hutter. Access Without Permission: A Practical RFID Relay Attack . In *Austrochip 2013, 21st Austrian Workshop on Microelectronics, Linz, Austria, October 10, 2013, Proceedings*, pages 59 – 64, 2013.

[123] SJB Research Ltd. NFC World. Available online at http://www.nfcworld.com.

[124] S. Skorobogatov. Low Temperature Data Remanence in Static RAM. Technical report, University of Cambridge Computer Laboratory, June 2002.

[125] S. Skorobogatov and C. Woods. In the Blink of an Eye: There Goes your AES Key. *IACR Cryptology ePrint Archive*, 2012:296, 2012.

[126] S. P. Skorobogatov. *Semi-invasive attacks - A new approach to hardware security analysis*. PhD thesis, University of Cambridge - Computer Laboratory, 2005.

[127] S. P. Skorobogatov. Optically Enhanced Position-Locked Power Analysis. In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems – CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, volume 4249 of *Lecture Notes in Computer Science*, pages 30–45. Springer, 2006.

[128] S. P. Skorobogatov and R. J. Anderson. Optical Fault Induction Attacks. In B. S. Kaliski Jr., Ç. K. Koç, and C. 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 2–12. Springer, 2003.

[129] O. Söll, T. Korak, M. Muehlberghuber, and M. Hutter. EM-Based Detection of Hardware Trojans on FPGAs. In IEEE, editor, *IEEE International Symposium on Hardware-Oriented Security and Trust – HOST 2014, Arlington, Virginia, USA, May 6-7, 2014, Proceedings*, pages 84 – 87. IEEE, 2014.

[130] Y. Souissi, S. Bhasin, S. Guilley, M. Nassar, and J.-L. Danger. Towards Different Flavors of Combined Side Channel Attacks. In O. Dunkelman, editor, *CT-RSA 2012*, pages 245 —259. Springer Berlin Heidelberg, March 2012.

[131] R. Specht, J. Heyszl, M. Kleinsteuber, and G. Sigl. Improving Non-profiled Attacks on Exponentiations Based on Clustering and Extracting Leakage from Multi-channel High-Resolution EM Measurements. In *COSADE*, pages 3–19. Springer, 2014.

[132] L. Sportiello and A. Ciardulli. Long Distance Relay Attacks. In *The 9th Workshop on RFID Security*, 2013.

[133] R. Spreitzer and B. Gérard. Towards More Practical Time-Driven Cache Attacks. In *Information Security Theory and Practice. Securing the Internet of Things*, pages 24–39. Springer, 2014.

[134] F.-X. Standaert, T. G. Makin, 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.

[135] M. Taha and P. Schaumont. Side-Channel Analysis of MAC-Keccak. In *Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on*, pages 125–130. IEEE, 2013.

[136] Texas Instruments. MSP Ultra-Low-Power microcontrollers (MCUs) from Texas Instruments (TI). Available online at `http://www.ti.com/lsds/ti/microcontrollers_16-bit_32-bit/msp/ultra-low_power/overview.page`, March 2015.

[137] Texas Instruments. MSP430FR59xx Mixed-Signal Microcontroller Online datasheet, 2015. [Online; accessed 24-March-2015].

[138] P. Thevenon, O. Savry, and S. Tedjini. On the Weakness of Contactless Systems Under Relay Attacks. In *19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–5, 2011.

[139] E. Thomas, von Maurich Ingo, and Y. Xin. Faster Hash-based Signatures with Bounded Leakage. In *SAC*, page 12. ACM, 2013.

[140] S. Tillich, C. Herbst, and S. Mangard. Protecting AES Software Implementations on 32-bit Processors against Power Analysis. In *ACNS 2007*, volume 4521 of *LNCS*, pages 141–157. Springer, June 2007.

[141] E. Trichina. Multi-Fault Laser Attacks on Protected CRT RSA. Invited Talk - FDTC 2010, 2010.

[142] Y.-T. Tsou, C.-S. Lu, and S.-Y. Kuo. MoteSec-Aware: A Practical Secure Mechanism for Wireless Sensor Networks. *Wireless Communications, IEEE Transactions on*, 12(6):2817–2829, 2013.

[143] P. Tuyls and L. Batina. RFID-Tags for Anti-counterfeiting. In *Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings*, 2006.

[144] J. Waddle and D. Wagner. Towards Efficient Second-Order Power Analysis. In M. Joye and J.-J. 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 1–15. Springer, 2004.

[145] E. Wenger, T. Korak, and M. Kirschbaum. Analyzing Side-Channel Leakage of RFID-Suitable Lightweight ECC Hardware. In *RFID Security - RFIDSec 2013, 9th Workshop, Graz, Austria, July 9-11, 2013, Proceedings*, Lecture Notes in Computer Science, pages 128 – 144. Springer, 2013.

[146] J. Westhues. Proxmark.org A Radio Frequency IDentification Tool. Available online at http://www.proxmark.org/, 2013.

[147] C. Whitnall and E. Oswald. A Comprehensive Evaluation of Mutual Information Analysis Using a Fair Evaluation Framework. In *LNCS*, volume 6841/2011, pages 316–334, 2011.

[148] D. M. Xiaofei Guo and R. Karri. Provably Secure Concurrent Error Detection Against Differential Fault Analysis. Cryptology ePrint Archive, Report 2012/552, 2012. `http://eprint.iacr.org/`.

[149] S.-M. Yen and M. Joye. Checking Before Output May Not be Enough Against Fault-Based Cryptanalysis. *Computers, IEEE Transactions on*, 49(9):967–970, 2000.

[150] L. Zussa, J.-M. Dutertre, J. Clediere, and A. Tria. Power Supply Glitch Induced Faults on FPGA: An In-depth Analysis of the Injection Mechanism. In *On-Line Testing Symposium (IOLTS), 2013 IEEE 19th International*, pages 110–115. IEEE, 2013.

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am …… ……..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…… ……..
date (signature)