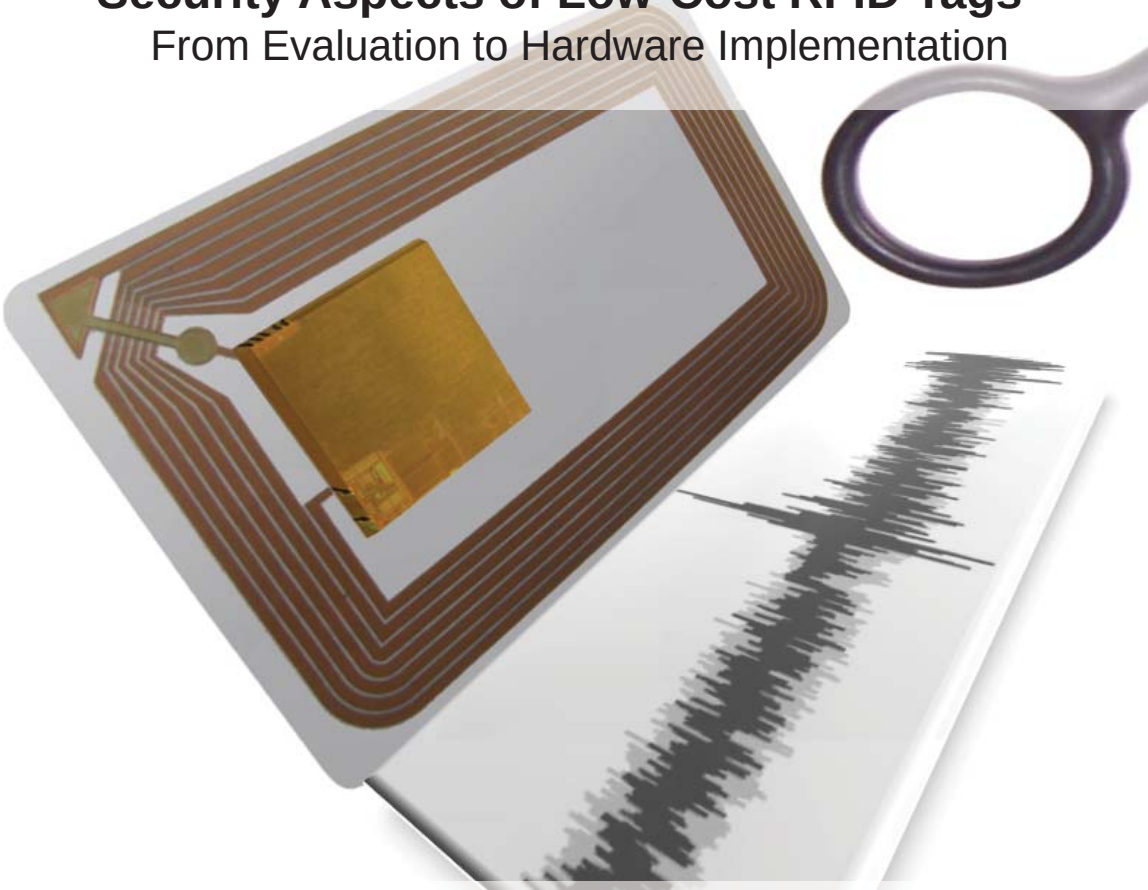
Graz University of Technology
Faculty of Computer Science
Institute for Applied Information Processing and Communications
IAIK

Security Aspects of Low-Cost RFID Tags –
From Evaluation to Hardware Implementation



Assessors:
Prof. Dr. Karl Christian Posch
Prof. Dr. Kerstin Lemke-Rust

September 2011

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

by **Thomas Plos**

Security Aspects of Low-Cost RFID Tags – From Evaluation to Hardware Implementation

by

Thomas Plos

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

Assessors

Prof. Dr. Karl Christian Posch (TU Graz, Austria)
Prof. Dr. Kerstin Lemke-Rust (UoAS BRS, Germany)

September 2011

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

Abstract

Radio-frequency identification (RFID) technology is the enabler for applications like the future Internet of Things (IoT). Especially passive RFID tags that are cheap in price will we used in the future IoT. With the IoT, new applications will arise where security will play an important role. In this work we focus on security aspects that are important for designing the next generation of passive low-cost tags.

In the first part of this work we evaluate the susceptibility of current low-cost tags against implementation attacks. We conduct side-channel analysis (SCA) attacks as well as fault-analysis attacks on commercially available RFID tags from various tags vendors. All evaluated tags have shown a vulnerability to implementation attacks. This emphasizes that integrating proper countermeasures to low-cost tags is necessary when adding security to them. We further analyze the effectiveness of two countermeasures that aim for protecting low-cost tags from SCA attacks: randomizing the execution order of a cryptographic algorithm and detaching the power supply.

In the second part of this work we present a flexible tag architecture that bases on a low-resource 8-bit microcontroller. The flexible architecture allows to efficiently handle complex protocol tasks on low-cost tags. We further show that it is advantageous to reuse the microcontroller for computing cryptographic algorithms on it. By using the microcontroller for both protocol handling and computing cryptographic algorithms, costly resources like memory can be easily reused. Our results clearly point out that this combined approach is even more efficient in terms of additional hardware costs than using dedicated hardware coprocessors that are optimized for low chip area.

Acknowledgements

I would like to thank all the people who supported me during my research within the last years and with whom I had the pleasure to work with. Especially, I want to thank my parents and my girlfriend Christiane. I am also grateful to my good friends Mario and Mihai for their moral support over all the years.

I would like to thank the advisor of this thesis Karl Christian Posch for his guidance through my PhD and for the interesting discussions. A dedicated thanks goes to my assessor Kerstin Lemke-Rust for the valuable comments and for taking the time to travel to Graz. I also want to thank the former group leader Manfred Aigner as well as the current group leader Jörn-Marc Schmidt for decoupling us from management and administrative tasks so that we could focus on our research work.

I am grateful to all people from IAIK and especially the SEnSE group for the good cooperation and the nice working environment. In particular I would like to thank my colleagues Michael Hutter, Jörn-Marc Schmidt, Mario Kirschbaum, and Martin Feldhofer for their help and the fruitful discussions about technical and non-technical topics. Finally I want to thank my coauthors Michael Hutter, Martin Feldhofer, Jörn-Marc Schmidt, Mario Kirschbaum, Stefan Tillich, Christoph Herbst, Marcel Medwed, Hannes Groß, Erich Wenger, Manfred Aigner, and Alexander Szekely who have contributed to my work.

Thomas Plos
Graz, September 2011

Table of Contents

List of Publications

1. Thomas Plos. Susceptibility of UHF RFID Tags to Electromagnetic Analysis. In Tal Malkin, editor, *Topics in Cryptology - CT-RSA 2008, The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008, Proceedings*, volume 4964 of *Lecture Notes in Computer Science*, pages 288–300. Springer, April 2008.

2. Thomas Plos, Michael Hutter, and Martin Feldhofer. Evaluation of Side-Channel Preprocessing Techniques on Cryptographic-Enabled HF and UHF RFID-Tag Prototypes. In Sandra Dominikus, editor, *Workshop on RFID Security 2008 (RFIDSec08), July 9-11, Budapest, Hungary*, pages 114–127, July 2008.

3. Michael Hutter, Jörn-Marc Schmidt, and Thomas Plos. RFID and its Vulnerability to Faults. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2008, 10th International Workshop, Washington DC, USA, August 10-13, 2008, Proceedings*, volume 5154 of *Lecture Notes in Computer Science*, pages 363–379. Springer, August 2008.

4. Thomas Plos, Michael Hutter, and Christoph Herbst. Enhancing Side-Channel Analysis with Low-Cost Shielding Techniques. In Michael Sams Christoph Lackner, Timm Ostermann and Ronald Spilka, editors, *Proceedings of Austrochip 2008, October 8, 2008, Linz, Austria*, pages 90–95, October 2008. ISBN 978-3-200-01330-8.

5. Thomas Plos. Evaluation of the Detached Power Supply as Side-Channel Analysis Countermeasure for Passive UHF RFID Tags. In Marc Fischlin, editor, *Topics in Cryptology - CT-RSA 2009, The Cryptographers' Track at the RSA Conference 2009, San Francisco, CA, USA, April 20-24, 2009, Proceedings*, volume 5473 of *Lecture Notes in Computer Science*, pages 444–458. Springer, April 2009.

6. Michael Hutter, Jörn-Marc Schmidt, and Thomas Plos. Contact-Based Fault Injections and Power Analysis on RFID Tags. In *European Conference on Circuit Theory and Design 2009, ECCTD*, 2009.

7. Thomas Plos, Michael Hutter, and Martin Feldhofer. On Comparing Side-Channel Preprocessing Techniques for Attacking RFID Devices. In

Heung Youl Youm and Moti Yung, editors, *Information Security Applications, 10th International Workshop, WISA 2009, Busan, Korea, August 25-27, 2009, Revised Selected Papers*, volume 5932 of *Lecture Notes in Computer Science*, pages 163–177. Springer, December 2009.

8. Jörn-Marc Schmidt, Michael Hutter, and Thomas Plos. Optical Fault Attacks on AES: A Threat in Violet. In David Naccache and Elisabeth 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.

9. Jörn-Marc Schmidt, Thomas Plos, Mario Kirschbaum, Michael Hutter, Marcel Medwed, and Christoph Herbst. Side-Channel Leakage Across Borders. In Dieter Gollmann and Jean-Louis Lanet, editors, *Smart Card Research and Advanced Applications 9th IFIP WG 8.8/11.2 International Conference, CARDIS 2010, April 13-16, 2010, Passau, Germany, Proceedings*, Lecture Notes in Computer Science, pages 36–48. Springer, April 2010.

10. Michael Hutter, Martin Feldhofer, and Thomas Plos. An ECDSA Processor for RFID Authentication. In Siddika Berna Ors 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 189–202. Springer, 2010.

11. Thomas Plos, Hannes Groß, and Martin Feldhofer. Implementation of Symmetric Algorithms on a Synthesizable 8-Bit Microcontroller Targeting Passive RFID Tags. In Alex Biryukov, Guang Gong, and Douglas Stinson, editors, *17th Annual Workshop on Selected Areas in Cryptography - SAC 2010, Waterloo, Canada, August 12-13, 2010, Proceedings*, volume 6544 of *Lecture Notes in Computer Science*, pages 114–129. Springer, 2010.

12. Stefan Tillich, Martin Feldhofer, Mario Kirschbaum, Thomas Plos, Jrn-Marc Schmidt, and Alexander Szekely. Uniform Evaluation of Hardware Implementations of the Round-Two SHA-3 Candidates. August 2010.

13. Stefan Tillich, Martin Feldhofer, Mario Kirschbaum, Thomas Plos, Jrn-Marc Schmidt, and Alexander Szekely. Hardware Implementations of the Round-Two SHA-3 Candidates: Comparison on a Common Ground. In *Proceedings of Austrochip 2010, October 6, 2010, Villach, Austria*, pages 43–48, October 2010. ISBN 978-3-200-01945-4.

14. Martin Feldhofer, Manfred Josef Aigner, Michael Hutter, Thomas Plos, Erich Wenger, and Thomas Baier. Semi-Passive RFID Development Platform for Implementing and Attacking Security Tags. In *Workshop on RFID / USN Security and Cryptography - RISC 2010, November 9-10, London, UK, 2010.*, 2010.

15. Michael Hutter, Thomas Plos, and Martin Feldhofer. On the Security of RFID Devices Against Implementation Attacks. *International Journal of Security and Networks 2010*, 5(2/3):106–118, 2010.

16. Thomas Plos and Martin Feldhofer. Hardware Implementation of a Flexible Tag Platform for Passive RFID Devices. In *Proceedings of the 14th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2011), Oulu, Finland, August, 2011, Proceedings*. IEEE Computer Society, August 2011.

17. Thomas Plos and Martin Feldhofer. Analyzing the Hardware Costs of Different Security-Layer Variants for a Low-Cost RFID Tag. In *Security and Privacy in Communication Networks - 7th International ICST Conference, SecureComm 2011, London, UK, September 7-9, 2011, Proceedings*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer, 2011.

List of Tables

List of Figures

Acronyms

ACC	Accumulator
AES	Advanced Encryption Standard
ALU	Arithmetic-Logic Unit
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
APDU	Application Protocol Data Unit
BIST	Built-In Self Test
CISC	Complex Instruction-Set Computer
CMOS	Complementary Metal-Oxide-Semiconductor
CRC	Cyclic Redundancy Check
CRT	Chinese Remainder Theorem
CRYPTA	Cryptographic Protected Tags for new RFID Applications
DEMA	Differential Electromagnetic Analysis
DES	Data Encryption Standard
DFA	Differential Frequency Analysis
DPA	Differential Power Analysis
DRP	Dual-Rail Precharge
DST	Digital Signature Transponder
DUT	Device Under Test
EAN	European Article Number
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic-Curve Digital Signature Algorithm
EDA	Electronic Design Automation
EEPROM	Electrically Erasable Programmable ROM
EM	Electromagnetic
EPC	Electronic Product Code
EPROM	Erasable Programmable ROM
FFT	Fast Fourier Transform
FIB	Focused Ion Beam
FIFO	First-In First-Out
FIPS	Federal Information Processing Standards
FL	Framing Logic
FPGA	Field-Programmable Gate Array
FSM	Finite State Machine
GCD	Greatest Common Divisor
GE	Gate Equivalent

HDL	Hardware Description Language
HF	High Frequency
IC	Integrated Circuit
iMDPL	Improved MDPL
ISE	Instruction-Set Extension
ISO	International Organization for Standardization
ISP	In-System Programming
ISS	Instruction-Set Simulator
IV	Initial Value
IoT	Internet of Things
JTAG	Joint Test Action Group
LF	Low Frequency
MDPL	Masked Dual-Rail Precharge Logic
NFC	Near-Field Communication
NIST	National Institute of Standards and Technology
NOP	No Operation
NVM	Non-Volatile Memory
OBIC	Optical Beam Induced Current
PCB	Printed Circuit Board
PDA	Personal Digital Assistant
PET	Polyethylene Terephthalate
POR	Power-On-Reset
PPS	Protocol and Parameter Selection
RAM	Random Access Memory
RATS	Request for Answer to Select
RF	Radio Frequency
RFID	Radio-Frequency Identification
RISC	Reduced Instruction-Set Computer
ROM	Read-Only Memory
RSA	Rivest-Shamir-Adleman
RSL	Random Switching Logic
RTL	Register-Transfer Level
SABL	Sense Amplifier Based Logic
SCA	Side-Channel Analysis
SEA	Scalable Encryption Algorithm
SEMA	Simple Electromagnetic Analysis
SHA	Secure Hash Algorithm
SNR	Signal-to-Noise Ratio
SPA	Simple Power Analysis
SPN	Substitution-Permutation Network
TRNG	True Random-Number Generator
UCC	Uniform Code Council
UHF	Ultra-High Frequency
UID	Unique Identifier
USB	Universal Serial Bus

VLSI	Very Large Scale Integration
WDDL	Wave Dynamic Differential Logic
WTX	Waiting Time Extension
XTEA	Extended Tiny Encryption Algorithm

1 Introduction

The technological advances of the last decades have strongly influenced our daily life. We are surrounded by devices that are equipped with small integrated circuits that have computing capability. Integrated circuits are not only found in personal computers and laptops, but also in cars, domestic appliances, and in any device that has communication functionality. Computers that once filled a whole room fit now in the palm of your hand and you can easily carry them around in your pocket. Computers even no longer look like computers since they are inherently integrated into the devices.

Integrated circuits are manufactured from silicon as so-called microchips. Continuous migration to more-advanced manufacturing techniques allows increasing the functionality of the microchips and lowering their power consumption. Reducing the power consumption of integrated circuits has pushed the development of mobile devices and contactless communication techniques. An important contactless communication technique is radio-frequency identification (RFID) technology, which is used for example in ticketing, electronic passports, logistics, and car immobilizers. Even the latest generation of smart phones has integrated RFID functionality, which emphasizes the relevance of this technology.

In an RFID system a reader and a tag communicate contactlessly by means of a radio-frequency (RF) field. The tag is a small microchip attached to an antenna. More than 2 billion tags have been sold in 2010. Most of them are so-called passive tags that directly receive their power from the RF field. The functionality of passive tags ranges from contactless smart cards with cryptographic coprocessors and large memories to low-cost tags that only provide a unique identifier and have very limited resources. Especially low-cost tags that can be produced at high volume are the enabler for the future Internet of Things

(IoT). The vision of this future IoT is to provide communication capabilities to every object by attaching an RFID tag to it. With the IoT, many new applications will arise that have increased demands concerning functionality and security of the underlying RFID system. Since this concerns all components of the RFID system, also the tags will have to integrate more advanced functionality and security.

Integrating additional functionality to passive low-cost tags is challenging because of two constraints: chip size and power consumption. Chip size influences the tag price and has to be kept low. Power consumption is limited since passive tags are supplied by the RF field of the reader. Current low-cost tags are implemented as hardwired finite-state machines (FSM) to minimize chip size and power consumption. This approach is time consuming and gets even inefficient when functionality and control complexity of tags increases. Hence, programmable solutions based on a simple microcontroller are advantageous to cope with increased complexity of the tags. Moreover, using a simple microcontroller also eases the integration of security into low-cost tags.

A lot of effort has been made by the research community to bring state-of-the-art cryptographic security to RFID tags. Prominent examples among others are symmetric-key schemes like the Advanced Encryption Standard (AES) [55, 69, 127], or public-key schemes like Elliptic Curve Cryptography (ECC) [16, 20, 71, 183]. However, low-cost tags currently available on the market neglect to consider state-of-the-art cryptographic security. They rather rely on weak security measures like passwords or make use of proprietary cryptographic algorithms that are not widely approved and kept secret instead. Several incidents of the last years have shown that proprietary cryptographic algorithms can easily be broken when they get public. The most famous examples are the reverse engineering of the CRYPTO 1 algorithm in Philips' Mifare tags [135] and the breaking of the DST40 cipher in Texas Instrument's Digital Signature Transponders (DST) [27]. Moreover, weak structures of proprietary algorithms are an easy target for algebraic attacks, as it has been shown in case of the Hitag 2 cipher [40]. Hence, using cryptographic algorithms that provide state-of-the-art security is inevitable for protecting RFID systems.

State-of-the-art cryptographic algorithms are secure in a mathematical and cryptanalytical sense. However, also the devices and systems that implement the algorithms need to be secure. Weaknesses at implementation level can be exploited to dramatically lower the effort that is necessary to deduce secret information from a device (*e.g.* the secret key). Techniques that use such weaknesses are called implementation attacks. When implementing cryptographic algorithms on RFID tags, appropriate countermeasures have to be integrated to make implementation attack less effective. Further, it is necessary to evaluate the susceptibility of RFID tags against implementation attacks to get a better understanding of the potential threat scenarios.

In this thesis we address two topics that are very important for designing future low-cost RFID tags. First, we evaluate the vulnerability of low-cost tags against implementation attacks. We present successful side-channel analysis

(SCA) attacks and fault-analysis attacks on commercially available RFID tags from various tag vendors. Our results clearly illustrate that low-cost RFID tags are susceptible to implementation attacks similar to contact-based devices. Hence, when implementing cryptographic algorithms on low-cost tags, countermeasures have to be integrated. This has motivated us to also evaluate the effectiveness of countermeasures like shuffling of operations and detaching the power supply that are potentially interesting for protecting RFID tags against SCA attacks.

The second topic covers hardware-implementation aspects of low-cost RFID tags. We suggest a flexible tag architecture based on a low-resource microcontroller. The flexible architecture addresses the demand of future low-cost tags in the IoT that have to deal with increased protocol complexity and advanced tag functionality. Our architecture is easily adaptable for integrating new functionality and fulfils the fierce requirements of passive low-cost tags in terms of power consumption and chip area. We further show that the microcontroller of the flexible tag architecture can be used for both protocol handling and execution of cryptographic algorithms. This combined approach is highly advantageous as it allows to reuse expensive hardware resources like memory. In that way, we can implement cryptographic algorithms like the AES with less resource usage than dedicated low-resource hardware modules.

1.1 Contribution of this Thesis

We have started our research on the topic security in RFID systems with a master thesis in [146]. A security-enabled semi-passive tag prototype for the ultra-high frequency (UHF) range has been implemented. The prototype allows not only extension of protocols but it is also very practical for evaluating the security of existing RFID systems. A description of semi-passive high frequency (HF) and UHF tag prototypes is given in [54]. This is a joint work with Martin Feldhofer, Manfred Aigner, Michael Hutter, Erich Wenger, and Thomas Baier.

Using the semi-passive UHF tag prototype has allowed us to mount first differential electromagnetic analysis (DEMA) attacks on commercially available UHF RFID tags [147]. Passive UHF RFID tags suffer from so-called parasitic-backscatter attacks that allow remote attacks from one meter and more.

In a joint work with Michael Hutter and Jörn-Marc Schmidt, the vulnerability of commercially available HF and UHF RFID tags against fault analysis has been evaluated. Fault-injection techniques based on temporarily antenna-tearing, electromagnetic interferences, and optical induction have been applied. First results on successful fault attacks on RFID tags have been published in [80].

Contact-based fault injections and power-analysis techniques on RFID tags have been presented in [81]. This is also a work that has been carried out together with Michael Hutter and Jörn-Marc Schmidt. Further results on implementation attacks on RFID tags have been presented with Michael Hutter and Martin Feldhofer in a Journal paper in [79].

The effectiveness of randomization as a side-channel analysis (SCA) counter-

measure for passive RFID tags has been evaluated in a joint work with Michael Hutter and Martin Feldhofer in [152, 153]. Several preprocessing techniques such as differential frequency analysis and windowing have been applied to enhance SCA attacks on randomization-based countermeasures. Another SCA countermeasure that has been analyzed is the detached power supply. The countermeasure is intended for protecting passive UHF RFID tags from remote attacks. Results of this work have been published in [148].

A flexible tag platform that is based on a low-resource 8-bit microcontroller has been published in [150]. The tag platform aims for low resource usage and has been developed together with Martin Feldhofer. Using this flexible tag platform allows efficiently integrating additional functionality to future low-cost RFID tags. The flexible tag platform has been used within the CRPYTA prototype chip, which is a near-field communication (NFC) enabled tag with advanced file-management functionality and security features. The prototype chip works fully passive and contains besides the flexible tag platform an analog front-end, an EEPROM, and a cryptographic unit. Analog front-end and EEPROM have been designed by the CRYPTA project partner Austriamicrosystems. The cryptographic unit that contains symmetric-key as well as public-key cryptography has been implemented by Michael Hutter. A paper that describes the cryptographic unit of the prototype chip has been published in [76] together with Michael Hutter and Martin Feldhofer.

The low-resource 8-bit microcontroller from the flexible tag platform has been adapted and used for implementing several symmetric-key algorithms on it. Symmetric-key algorithms such as AES, NOEKEON, SEA, and Trivium have been implement and compared with implementations on other microcontroller platforms. The work has been carried out together with Hannes Groß and Martin Feldhofer. The results have been published in [151] and clearly point out that our low-resource 8-bit microcontroller allows a very compact implementation of the algorithms.

We have integrated the symmetric-key algorithm implementations into the flexible tag platform in [149]. This is a joint work with Martin Feldhofer and illustrates that a combined implementation of high-level protocol handling and computation of cryptographic algorithms on the microcontroller is a highly advantageous approach. Expensive resources like memory that are used by both protocol handling and cryptographic algorithms can be easily reused. This leads to a very compact implementation that even outperforms dedicated hardware implementations of cryptographic algorithms.

A so-called new work-item proposal has been submitted to the Austrian ISO standardization committee. The proposal suggests the integration of a security layer to low-cost UHF RFID tags that operate according to the ISO 18000-6C standard. The proposal has been accepted and has served as starting point for the development of the ISO 29167 standard, which is still in its early stages.

The following publications are beyond the scope of RFID systems. In [153] a low-cost shielding device has been presented that allows to improve SCA attacks in presence of environmental noise. This is a joint work with Michael Hutter

and Christoph Herbst. Optical fault attacks on AES implementations have been published in [164] with Jörn-Marc Schmidt and Michael Hutter. We have evaluated the SCA leakage of I/O pins in [165]. The work has been carried out with Jörn-Marc Schmidt, Mario Kirschbaum, Michael Hutter, Marcel Medwed, and Christoph Herbst. High-speed implementations of the round-two SHA-3 candidates have been published in [181, 182] with Stefan Tillich, Martin Feldhofer, Mario Kirschbaum, Jörn-Marc Schmidt, and Alexander Szekely.

1.2 Organization of this Thesis

We have organized this thesis into two parts. The first part covers Chapter 2-8. After a short introduction to RFID technology, we focus on implementation attacks and the evaluation of countermeasures in context of low-cost RFID tags. The second part involves Chapter 9-12 and addresses hardware-implementation aspects of low-cost RFID tags. In the following we present a short outline of the chapters in this thesis.

Chapter 2 provides a brief introduction to RFID technology. We describe frequency ranges as well as coupling techniques of RFID systems and give details about the functionality of typical RFID tags.

Chapter 3 gives information about implementation attacks and provides the basis for the chapters afterwards. We start with a description of side-channel analysis attacks, followed by an introduction to fault-analysis attacks.

Chapter 4 provides basic information about countermeasures against implementation attacks. First, countermeasures against side-channel analysis attacks that use timing information, power consumption, and electromagnetic emanations are presented. Afterwards, countermeasures against fault-analysis attacks are described.

Chapter 5 presents practical side-channel analysis results of low-cost UHF RFID tags. We give details about UHF RFID tags in general and describe the examined UHF tags as well as the utilized measurement setup. Side-channel analysis results of both a semi-passive UHF tag prototype and commercially available UHF tags from various tag vendors are presented. Attacks are conducted in the near field of the tags and in the far field.

Chapter 6 deals with fault-analysis attacks on low-cost RFID tags. After elaborating fault-analysis techniques that are suitable for passive RFID tags, we give a description of the measurement setups that we have used for fault analysis. Further, results for globally as well as locally induced faults on HF and UHF tags are presented. Temporarily antenna tearing and electromagnetic interferences are used for global fault injections. Optical inductions are used for both global and local fault injections.

Chapter 7 evaluates the effectiveness of randomization as a countermeasure for RFID devices against side-channel analysis attacks. We start with a description of HF and UHF tag prototypes that have been used for the evaluation. Afterwards we deal with noise in SCA measurements, followed by a discussion of techniques that lower the impact of noise and that ease the attacking of hid-

ing countermeasures. We give a description of the deployed measurement setups and present the achieved evaluation results.

Chapter 8 evaluates the suitability of the detached power supply as side-channel analysis countermeasure for passive UHF RFID tags. We start with a description of the principle of the detached power supply and present a practical implementation of it. Afterwards, the measurement setup is elaborated and the side-channel analysis results are given. A simple countermeasure for preventing side-channel leakage at the modulation pin of RFID tags is presented, and the costs for integrating the detached power supply into passive UHF tags is discussed.

Chapter 9 introduces to the design of digital hardware circuits. We describe design cycle, design space, and testability of digital hardware circuits. Afterwards, power consumption and chip-area requirements for hardware circuits used in passive low-cost tags are given.

Chapter 10 describes the hardware implementation of a flexible tag platform that is suitable for passive low-cost tags. We start with an overview of the flexible tag platform and describe its functionality as well as the separation of tasks in hardware and software. The components of the flexible tag platform are described and implementation results are provided, followed by a short presentation of the CRYPTA tag chip that used the flexible platform.

Chapter 11 presents the implementation of symmetric-key algorithms on a low-resource 8-bit microcontroller. Six symmetric-key algorithms are implemented: the Advanced Encryption Standard (AES), NOEKEON, Present, the Scalable Encryption Algorithm (SEA), the Extended Tiny Encryption Algorithm (XTEA), and Trivium. After a short overview of the algorithms, implementation results are provided and discussed with respect to passive RFID tags.

Chapter 12 describes a combined implementation of protocol handling and cryptographic algorithm on a low-resource 8-bit microcontroller. We start with a system overview and introduce three security-layer variants that are implemented on the microcontroller. The security-layer variants base on the symmetric-key algorithms AES and NOEKEON, respectively. After presenting the implementation results of the different security-layer variants, their resource usage is compared.

Chapter 13 finalizes this thesis. We give a short summary of the achieved results, draw conclusions, and discuss open research points.

Part I

Implementation Attacks and Evaluation of Countermeasures in Context of Low-Cost RFID Tags

2 RFID Technology

Radio-frequency identification (RFID) technology is a contactless communication technique that has gained a lot of attention over the last years. Many applications already relay on RFID technology. The most prominent examples among others are: inventory control, pallet tracking, ski ticketing, public transportation, access-control systems, electronic passport, animal identification, anti-theft systems, and immobilizers. The basis for modern RFID technology has been laid by the Swedish engineer Harry Stockmann in 1948. Inspired by radar engineering, he came up with the idea of transmitting data contactlessly by using reflected power [175]. However, it has taken more than forty years until the technological progress has allowed the integration of RFID technology into commercial applications.

This chapter aims to provide a brief introduction to RFID technology. We describe a basic RFID system in Section 2.1. Frequency ranges and coupling techniques used by RFID systems are explained in Section 2.2 and Section 2.3, respectively. A description of the functionality of typical RFID tags is given in Section 2.4.

2.1 Description of a Basic RFID System

A typical RFID system consists of three components: a back-end database, a reader, and one or more tags. As depicted in Figure 2.1, the reader is connected to the back-end database and communicates with the tags contactlessly by means of a radio-frequency (RF) field. The tag is a small microchip attached to an antenna. Data is transmitted from the reader to the tags by modulating the carrier signal of the RF field. Tags demodulate the carrier signal to extract

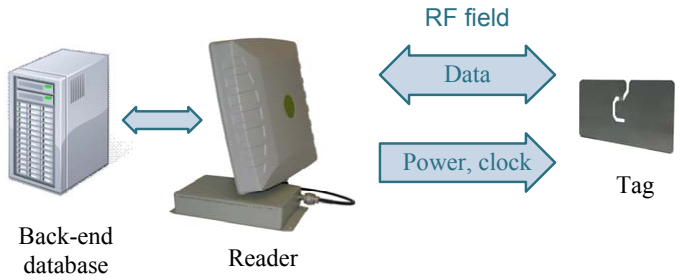


Figure 2.1: Overview of a basic RFID system consisting of a back-end database, a reader, and a tag.

the data and potentially also derive the clock signal from the RF field. Data transmission from tag to reader is done in a similar way. The tags directly modulate the field (load modulation) or change the amount of power that is reflected by their antenna in step with the data (backscatter modulation). Power supply of the tags is either obtained from a battery or from the reader field itself. Active tags are equipped with a battery that allows them to send their response independently of the reader field. Passive tags on the other hand are completely powered by the RF field and require no additional power supply. Semi-passive tags are something between an active tag and a passive tag. They are equipped with a battery like active tags, but use the RF field for sending their response like passive tags. Especially passive tags are widely deployed since they have a simple design and are cheap in price. However, the read range of passive tags is limited because the strength of the RF field rapidly decreases with increasing distance to the reader.

2.2 Frequency Ranges of RFID Systems

RFID systems operate at different frequency ranges starting from 125 kHz up to 5.8 GHz. Exact frequencies can vary from country to country due to local regulations. The frequency of an RFID system has a strong impact on achievable read range, maximum data rate, and size (*i.e.* dimensions) of the tag. Mainly there are four frequency ranges used by RFID systems: low frequency (LF) range, high frequency (HF) range, ultra-high frequency (UHF) range, and microwave range [59].

2.2.1 Low-Frequency Range

LF systems typically operate at frequencies of 125 kHz and 135 kHz. Due to the low frequencies, there are no problems with reflections or absorbtion of the RF signal. This makes LF systems ideal for applications like animal identification. Data rates are rather low and read ranges are less than 1 m.

2.2.2 High-Frequency Range

RFID systems in the HF range are the ones that are most prevalent. The primer operating frequency of HF systems is 13.56 MHz. The higher frequency of the RF signal allows to use also higher data rates (*e.g.* 848 kHz [90]). Read ranges of HF systems are similar to the ones of LF systems. Well-known examples that utilize HF tags are contactless payment systems, ticketing, and library systems.

2.2.3 Ultra-High Frequency Range

UHF systems use frequencies from 860 MHz to 960 MHz. In contrast to HF and LF systems, read ranges of UHF systems are much larger. Typical read ranges of UHF systems are 3 to 8 m. Due to the high frequency of the RF signal, environmental influences such as refraction, absorption, and reflection have to be considered when using UHF systems. Important applications for UHF systems are toll collection, logistics, and supply-chain management.

2.2.4 Microwave Range

The highest frequencies are used by RFID systems in the microwave range. Widely used frequencies are 2.45 GHz and 5.8 GHz. Microwaves have only a wave length of several centimeters, which allows to build tags with antennas of compact size. The data rates and read ranges of microwave systems are similar to the ones of UHF systems. Environmental influences play also an important role in microwave systems and have to be taken into account. Typical applications of microwave systems are toll collection and baggage identification.

2.3 Coupling Methods of RFID Systems

The coupling method of an RFID system is closely related to the utilized frequency. Three coupling methods are used by RFID systems [59]: electric (capacitive) coupling, magnetic (inductive) coupling, and electromagnetic coupling. The coupling method describes how energy is transferred between reader and tag. Electric coupling and magnetic coupling are used by RFID systems with a carrier frequency below 30 MHz (LF and HF systems). Especially magnetic coupling is widely deployed. The tags operate in the so-called near field of the reader at distances of several centimeters (close coupling) up to 1 m (remote coupling). In the near field of the reader antenna, tags have direct influence on the RF signal and can use load-modulation techniques to transmit data. This is typically done by switching a load (*e.g.* resistor) in parallel to the tag antenna. Electromagnetic coupling on the other hand is used by RFID systems with much higher frequencies (UHF and microwave systems). The tags operate in the far field of the reader at distances of several meters. The far field starts at a distance of about $\frac{\lambda}{2\pi}$ from the reader antenna, whereas λ is the wavelength of the RF signal (λ is defined as the speed of light in vacuum divided by the frequency of the RF signal) [59]. In the far field, the RF signal is completely separated from

the reader antenna and propagates as so-called plane wave. Since tags in the far field have no longer direct influence on the RF signal of the reader, techniques from radar engineering are used to transmit data. The technique utilized by electromagnetically coupled tags is called backscatter modulation and changes the amount of power that is reflected by the tag antenna.

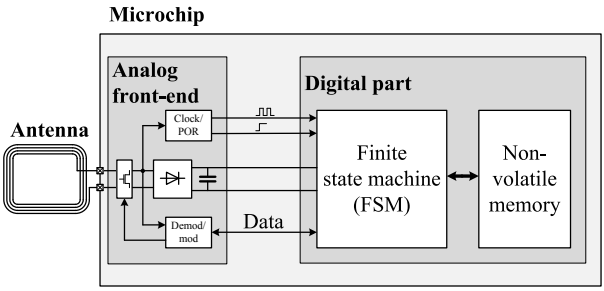


Figure 2.2: Schematic overview of the components of a typical low-cost RFID tag.

2.4 Functionality of RFID Tags

RFID tags are available in different sizes and shapes and provide also different functionality depending on the targeted application. The tag is basically a microchip attached to an antenna. As depicted in Figure 2.2, the microchip consists of an analog front-end and a digital part. The analog front-end is responsible for demodulating the incoming data, extracting the clock signal, and modulating the response data. In case of passive tags, the analog front-end also extracts the power supply from the RF field. The digital part interprets the received data, performs the required actions (*e.g.* write data), and generates appropriate responses. Complexity of the digital part and functionality provided by it largely varies. At the upper end there are contactless smart cards, followed by sensor-enabled tags in the middle, and low-cost tags at the lower end.

Contactless smart cards have integrated powerful microcontrollers (the latest use 32-bit RISC processors) and cryptographic coprocessors that contain a large amount of volatile and non-volatile memory [85, 138, 173]. The coprocessors allow fast execution of cryptographic algorithms. Volatile memory is in the range of tens of kB and non-volatile memory in the range of hundreds of kB. Such tags are expensive in terms of chip-area requirement. Moreover, contactless smart cards provide only short read ranges due to their high power consumption (in the range of milliamps).

Sensor-enabled tags on the other hand have a lower power consumption. They use commercially available microcontrollers like PIC16 from Microchip [122] or MSP430 from Texas Instruments [179] that are optimized for low power consumption. Sensor-enabled tags contain typically a considerable amount of non-volatile memory to store sensor data (*e.g.* flash or EEPROM with several tens of kB) and are often equipped with a battery that allows them to sense data even

in absence of the reader field [3, 119, 126, 198]. However, the resource usage of sensor-enabled tags is still beyond that what is acceptable for low-cost tags.

The digital part of low-cost tags contains only a small memory for storing a unique identifier (UID) and probably some additional configuration data. The architecture of low-cost tags is based on a finite state-machine (FSM) approach that is fixed in hardware (see Figure 2.2). This allows optimizing the tags for low resource usage and low power consumption. Hence, low-cost tags are much cheaper in price than contactless smart cards or sensor-enabled tags and have much lower power requirements (in the range of microamps). This allows them to achieve larger read ranges when they are supplied by the RF field. Such low-cost tags are of special interest for the future internet of things, where tags have to be available in large quantities and need to be competitive in price.

3

Basics of Implementation Attacks

Modern cryptographic algorithms follow Kerckhoffs' principle. This means that the security of an algorithm only relies on the cryptographic key that needs to be kept secret and not on the knowledge of the algorithm itself. Hence, new cryptographic algorithms are immediately published after their design to allow extensive evaluation of their security. An algorithm gets only standardized if no weaknesses are found during its evaluation phase, which typically lasts several years. An actual example is the ongoing SHA-3 competition whose aim is to find a new hash-function standard. The importance of the evaluation phase is underlined by the design flaws that have been found in the MD5 [186] and SHA-1 [185] algorithms several years after their standardization. A flaw in an algorithm can lead to a significant reduction of its security, or even worse, totally break it. Especially in the area of low-cost RFID tags, Kerckhoffs' principle was often ignored in the past by relying on so-called proprietary algorithms that were kept secret and did not undergo a public evaluation phase.

The security of a cryptographic algorithm is determined by the effort (computational power and memory) that is required to break it without knowing its secret key. State-of-the-art algorithms are typically computationally secure, *i.e.* much more time and computing power is needed to break them than there is practically available. However, security on algorithmic level alone is not enough. Also the systems and devices that implement and use the algorithms need to be secure. Hence, weaknesses at implementation level can be exploited to significantly reduce the effort for deducing, for example, the secret key of a device. Techniques that use such weaknesses are called implementation attacks.

At the end of the last century, various cryptographic researchers like Kocher *et al.* [103, 104], Boneh *et al.* [26], Biham *et al.* [22], and Kömmerling *et al.* [105] pointed out that implementation attacks are a serious concern for the security of

cryptographic devices. Their work created a new research field that deals with the implementation security of cryptographic devices. However, implementation attacks are not a new concept, they have their origins about 100 years earlier. A first form of implementation attack was reported from World War I, where the German army eavesdropped the communication of their enemies by measuring the earth-loop current of field-phone lines [106]. Later in 1943, a researcher from Bell Labs noticed by chance that rotor machines that were used for encrypting messages caused voltage spikes during operation. These voltage spikes appeared on a freestanding oscilloscope and allowed to deduce the content of the original message in plain [136]. Another example was reported in 1956, where the clicking sounds of the rotors of a Hagelin ciphering machine were exploited to obtain access to secret information [190]. All these primary implementation attacks were kept secret for a long time and were only known by a small community.

Implementation attacks can be divided into two classes of attacks: passive attacks and active attacks. Passive attacks make typically no modification of the analyzed device. Instead, such attacks only measure some physical leakage of the device—like power consumption, electromagnetic emanation, or timing behavior—or analyze the communication traffic between devices to deduce secret information. Prominent examples for passive attacks are side-channel analysis and logical attacks. Active attacks on the other hand require a manipulation the operating environment of the analyzed device (*e.g.* temperature) or its inputs (clock lines, supply lines), or a modification of the device itself (*e.g.* physically open it). In all cases, the attacks aim to provoke an abnormal behavior (a fault) that reveals some secret information. Fault attacks are typically active attacks (an exception are faults by change caused *e.g.* by natural radiation).

The remainder of this chapter gives more detailed information about side-channel analysis and fault analysis, providing the basis for the chapters afterwards that deal with countermeasures against implementation attacks and practical examples of implementation attacks on RFID devices.

3.1 Side-Channel Analysis

Side-channel analysis (SCA) is a very powerful kind of implementation attack. When applying SCA techniques to a cryptographic device, no evidence is typically left that indicates that such an attack has been conducted. Or even worse, when performing SCA attacks from a distance (*e.g.* of several meters) they can be conducted without being noticed by the owner of the device. Especially RFID devices that receive their power and their data contactlessly from a radio-frequency field can be prone to attacks from a distance. SCA attacks measure a physical property (often also called side channel) of the examined device to reveal secret information. Most-relevant physical properties in that context are timing information, power consumption, and electromagnetic emanation. Other physical properties that could also be used as side channels but which are of less importance are, for example, acoustic emanations [17, 169], temperature [28], or light emissions [45].

In the following, we give deeper insight into SCA attacks. We concentrate on the three most prevalent side-channel sources: timing information, power consumption, and electromagnetic emanations.

3.1.1 SCA Attacks Using Timing Information

The first SCA attack that has used timing information was published by Paul Kocher in 1996 [103]. The work describes how the secret exponent in Diffie-Hellman and RSA implementations can be recovered bit-by-bit when a simple modular exponentiation algorithm is deployed. Kocher's timing attack has been the first published SCA attack. Other timing attacks followed over the years. Dhem *et al.* [44] for example, presented results about a timing attack against an RSA implementation that uses the square-and-multiply algorithm. Handschuh *et al.* [70] conducted a timing attack against the RC5 block cipher. Brumley *et al.* [29] and Aciiçmez *et al.* [6] presented attacks against SSL implementations.

Timing attacks are SCA attacks that use timing information (*i.e.* the execution time) of a cryptographic algorithm as source of physical leakage. When the execution time is not constant but dependent on the input data of the algorithm, an SCA attack can be mounted. A good example for illustration is the aforementioned timing attack of Dhem *et al.* [44] against an RSA implementation. The authors show how to extract the private key of an RSA implementation by only decrypting chosen input values and measuring the corresponding execution time. For decrypting a chosen input value c, the modular exponentiation $exp = c^d \ mod(N)$ has to be computed, with d the private key and N the RSA modulus (is publically known). The modular exponentiation is realized with the square-and-multiply algorithm that is given in Algorithm 1. The algorithm computes the result sequentially for each bit d_i of the exponent (private key). If d_i is one both a squaring and a multiply operation are executed. If d_i is zero only a squaring operation is executed. Hence, the computation should take longer if $d_i = 1$. Moreover, the modular reduction $mod(N)$ is only applied if the intermediate result of exp is larger than N. These two properties make the execution time of the algorithm data dependent and thus vulnerable to timing attacks.

Algorithm 1 Square-and-multiply $exp = c^d \ mod(N)$

1: $n \leftarrow$ bit width of d (without leading zeros, *i.e.* $d_{n-1} = 1$)
2: $exp \leftarrow c$
3: **for** $i = n - 2$ to 0 **do**
4: $\quad exp \leftarrow exp^2 \ mod(N)$
5: $\quad$ **if** $d_i = 1$ **then**
6: $\quad\quad exp \leftarrow exp \cdot c \ mod(N)$
7: $\quad$ **end if**
8: **end for**
9: **return** exp

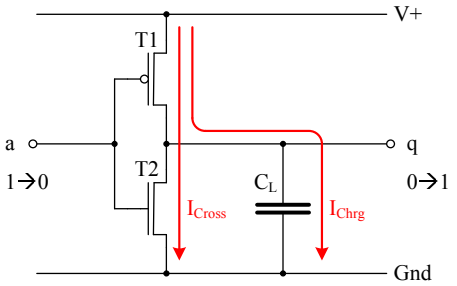


Figure 3.1: Inverter circuit that demonstrates the dynamic power consumption in CMOS devices.

For revealing d, an attacker starts with d_{n-2} (note that $d_{n-1} = 1$ by definition). Two sets of input values $C1_j$ and $C2_j$ with $j = 0 \ldots k$ have to be selected according to a selection criteria such that for one set a modular reduction is necessary and for the other not. For d_{n-2} the selection criteria can be for example ${C1_j}^3 < N$ and ${C2_j}^2 < N < {C2_j}^3$ for $j = 0 \ldots k$ as shown in [188]. If $d_{n-2} = 1$, the algorithm will compute the multiplication step in Line 6: $exp = exp \cdot exp^2 \ mod(N)$. Modular reductions will be necessary for input values $C2_j$, but not for input values $C1_j$. If $d_{n-2} = 0$, the algorithm will skip the multiplication step in Line 6. No modular reductions will be necessary for both sets of input values $C1_j$ and $C2_j$. Hence, when computing the average execution times $\bar{t}_{C1}$ and $\bar{t}_{C2}$ for the input-data sets $C1_j$ and $C2_j$, they will only significantly differ from each other if $d_{n-2} = 1$. The difference of the average execution times leaks the value of d_{n-2}. The remaining bits of the private key can be determined analogously bit-by-bit.

3.1.2 SCA Attacks Using Power Consumption

When Kocher *et al.* [104] published their ground-breaking work about power analysis attacks in 1999, relevance of SCA attacks dramatically increased. Power analysis attacks measure the power consumption of cryptographic devices to reveal secret information. Basic idea behind the attack is that the power consumption of devices that base on CMOS technology is dependent on the processed data. Most modern cryptographic devices are implemented using CMOS circuits.

The power consumption of a CMOS circuit is composed of a static part and a dynamic part. Due to the nature of CMOS circuits, where ideally no power is dissipated when signals do not change, the dynamic part typically dominates the overall power consumption. Note that newer CMOS technologies have significantly higher static power consumption because of increased leakage effects. In order to describe the characteristic of dynamic power consumption, the CMOS inverter circuit (most simple CMOS circuit) shown in Figure 3.1 can be used. The two transistors $T1$ and $T2$ isolate the output q from the

input a. When the input a is fixed to logic 1 or logic 0, either $T1$ or $T2$ is nonconducting. No current is flowing. When a changes from $0 \rightarrow 1$ or from $1 \rightarrow 0$ (*i.e.* it makes a transition), $T1$ and $T2$ are concurrently conducting for a very short time, resulting in a cross current I_{Cross} from $V+$ to Gnd. Moreover, the load capacitance C_L needs to be either charged or discharged when the input a changes, causing a charge/discharge current I_{Chrg}. These two currents build together the dynamic power consumption of CMOS devices. Since the dynamic power consumption depends on the transitions within a circuit, it is also dependent on the processed data. This data-dependent behavior makes the power consumption of a CMOS device a very effective side channel.

A widely used method to determine the power consumption of a device, is measuring the voltage drop across a resistor that is connect in series to the supply line. The value of the resistor depends on the power consumption of the examined device itself (*i.e.* a larger resistor value is required when power consumption is low) and on the sensitivity of the equipment that is used for measuring the voltage drop. Alternatively, a special current probe can be used to directly measure the power consumption. For recording the power consumption of a device, a digital-storage oscilloscope is typically used. The oscilloscope records the power consumption over a certain period of time, which leads to a so-called power trace that is stored for further processing and analysis.

When using the power consumption of a cryptographic device as side channel, two basic attack techniques can be distinguished: simple power analysis (SPA) and differential power analysis (DPA). In the following the two techniques are described in short.

Simple Power Analysis

Simple power analysis tries to reveal the secret key of a device by visually inspecting the recorded power trace(s). According to [117], single-shot SPA attacks using only one power trace and multi-shot SPA attacks using several power traces can be distinguished. In case of multi-shot SPA attacks, power traces can be averaged to remove noise before inspecting them. The implementation of a cryptographic algorithm is vulnerable to SPA attacks, if different key-dependent operations result in different patterns in the power trace that can be distinguished. An implementation of the square-and-multiply algorithm (see Algorithm 1) presented in Section 3.1.1 would be susceptible to SPA attacks, if executing the conditional multiplication operation results in a clearly discernable pattern in the power trace. In such a case, an adversary can easily extract the value of the bits of the private key d by looking at a power trace that has been recorded while executing the algorithm.

Kocher *et al.* [104] have been the first that mentioned that SPA attacks can be used to reveal secrets of cryptographic devices. Practical examples that deploy SPA attacks have been presented, for example, by Messerges *et al.* [120] on an implementation of the Data Encryption Standard (DES), by Mangard *et al.* [114] on the key schedule of AES, by Compton *et al.* [37] on the key schedule of Serpent, or by Coron *et al.* [38] on ecliptic curve cryptography (ECC). SPA

attacks have the advantage that they are very simple and fast, since they only require a small number of power traces. However, integration of more and more functionality on a single chip (often called system-on-chip) significantly increases the noise in the power traces and makes it rather difficult to mount SPA attacks in practise on such devices.

Differential Power Analysis

A more advanced SCA technique is differential power analysis (DPA), which uses a large number of power traces to extract secret information from a cryptographic device by means of statistical methods. DPA attacks have the advantage that even noisy power traces can be used and that no detailed knowledge about the attacked device is necessary. A DPA attack follows always the same principle and mainly consists of five steps [117]:

1. **Selecting an appropriate intermediate value.** An appropriate intermediate value of the executed cryptographic algorithm has to be selected for the DPA attack. The intermediate value should be a function of known input data (*e.g.* plaintext or ciphertext) and the secret key (or a part of it). Typically, a nonlinear function like a substitution box (S-box) operation is used.

2. **Recording power traces.** Power traces t are recorded with a digital-storage oscilloscope while the examined device, often called device under test (DUT), computes the key-dependent function with the known input data. The number of required power traces for a successful attack mainly depends on the leakage behavior of the attacked device. The higher the data-dependent leakage the less power traces have to be recorded. Another factor that influences the number of required power traces is the signal-to-noise ratio (SNR) of the measurement setup, which should ideally be as high as possible. Also important for successful DPA attacks is appropriate triggering of power-trace recording. Hence, efficient attacks are only possible if recorded traces are well aligned along the time axis. When the traces are not well aligned, proper preprocessing techniques have to be applied first before conducting the attack.

3. **Calculating hypothetical intermediate values.** After recording the power traces, hypothetical intermediate values are computed for the input data used during the measurement and for all possible key hypotheses. When attacking for example one byte of the secret key, intermediate values for $2^8 = 256$ different key hypotheses have to be computed.

4. **Calculating hypothetical power-consumption values.** The previously computed hypothetical intermediate values are then transferred to hypothetical power-consumption values h by using an appropriate power model. Typical power models are the bit model, the Hamming-weight model, the Hamming-distance model, and the zero-value model. Especially

Hamming weight and Hamming-distance model are widely deployed. The Hamming-weight model assumes that the power consumption of a device relates to the number of bits that are set to one (the value $0x03$ has *e.g.* a Hamming weight of two). The Hamming-distance model assumes that the power consumption of a device relates to the number of $1 \rightarrow 0$ and $0 \rightarrow 1$ transitions (the Hamming distance of the two values $0x03$ and $0x\text{C}0$ is 4). The better the deployed model fits the power consumption of the examined device, the less power traces are necessary for a successful attack.

5. **Comparing measured traces with hypothetical power-consumption values.** For comparing the measured power traces t with the hypothetical power-consumption values h, statistical methods are used. A widely used method is the Pearson correlation r that calculates the linear dependency between measured traces and the hypothetical power-consumption values. The formula for computing the Pearson correlation is given in (3.1), where D relates to the number of measured traces, and $r_{i,j}$ corresponds to the correlation value for the *i–th* key hypothesis and the *j–th* point within the power traces. The higher the resulting correlation value the higher is the linear dependency between measured traces and hypothetical power-consumption values. In case of a successful DPA attack, only the hypothesis that relates to the correct key contains high correlation values (correlation peaks). All other hypotheses have significantly lower correlation values.

$$r_{i,j} = \frac{\sum_{d=1}^{D}(h_{d,i} - \bar{h}_i) \cdot (t_{d,j} - \bar{t}_j)}{\sqrt{\sum_{d=1}^{D}(h_{d,i} - \bar{h}_i)^2 \cdot \sum_{d=1}^{D}(t_{d,j} - \bar{t}_j)^2}} \tag{3.1}$$

Figure 3.2 gives a graphical overview of the steps required for a DPA attack. After performing all steps, a part of the secret key (*e.g.* a byte) has been determined. In order to reveal the hole secret key, several DPA attacks have to be applied successively for all parts. Often, power traces are recorded for the hole computation of the cryptographic algorithm. Hence, only steps 3. to 5. have to be repeated successively.

Conducting DPA attacks requires typically much more time than conducting SPA attacks due to the large number of power traces and the additional statistical analysis step. However, even very weak data-dependencies in power traces can be detected and used for such attacks (*e.g.* when covered by noise). There are numerous publications that describe DPA attacks. First DPA-attack results have been presented by Kocher *et al.* [104] on an implementation of DES. Later attacks on AES [32, 94, 142], IDEA and RC6 [109], ARIA [68, 144], SEED [200], KeeLoq [48], RSA [121], and ECC [38, 143] followed. Especially the attack on KeeLoq in 2008 presented by Eisenbarth *et al.* [48] has illustrated that DPA attacks are not only of academic interest, but can also be used to break real-world devices, underlining their practical relevance.

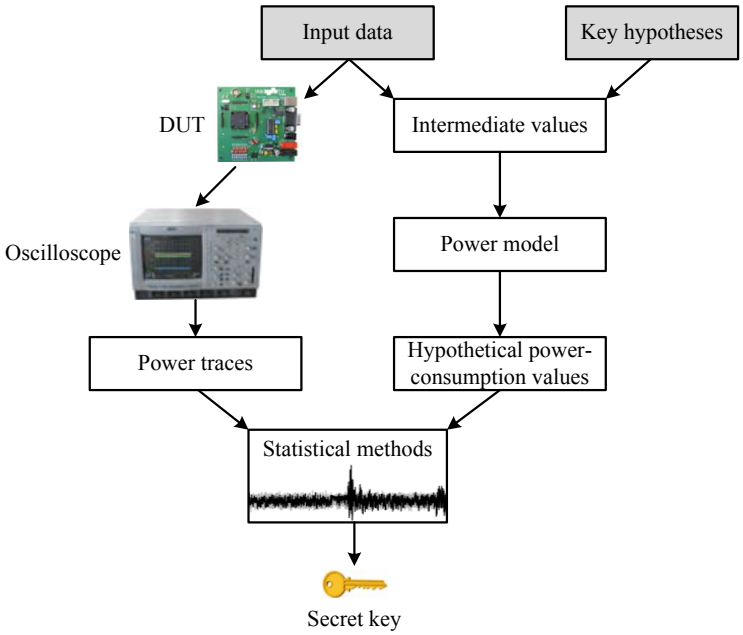


Figure 3.2: Overview of the steps required for a DPA attack.

3.1.3 SCA Attacks Using Electromagnetic Emanations

As shown in Section 3.1.2, the power consumption of CMOS devices contains data-dependent information that can be used to mount DPA attacks. However, not only the power consumption of CMOS devices is a suitable side channel, but also the electromagnetic (EM) emanation.

Whenever a current I is flowing through a conductor, a magnetic field H is created. In case of a single wire, the magnetic field H depends on the current I and the distance r to the wire as illustrated in Figure 3.3. The magnetic field decreases with the distance to the wire. In order to consider the property of the material that is penetrated by the field, a material constant μ (called permeability) is multiplied with the field H, resulting in the magnetic induction B. Further, integrating the magnetic induction B over an area A, leads to the magnetic flux Ψ that is present within this area. Consequently, a changing current $i(t)$ leads to a changing magnetic flux $\Psi(t)$. According to Faraday's law that is given in (3.2), a changing magnetic flux $\Psi(t)$ induces an electric field E_i. This relation inseparably links magnetic field and electric field with each other, forming the electromagnetic field.

$$u_i = \oint E_i \cdot ds = -\frac{d\Psi(t)}{dt} \tag{3.2}$$

Data-dependent information that is present in the power consumption (*i.e.* supply current) of a CMOS device is therefore also discernable in the corre-

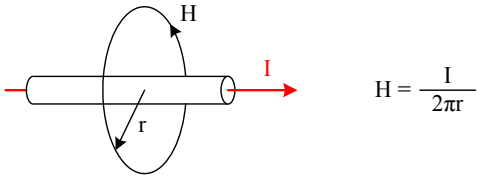


Figure 3.3: Relation between current I and magnetic field H in a single wire.

sponding electromagnetic field that is radiated. For gathering the electromagnetic field, antennas are used. When measuring the EM field in close proximity of the examined device (near field), magnetic loop antennas (near-field probes) are typically deployed. When measuring the EM field from a greater distance of the device (far field), dipole antennas are used.

The same kinds of attacks that are used for power-consumption measurements (see Section 3.1.2) can also be applied when measuring the EM field of a cryptographic device. Instead of power traces, EM traces are used. As initially suggested by Quisquater *et al.* [158], SPA attacks are called simple electromagnetic analysis (SEMA) attacks, and DPA attacks are called differential electromagnetic analysis (DEMA) attacks. Often, EM-field measurements in the near field lead to even better attack results (*e.g.* with less traces) than power measurements, since advantageous placement of the measurement probe or antenna allows to reduce the influence of noise sources that do not contain data-dependent information. Reducing such noise sources by using local information is of special interest when analyzing for example system-on-chip (SoC) devices, where several components are located on a single chip. Instead of measuring the power consumption of the overall chip, a small probe is precisely placed above the chip, gathering only the EM emissions of the component that is of interest for the attack.

First attack results based on EM traces have been presented by Gandolfi *et al.* [61] on DES and RSA. EM attacks on Rijndael and ECC followed by Gebotys *et al.* [63] and Mulder *et al.* [128]. Mangard has shown that DEMA attacks are not limited to the near field of a device but can also be successful in the far field [115]. Moreover, Agrawal *et al.* [8] have suggested to combine information from both EM emissions and power consumption of a device to make SCA attacks more efficient. Combining information from several side-channel sources leads to so-called multi-channel attacks.

3.2 Fault Analysis

Another important kind of implementation attack is fault analysis. In contrast to side-channel analysis, fault analysis is an active technique, where either the device itself if manipulated, its inputs signals, or the operating environment. Fault analysis aims to reveal secret information stored on a cryptographic device by provoking an abnormal behavior that results, for example, in a faulty

computation.

A well-known example that illustrates how faulty computations can be utilized to get the secret key of a cryptographic device is the implementation of the RSA algorithm [162] with Chinese Remainder Theorem (CRT). The RSA algorithm uses a modulus N that is the product of two large primes p and q, as well as a public key e and a secret key d. The public key e is randomly selected in a way, such that its multiplicative inverse modulo $\varphi(N)$ exists (φ is Euler's totient function). This property of e allows to derive the private key with $d = e^{-1}\ mod(\varphi(N))$. When signing a message m (in practise m is first hashed before signing) with the RSA algorithm, the signature s is computed with $s = m^d\ mod(N)$. In order to verify the correctness of the signature $s^e\ mod(N)$ is computed, which has to result in m if the signature is valid. Since the modular exponentiation es a very computation-intensive operation, a mathematical trick based on the CRT is used to speed-up computation. Instead of computing s directly $mod(N)$, the CRT is used to compute the signature as a linear combination of $s_p = x^d\ mod(p)$ and $s_q = x^d\ mod(q)$ with $s = as_p + bs_q\ mod(N)$, whereas $a = q(q^{-1}\ mod(p))$ and $b = p(p^{-1}\ mod(q))$. For a fault attack, an adversary computes two different signatures s and $\hat{s}$ using the same input message. The signature s is determined as described above without inducing a fault, while $\hat{s}$ is determined with inducing a fault during the computation of the s_p leading to $\hat{s} = a\hat{s}_p + bs_q\ mod(N)$. As shown in (3.3), only the terms with bs_q cancel out when subtracting s and $\hat{s}$ from each other:

$$\begin{aligned} \Delta s = s - \hat{s} &= (as_p + bs_q) - (a\hat{s}_p + bs_q) \\ &= a(s_p - \hat{s}_p) \\ &= q(q^{-1}\ mod(p)) \cdot (s_p - \hat{s}_p). \end{aligned} \tag{3.3}$$

If the difference Δs is not divisible by p, the Greatest Common Divisor (GCD) of Δs and N will lead to q (because a is a multiple of q according to CRT), which is a factor of N. Since the security of the RSA algorithm is based on the assumption that factoring N is hard, knowing q allows an adversary to compute $\varphi(N) = (N/q - 1)(q - 1)$ and further to derive the secret key d by inverting e modulo $\varphi(N)$. Fault attacks on RSA with CRT have been first presented by Boneh *et al.* [26] and Joye *et al.* [95].

Fault attacks can basically be divided into three groups of attacks: non-invasive attacks, semi-invasive attacks, and invasive attacks. Non-invasive attacks make no modification of the analyzed device itself, but only change input signals or the operating environment to provoke a faulty behavior. Semi-invasive attacks require to modify the analyzed device, *e.g.* decapsulating an IC to get direct access to the die for inducing a fault. Invasive attacks do not only modify the device, but also contact it electrically. By using special equipment like a Focused Ion Beam (FIB), the layout of a chip can be easily modified (*e.g.* cut or reconnect wires) and wires on the chip can be contacted through probes. Invasive attacks are the most powerful ones, but the equipment that is required for such attacks is quite expensive and often only available in special laboratories.

Other properties of fault attacks comprise: fault duration, fault control, timing behavior, and fault model. Fault duration indicates how long a fault remains present after inducing it. Transient faults disappear after some time, whereas permanent faults are irreversibly and remain present after inducing them. Permanent faults can result from overstressing a device. Extreme overstress of a device (*e.g.* operating it with a higher supply voltage) can even completely destroy it. Fault control describes which parts of the analyzed device are affected during fault induction. Global faults affect the whole device. Local faults can be limited to a part of the analyzed device, making an attack more precise. For some attacks, timing behavior is very important, *i.e.* accurate triggering of fault induction is required. The fault model indicates what kind of erroneous behavior of the attacked device is expected. This comprises for example, the number of bits that are flipped in a memory during an attack (in both directions, from 1 to 0, or from 0 to 1), or how the program flow is changed (skipping of instructions). In the following, we give a short overview of practical fault-analysis techniques.

3.2.1 Temperature Variations

When cryptographic devices are exposed to extreme temperature variations, faulty behavior can occur. Devices that base for example on CMOS technology (which are most cryptographic devices) change their properties when they are heated up or cooled down. A CMOS device that operates at increased temperature has different wire resistance, leakage current, and diode voltage drop as when it operates at room temperature. Potential consequences are that the content of RAM cells can randomly change when heated up, or that writing to non-volatile memories (NVMs) is no longer possible (reading could still be possible due to different read and write temperature thresholds) [19, 159]. Temperature variations work in a global manner and require no precise timing, since temperature changes only gradually.

3.2.2 Supply Voltage and Clock Variations

A widely used technique to provoke a faulty behavior of a device is to use sudden changes in the supply voltage (spikes) or in the external clock signal (glitches). Sudden changes in the supply voltage can cause a microcontroller to misinterpret or to skip instructions. Variations of the external clock signal can lead to data misread and instruction miss. Data misread can occur when for example reading data from a memory bus before all data have been latched. An instruction miss describes the effect where the execution of the next instruction is started before the execution of the actual instruction is completely finished [11, 12, 19, 105]. Both supply voltage and clock variations require precise timing (*i.e.* have to be triggered at appropriate points in time during computation of the device) and can only be applied globally.

3.2.3 Electromagnetic Interferences

A fast-changing current in a coil produces an electromagnetic field. When placing the coil close to a conducting surface (*e.g.* the chip of an IC), eddy currents are induced into the conducting surface by the electromagnetic field. Pulse strength and efficiency of the so created electromagnetic injection depends on several factors like the frequency of the electromagnetic field (*i.e.* the frequency of the current through the coil), the characteristic of the coil (size, number of windings), and distance of the coil to the conducting surface [159, 163]. Depending on the strength of the electromagnetic injection, different effects can be achieved. Weak injections may not affect the operation of a device at all, whereas strong injections can even alter the content of non-volatile memories (EPROM, EEPROM, or FLASH) and consequently lead to erroneous computations. Fault attacks that use electromagnetic interferences do not necessarily require to decapsulate a chip from its package. However, when fault attacks should not only be conducted globally but also in a local manner by focusing on a part of the chip, accurate position of the coil is required (which is easier to accomplish when decapsulating a chip). Electromagnetic interferences allow also precise timing of the induction of a fault.

3.2.4 Optical Inductions

When light illuminates the surface of a chip, free electrons are generated that can drive a current. This effect is often also named Optical Beam Induced Current (OBIC) [178]. The current that is induced by the light causes transistors to conduct, leading to transient faults that can be used for mounting fault attacks [172]. For inducing faults, the light (typically laser light is used) has to be focused on the chip surface during computation of data. Hence, decapsulation of the chip is required to get visibility of the regions that are attacked. Faults caused by optical inductions can be triggered very precisely in time and can be applied globally as well as locally, making them rather powerful. Another advantage of such attacks is that no expensive equipment is necessary, a microscope and a modified laser pointer are enough.

Besides optical attacks from the front side of the chip, there exist also attacks from the back side (rear-side attacks) [171]. These attacks use light with longer wavelength (infrared light) that reaches the transistors through the substrate of the chip (the substrate is transparent for infrared light). Hence, such attacks can even be applied when the transistors are covered by several metal layers that prevent light from the front side of the chip to reach them. Optical attacks from the back side are of special interest for newer CMOS process technologies since they use an increasing number of metal layers.

3.3 Summary

In this chapter we have given basic information about implementation attacks which are a very powerful technique. Even if algorithms are mathematically

secure, implementation attacks can be used to exploit weaknesses of the physical implementation to reveal secret information. We have covered side-channel analysis as well as fault analysis. Side-channel analysis is a passive attack technique that only measures physical properties of a device. Fault analysis on the other hand is an active attack that either manipulates the device itself, the input signals, or the operating environment. In order to make implementation attacks even more powerful, both side-channel analysis and fault analysis can be applied jointly, leading to so-called combined attacks [10].

4

Countermeasures Against Implementation Attacks

Whenever new implementation attacks have been presented, researches have immediately tried to find proper techniques to prevent these attacks or at least to make them less efficient. Such techniques are called countermeasures. In some cases, publication of newly detected implementation attacks has been delayed to prevent misuse of them and to gain time for developing proper countermeasures.

In this chapter we discuss basic countermeasures that aim to impede the application of implementation attacks like side-channel analysis and fault analysis. Typically, the perfect countermeasure that protects against all kinds of attacks does not exist. In order to make cryptographic devices resistant against a wide spectrum of attacks, several countermeasure approaches are combined. The targeted protection level of a cryptographic device is typically defined by the skills and the resources that a potential adversary has. Hence, more advanced countermeasures have to be integrated when protecting a device against an adversary with deep insider knowledge and powerful equipment, in comparison to protecting the device for example against a clever hobbyist with only limited budget [4]. Unfortunately, integrating countermeasures usually goes along with an increase of power consumption and/or execution time, and design complexity of a device. For highly constrained devices like low-cost RFID tags, finding suitable countermeasures that provide a good trade off between achieved security and additional costs is very important.

In the remainder of this chapter we first give details about countermeasures against side-channel analysis attacks, followed by countermeasures against fault-analysis attacks. The chapter closes with a short summary. Parts of this chapter contain information that have been published in papers at the CT-RSA confer-

ence 2009 [148] and the WISA workshop 2009 [152], respectively.

4.1 Side-Channel Analysis Countermeasures

As illustrated in Chapter 3, side-channel analysis is a very powerful technique to reveal secret information from cryptographic devices. Consequently, high effort has been spent by the research community in the last decade to come up with proper countermeasures to strengthen the resistance of implementations against side-channel analysis. First, we describe countermeasures against timing-based attacks, followed by countermeasures that try to prevent power analysis attacks as well as electromagnetic analysis attacks of cryptographic devices.

4.1.1 Countermeasures Against SCA Attacks Using Timing Information

In contrast to SCA attacks that measure the power consumption or the EM emissions of a device, timing-based attacks are much easier to prevent, especially in case of synchronously clocked circuits. When thinking of the square-and-multiply algorithm given in Algorithm 1 in Chapter 3, two approaches could be used to prevent timing attacks. First, the execution time of the algorithm can be made independent of the input value. This requires to always preform a modular reduction of the intermediate result regardless whether it is required or not. Additionally, the if branch of the algorithm can be extended with an appropriate else branch to obtain constant execution time for all input values. In the else branch, a multiplication with one could be added. A modified version of the square and multiply algorithm with constant execution time is presented in Algorithm 2. However, constant execution time has the disadvantage that the algorithm has the worst-case run time for all input values. The second approach for preventing timing attacks is blinding, which has been suggested in the paper of Kocher [103]. Blinding introduces randomness into the RSA computation by using a random value r. For each input value a new r is selected. When encrypting a message m with the public key e by computing $c = m^e \ mod(N)$, an additional multiplication with r^e is performed, leading to $c_r = r^e \cdot m^e \ mod(N)$. In order to decrypt c_r, the following calculation has to be carried out: $c_r^d \cdot r^{-1} \ mod(N) = m$. The execution time of the modular exponentiation is now uncorrelated from the value of the private key.

4.1.2 Countermeasures Against SCA Attacks Using Power Consumption

After Kocher *et al.* [104] have discovered that many of the cryptographic devices on the market are susceptible to simple power analysis (SPA) attacks or to the even more powerful differential power analysis (DPA) attacks, they have immediately started to work on developing appropriate countermeasures.

Algorithm 2 Modified square-and-multiply $exp = c^d \ mod(N)$

```
1: n ← bit width of d (without leading zeros, i.e. d_{n-1} = 1)
2: exp ← c
3: for i = n − 2 to 0 do
4:    exp ← exp^2 mod(N)
5:    if d_i = 1 then
6:       exp ← exp · c mod(N)
7:    else
8:       exp ← exp · 1 mod(N)
9:    end if
10: end for
11: return exp
```

Similar to timing-based attacks, SPA attacks can be prevented when avoiding, for example, conditional branches that relate to the secret key. However, constant execution time alone is not enough. Also differences in the power-consumption pattern caused by the individual operations or the value of the operands can be deployed to successfully mount SPA attacks (*e.g.* by averaging over several power traces). DPA attacks are much harder to impede than SPA attacks. Even a very weak data-dependent leakage that is covered by noise can be utilized to reveal secret information from a device. Hence, techniques that aim to strengthen the resistance against DPA attacks can also be used to prevent SPA attacks (*e.g.* increase noise or randomize execution).

According to Mangard *et al.* [117], countermeasures against DPA attacks can principally be divided into two groups: hiding and masking. The goal of hiding is to decouple the power consumption of a cryptographic device from its internally processed data values. Masking tries to break the link between the intermediate values and the values actually computed by the device.

Countermeasures Based on Hiding

Basically there are two approaches to achieve hiding of data-dependent information in the power consumption of a cryptographic device: hiding in the amplitude dimension and hiding in the time dimension.

Hiding in the Amplitude Dimension Hiding in the amplitude dimension blurs the data-dependent information by varying the power-consumption characteristic of a device in its amplitude. Variations in the amplitude dimension either lower the signal-to-noise ratio (SNR) or reduce the data-dependent part of the power consumption and thus make the detection of the data-dependent leakage more difficult (*i.e.* more power traces are required for an attack). In order to lower the SNR, noise generators are integrated into cryptographic devices or multiple operations are executed in parallel to increase the overall noise [187]. In case of RFID tags, additional noise is typically inherently introduced by the

strong RF field of the reader that covers the data-dependent information of the tag. Another way to achieve hiding in the amplitude dimension is to reduce the data-dependent part of the power consumption by equalizing the power consumption of a device. This is typically done by using special *dual-rail precharge* (DRP) logic styles. There, logic cells always consume the maximum amount of power, independent of the processed data. Examples of such logic styles are Sense Amplifier Based Logic (SABL) and Wave Dynamic Differential Logic (WDDL).

Hiding in the Time Dimension The second approach uses hiding in the time dimension, which is achieved by randomizing the execution of a cryptographic algorithm or by randomly changing the clock signal [117]. There are mainly two possibilities how the execution of an algorithm can be randomized. The first possibility is to insert dummy operations such as additional rounds (or only parts of it). These dummy operations can be processed before or after the execution of the actual algorithm, which results in a significantly increased runtime. The second possibility is to shuffle the sequence of operations [36], where the runtime of the algorithm remains unchained. For algorithms like the Advanced Encryption Standard (AES), several operations can be rather easily randomized. However, for other algorithms randomizing operations might be more difficult to achieve or even impossible. In practice, dummy operations and shuffling of operations are often jointly applied. The second technique to realize hiding in the time dimension is to randomly change the clock signal. This can either be done by using multiple clock domains or by varying the clock frequency. Devices with multiple clock domains randomly select clock signals from different clock domains. Devices that can vary their clock frequency have typically an internal oscillator that deduces its frequency from random numbers [196]. In both cases, execution time of the algorithm is no longer fixed but varies in time. Important for all countermeasure that are based on hiding in the time dimension is that an attacker should not be able to discern the insertion of dummy cycles or the changing of the clock signal.

Countermeasures Based on Masking

When using masking as countermeasure against DPA attacks, an intermediate value v is concealed by a random mask m using a masking operation $\times$. The resulting value $v_m = v \times m$ is called masked intermediate value [117]. In that way, the corresponding power consumption relates to v_m and no longer to v itself. Ideally, the mask m should be changed for every execution of the algorithm (*e.g.* for each encryption process). As masking operation, either boolean masking or arithmetic masking is used. An example for boolean masking is the exclusive-or function. Typical arithmetic masking schemes are modular addition and modular multiplication.

Masking can be seen as some kind of secret sharing, where the secret v is represented by the two shares v_m and m. Knowing only one of the two shares is not enough to deduce v, both have to be known. In order to increase resistance

against implementation attacks, multiple shares (*i.e.* several masks) can be used for concealing the intermediate value [32]. When applying masking to asymmetric encryption schemes, it is typically named blinding. Various approaches exist such as message blinding or exponent blinding.

Masking can be applied at different levels. At architectural level, intermediate values of an algorithm are masked before they are further processed, as described above. For look-up tables T that are used to compute non-linear operations of an algorithm, masked versions T_m of the tables have to be created. This increases computational effort and storage requirement of a masked algorithm implementation. An example for a masked implementation of AES in software is given for example in [72]. Masking can also be applied at the cell level by using masked logic styles. There, the logic cells of a circuit work only on the masked values and on the masks to decouple their actual power consumption from the original intermediate (unmasked) values [117]. A number of different masked logic styles have been introduced over the years, for example: Random Switching Logic (RSL) [176], Masked Dual-Rail Precharge Logic (MDPL) [155], Dual-Rail Random Switching Logic [33], and improved MDPL (iMDPL) [154].

4.1.3 Countermeasures Against SCA Attacks Using Electromagnetic Emanations

Power consumption and electromagnetic emanations of a device strongly relate to each other as illustrated in Section 3.1.3. Consequently, all the countermeasure approaches that have been presented for preventing SPA and DPA attacks can also be applied to make simple electromagnetic analysis (SEMA) and differential electromagnetic analysis (DEMA) attacks less effective. Since EM measurements can be utilized to focus only on the emissions of a specific part of a device (*e.g.* where the cryptographic module is located), critical parts can be covered by an additional shielding layer to lower the emissions.

4.2 Fault-Analysis Countermeasures

The most intuitive approach to detect whether a fault has occurred during an operation or not is computing it several times and comparing the results. If an error has occurred, further operation is stopped and all involved (temporary) data are discarded. Typically, the operation is computed twice and the results are checked on equality [160]. Computing an operation twice can be achieved by reusing the same module for both computations (time redundancy) or by using two distinct modules concurrently (space redundancy). In both cases either execution time or hardware effort is doubled. However, permanent faults might be difficult to detect or even not discovered when reusing the same module for computing an operation twice.

Another technique to detect the injection of a fault is to use the inverse of an operation to check the previously computed result [98]. When thinking of asymmetric cryptography, an RSA signature s that has been computed with $s =$

$m^d \ mod(N)$ from the message m, can easily be verified by computing $s^e \ mod(N)$ and checking if the result equals to m (compare Section 3.2). For symmetric cryptography, a similar check can be done. Encrypting a message m under the key k leads to the ciphertext $c = E_k(m)$. When decrypting the ciphertext under the same key by computing $D_k(c)$, the result has to be equal to m. Such a verification by utilizing the inverse of an operation comes also not for free. Implementing the inverse operation often causes a significant increase in code size or hardware effort (*i.e.* when the inverse operation is not require for normal functioning). Further, computing the inverse operation also increases execution time. Different granularity level can be selected for checking results. Hence, besides checking results after computing the whole algorithm, also checks after individual rounds of an algorithm are possible [191].

An important aspect that has to be considered in this context is that the comparison process itself can be the aim of a fault attack [100]. Thus, even if two separate computations of an algorithm lead to different results, or if computing the inverse of an operation results in a value different from the original one, skipping the comparison process by inducing a properly timed fault prevents detection of the fault. However, this requires to induce at least two faults, one during the computation of the algorithm and one during the comparison process. For some fault attacks, its even enough to detect whether the computation failed or not [199].

Other approaches use error detection as well as error-correction codes to make fault attacks less effective [97, 107, 108, 110, 157, 177]. Further, also hardware countermeasures against specific fault-injection techniques are integrated into cryptographic devices. Temperature sensors stop operation of the device as soon as the temperature is outside the specified/allowed range to prevent fault injections via temperature variations. Influence of supply voltage and clock variations is reduced by using glitch detectors and DC filters (in combination with low and high-voltage sensors). In order to completely prevent clock variations, internal oscillators are deployed for deducing the clock signal, making an external clock signal needless. For counteracting optical inductions an extra metal layer is used that covers the chip surface. However, this measure only protects against optical inductions from the front side of the chip. So-called rear-side attacks are still applicable [171]. Another protection measure against optical inductions is the integration of light sensors that stop the operation of the chip when an attack is detected [84].

4.3 Summary

In order to make implementation attacks less effective, a variety of countermeasures have been developed over the years by academia and industry. Countermeasures range from hiding and masking techniques for impeding side-channel analysis attacks based on power and EM measurements, to detection mechanisms for fault attacks like duplication of computations and error-detection codes. Typically, multiple countermeasure approaches are combined to achieve a proper

protection level of a device. Integrating countermeasures comes at expense of increased execution time, additional hardware costs, or higher design complexity. These aspects have to be considered when integrating countermeasures to resource-constrained devices such as RFID tags. In the following chapters, we focus on the evaluation of two side-channel analysis countermeasures that aim for application on low-cost RFID tags.

5

Side-Channel Analysis of Low-Cost UHF RFID Tags

After a basic introduction to implementation attacks in Chapter 3 and countermeasures against them in Chapter 4, we concentrate now on implementation attacks on real-word devices. In particular, we analyze the susceptibility of commercially available low-cost RFID tags to side-channel analysis as well as fault analysis. Further, we evaluate the effectiveness of countermeasures that seem to be suitable for integration on low-cost tags. In this chapter we present side-channel analysis results on UHF tags. The subsequent chapter presents fault-analysis results on HF and UHF tags, followed by two chapters that evaluate countermeasures based on hiding (shuffling and the detached power-supply approach) in context of RFID tags. Most of the work presented in these chapters has been carried out from 2007 to 2009 within the project "**B**uilding **R**adio-Frequency **Id**entification Solutions for the **G**lobal **E**nvironment" (BRIDGE), which has been funded by the European Commission under the Sixth Framework Programme.

When we started our research on evaluating the vulnerability of RFID tags to side-channel analysis in 2007, only little information has been available about this topic in published literature. Although it has been widely known at that time that power-analysis attacks [104] and EM-analysis attacks [7, 61, 115] are a serious threat for contact-based devices like smart cards, concrete results for contactless devices like commercially available RFID tags have been rare. Carluccio *et al.* [31] have presented first electromagnetic (EM) side-channel analysis results on contactless smart cards in the HF range at 13.56 MHz. Hutter *et al.* [78] have conducted EM and power-analysis measurements on RFID-enabled prototype devices operating also in the HF range. Two prototype devices have

been analyzed, one with a cryptographic primitive implemented in software, the other with a cryptographic primitive implemented in hardware. Results on UHF RFID tags and EM analysis have been given by Oren and Shamir [141]. The authors have described a new attack called *parasitic-backscatter attack*. This attack is possible since the amount of power that is reflected by a UHF RFID tag is related to the power consumption of its internal circuit. Further, the authors explain how the *parasitic-backscatter attack* can be used to extract the secret kill password from EPC Generation 1 tags.

Relying on the results of Oren and Shamir [141], we have gone a step further and focused on determining the susceptibility of EPC Generation 2 tags to differential electromagnetic analysis (DEMA). We have been the first that presented DEMA results on commercially available RFID tags. Our results of the DEMA attacks have been published at the CT-RSA conference 2008 [147]. Results on contact-based attacks on commercially available UHF tags have been published in [79], which is a joint work with Michael Hutter and Martin Feldhofer.

This chapter is organized as follows. Section 5.1 provides general information about UHF RFID tags. In Section 5.2, details about the examined UHF RFID tags are given, followed by a description of the measurement setup in Section 5.3. Side-channel analysis results that have been conducted are presented in Section 5.4. A summary is given in Section 5.5.

5.1 General Information About UHF RFID Tags

RFID systems can be classified by the frequency of the RF field and by the coupling method. The frequencies used by RFID systems range from about 125 kHz in the low-frequency range up to 5.8 GHz in the microwave range [59]. Deployed coupling methods are: electric coupling, magnetic coupling, and electromagnetic coupling. In this chapter we focus on electromagnetic-coupled systems in the UHF range operating at a frequency of 868 MHz. In contrast to electric coupling and magnetic coupling which operate in the near field, electromagnetic coupling operates in the far field by using electromagnetic waves.

Responsible for the existence of electromagnetic waves is the limited propagation speed of the electromagnetic field. At a certain distance from the antenna the electromagnetic field can no longer follow the voltage changes at the antenna. The electromagnetic field separates from the antenna and propagates as an electromagnetic wave. The region where the electromagnetic field is separated from the antenna is named far field [59]. For UHF RFID tags operating at a frequency of 868 MHz, the far field starts at a distance of about 5.5 cm from the antenna. The simplest antenna shape that is used for generating electromagnetic waves is the dipole antenna which consists of two wires. Since the attenuation of the RF field in the far field is less than in the near field, electromagnetic-coupled systems achieve longer read ranges. Typically, read ranges of 2 to 3 m and more can be achieved, depending on the power of the RFID reader. Because the electromagnetic field is completely separated from the antenna, modulating the reader field as done by HF tags is not possible for transmitting data from tag

to reader. Hence, a mechanism called backscatter modulation is used instead by UHF tags [204]. By changing the amount of energy reflected by the tag antenna, data is transmitted to the reader. As reported by Oren and Shamir in 2007 [141], the reflected signal contains not only the intended component for communicating with the reader, but also an unintended part named *parasitic backscatter*. The unintended part of the reflected signal relates to the power consumed by the tag and can be used for side-channel analysis attacks.

An important protocol for electromagnetic-coupled RFID systems in the UHF range is the Electronic Product Code (EPC) Generation 2 standard [50]. The EPC Generation 2 standard has also been approved as an ISO standard (ISO 18000-6C [92]) and is planned to be the future replacement for conventional bar codes. The vision of the inventors of the EPC Generation 2 standard is to attach an RFID tag to each individual product. For now, RFID tags are still too expensive to place them on each individual product, rather they are placed on groups of products like pallets. Equipping pallets with RFID tags allows to increase the efficiency and to reduce costs in supply-chain management. The driving force behind the introduction of the EPC Generation 2 standard is EPCglobal which is a not-for-profit organization that has been founded by GS1 in 2003. GS1 has emerged from Uniform Code Council (UCC) and European Article Number (EAN) International which are the two organizations that are responsible for managing the bar code systems. Large distributors such as Wal-Mart, Tesco, and Metro have already integrated RFID technology that uses the EPC Generation 2 standard into their supply-chain management [62].

Usually, RFID tags have to be fairly cheap and therefore can only integrate limited functionality which strongly affects the utilized protocol. Thus, protocols like the EPC Geneneration 2 standard neglect to include cryptographic security. The lack of cryptographic security makes the EPC Geneneration 2 standard vulnerable to various attacks such as cloning or revealing secrets like the kill password. However, when using future UHF tags to prevent valuable goods from forgery, a higher tag price is acceptable. Pharmacy for example is a use case where valuable goods are involved. Another important aspect is the technological progress that allows to integrate more and more functionality to future RFID tags. There exist numerous proposals that deal with enhancing the security of RFID protocols which furthermore enforce to include cryptographic functionality to RFID tags (see [9, 18, 55, 201]). As soon as RFID tags contain cryptographic functionality, vulnerability against side-channel analysis becomes a concern.

5.2 Description of Examined UHF RFID Tags

For analyzing the side-channel leakage, two different types of UHF RFID tags have been used. Firstly a self-made prototype of a UHF RFID tag that operates semi passively, and secondly commercially available UHF RFID tags that operate passively. The self-made prototype which has initially been built to evaluate current UHF RFID protocols has also been found useful for providing the trigger

signal when performing measurements on passive UHF RFID tags.

5.2.1 Description of the UHF Tag Prototype

The first EM measurements presented in this chapter have been done by using a self-made UHF tag prototype. When evaluating and enhancing the security of current UHF RFID protocols, it is helpful to have a programmable UHF RFID tag. A programmable UHF RFID tag can be used to easily integrate additional functionality such as new security mechanisms and new commands. Furthermore, the additional functionality can be verified and tested, showing a proof of concept. Standard UHF RFID tags do not provide the possibility to integrate additional functionality because their functionality is implemented in silicon.

Unlike most UHF RFID tags, the UHF tag prototype operates semi passively. Like a passive RFID tag, a semi-passive RFID tag only uses the RF field of the reader for communication, but uses an extra battery for power supply like an active RFID tag. Our UHF tag prototype is a printed circuit board (PCB) with discrete components. The UHF tag prototype can be divided into four parts: an antenna, an analog front-end, a digital part, and a protocol implementation on the programmable microcontroller. More detailed information about the tag prototype is given in Section 7.1 where we also describe an HF tag prototype.

5.2.2 Description of Passive UHF RFID Tags

In addition to the UHF tag prototype we have also used passive UHF RFID tags that are commercially available. In contrast to the UHF tag prototype, passive UHF RFID tags are completely powered by the RF field of the RFID reader requiring no extra battery. A passive UHF RFID tag consists of an antenna and a microchip that comprises an analog part and a digital part. Typically, the protocol handling is implemented via a state machine in dedicated hardware [59]. Principally, the overall structure of a passive UHF RFID tag is not that different from the structure of the UHF tag prototype, except that the tag prototype is larger in size and uses discrete components.

In order to obtain read ranges of several meters, passive UHF RFID tags should consume very little power. The power consumption of passive UHF RFID tags is in the range of some microwatts [99]. For detecting data-dependent emanation we have used passive UHF RFID tags from various tag vendors. All examined passive UHF RFID tags have shown data-dependent emanation.

5.3 Measurement Setup for UHF RFID Tags

This section describes the measurement setup that has been used to deduce side-channel information from UHF RFID tags. Contactless measurements have been done some centimeters away from the UHF RFID tags in the near field and up to 1 m away from the UHF RFID tags in the far field. For contact-based

measurements, we have separated the tag chip from the antenna and directly connected it with the RF output of the RFID reader. RF field of the RFID reader has been switched on during all the measurements. Initial measurements have detected data-dependent emanation of the UHF tag prototype. Measurements on passive UHF RFID tags could also reveal a significant amount of side-channel leakage.

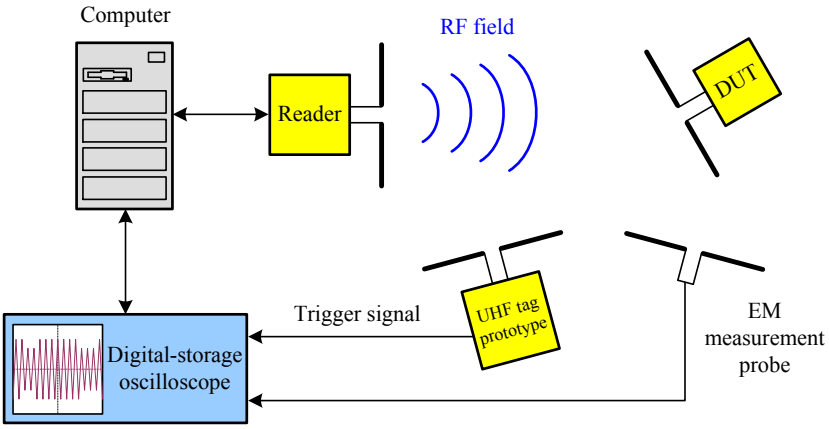


Figure 5.1: Measurement setup for examining the emanation of a passive UHF RFID tag (DUT) in the far field.

Automating the measurement setup is important for performing side-channel analysis. Only an automated measurement setup allows to gather thousands of individual measurements within an acceptable time. Besides the examined UHF RFID tag which we call device under test (DUT), the main components of the measurement setup are a digital-storage oscilloscope that is depicted in Figure 5.4, a UHF RFID reader that is compliant to the EPC Generation 2 standard, and a measurement probe. The digital-storage oscilloscope and the UHF RFID reader are connected to a computer. A program on the computer controls the whole measurement flow and performs the subsequent analysis of the recorded data. Depending on the measurement, different EM measurement probes are used to obtain the side-channel information from the DUT (for contact-based measurements a standard voltage probe is used). Figure 5.1 shows the measurement setup for examining the emanation of a passive UHF RFID tag in the far field.

Acquiring a single measurement follows always the same scheme and requires several steps. After initializing the DUT, the computer sends the command that is used for detecting the data-dependent emanation to the UHF RFID reader, which in turn communicates with the DUT over the air interface. While the DUT processes this command, its radiated EM field (or the power consumption) is recorded by the digital-storage oscilloscope with the help of an EM measurement probe (or a voltage probe in case of contact-based measurements). We have used a sampling rate of 2 GS/s for our measurements. The data acquisition of

Figure 5.2: Near-field probes that have been used for the measurements.

Figure 5.3: Self-made dipole antenna that has been used for the measurements.

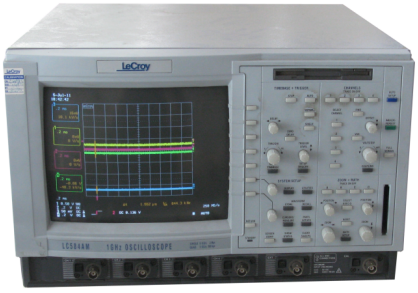


Figure 5.4: Picture of the Lecroy LC584AM digital-storage oscilloscope.

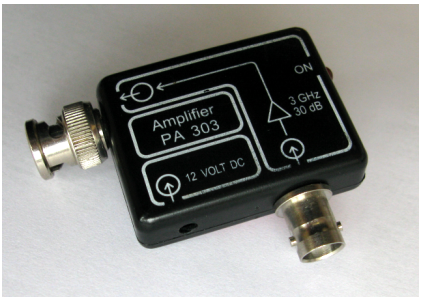


Figure 5.5: Picture of the 30 dB amplifier for the near-field measurements.

the digital-storage oscilloscope is started by a trigger signal. When examining the UHF tag prototype, the trigger signal directly comes from the UHF tag prototype. Passive UHF RFID tags are not suitable for directly providing a trigger signal. Thus, the software of the UHF tag prototype is modified such that it can be placed in parallel to the passive UHF RFID tag into the RF field to provide the external trigger signal (compare Figure 5.1). Acquiring a single measurement is finalized by transferring and storing the recorded data from the digital-storage oscilloscope to the computer.

5.3.1 Near-Field Measurements

For measuring the emanation of UHF RFID tags in the near field, we have used special near-field probes. Near-field probes are available in various sizes and shapes depending on the frequency range and the application they are dedicated for. During our measurements we have used three near-field probes from *Langer EMV* that are shown in Figure 5.2. The probes are designated for detecting magnetic fields. One near-field probe works for frequencies from 100 kHz to

50 MHz (LF B 3), the other two near-field probes work for frequencies from 30 MHz to 3 GHz (RF R 400 and RF B 3-2).

Since the signal amplitudes that can be obtained with near-field probes are rather small, we have deployed an additional preamplifier. The preamplifier that is shown in Figure 5.5 has a voltage gain of 30 dB and is connected between the output of the near-field probe and the input of the digital-storage oscilloscope. When doing measurements in the range of some tenths of megahertz, it is helpful to enable the internal bandwidth limitation of the digital-storage oscilloscope (the Lecroy LC584AM provides a 20 MHz and a 200 MHz bandwidth limitation). Limiting the bandwidth has the advantage that the strong RF field from the UHF RFID reader is suppressed, which furthermore increases the quality of the measurements.

5.3.2 Far-Field Measurements

Near-field probes are no longer suitable for far-field measurements, rather, electromagnetic antennas are required. The UHF RFID tags that have been examined in this work operate at a carrier frequency of about 868 MHz. Since our far-field measurements have concentrated on detecting data-dependent emanation of UHF RFID tags around the carrier frequency that result from parasitic backscatter, no special broadband antenna is necessary. A self-made dipole antenna shown in Figure 5.3 whose length is tuned to the carrier frequency is sufficient. The length of a dipole antenna for a carrier frequency of 868 MHz is about 17 cm [59]. While near-field measurements require an additional preamplifier in order to obtain acceptable signal amplitudes, far-field measurements do not. Figure 5.6 presents an EM trace recorded with our self-made dipole antenna, clearly showing the strong carrier signal from the reader. For transforming the EM trace to baseband, demodulation in software (computing the absolute value followed by low-pass filtering) or in hardware (with an additional spectrum analyzer such as the Rohde&Schwarz ESPI3) can be deployed. Figure 5.7 depicts the EM trace after software demodulation, where the 868 Mhz carrier signal is strongly suppressed. Using demodulation in hardware has the advantage that the sampling rate of the digital-storage oscilloscope can be reduced. Using a lower sampling rate results in measurements that consume less storage space on the computer and that can be analyzed in a faster way.

5.3.3 Contact-Based Measurements

For analyzing the side-channel leakage of passive UHF RFID tags we have also performed contact-based measurements. In order to conduct such measurements, the tag chip has been separated from the antenna and directly connected to the RF output of the RFID reader (*i.e.* antenna of reader has been disconnected) via a shielded cable with a measurement resistor in series. In that way, the tag chip no longer communicates contactlessly by means of the RF field, but through the shielded cable that connects tag chip and reader. Side-channel information that is normally radiated by the tag antenna and which have to be gathered

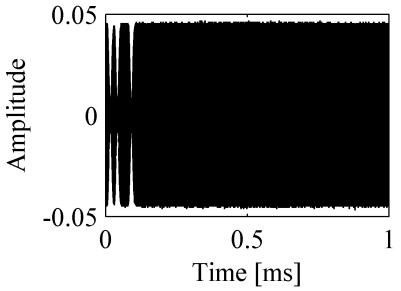


Figure 5.6: EM trace recorded with our self-made dipole antenna.

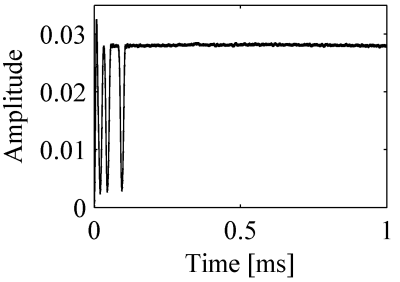


Figure 5.7: EM trace after transformation to baseband using demodulation in software.

with an EM probe can now be directly deduced by measuring the voltage drop across the resistor. A contact-based measurement setup has the advantage that measurements are easier to reproduce since environmental influences (*e.g.* interferences with reflected EM waves of surrounding objects) play no role. On the other hand, contact-based measurements have the drawback that establishing a connection between tag chip and RF output of the reader is required. This makes it necessary to irreversibly modify the tag.

5.4 Side-Channel Analysis Results

For analyzing the susceptibility of UHF RFID tags to side-channel analysis (SCA) we have used the same kind of attacks for both contactless measurements and contact-based measurements. The attacks only differ in the measured side channel. For contactless measurements we have recorded the electromagnetic field emanated by a tag. For contact-based measurements we have directly recorded the power consumption of a tag. When measuring the electromagnetic field, the attacks are called differential electromagnetic analysis (DEMA). When measuring the power consumption, the attacks are named differential power analysis (DPA). Both attacks have the advantage that only a simple model of the analyzed device is necessary and that even very noisy measurements can be used [117]. For more details on SCA we refer to Chapter 3.

Before starting a DEMA or a DPA attack an appropriate operation needs to be selected that is suitable for revealing data dependencies. The UHF RFID tags that we have examined are EPC Generation 2 tags, which are low-cost tags that do not have cryptography implemented. Hence, we had to look for an alternative operation that can be utilized for an attack. For the analyzed UHF tags, it has turned out that the *Write* command is a useful operation to detect data dependencies. The *Write* command as it is defined in the EPC Generation 2 standard [50] allows to write a 2-byte value to the non-volatile memory of a UHF RFID tag. Since the 2-byte value is a freely selectable parameter of the *Write*

command, it can be used as chosen input data of the attack.

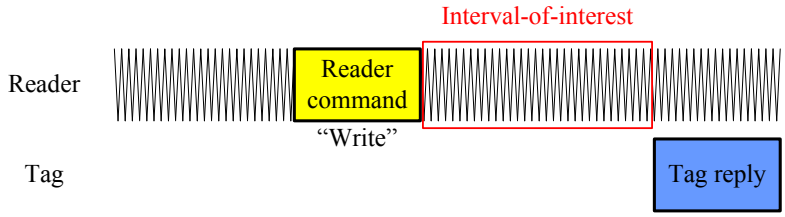


Figure 5.8: Communication between reader and tag when handling a *Write* command. The interval-of-interest marks the time range that is used for recoding EM and power traces.

By using the measurement setup and the measurement-acquisition strategy described in Section 5.3 we have obtained various electromagnetic traces (power traces in case of contact-based measurements). The traces are recorded while the examined UHF RFID tag processes a *Write* command with a chosen 2-byte value. Figure 5.8 shows the communication between reader and tag when handling a *Write* command. The so-called interval-of-interest marks the time range where the *Write* command is processed by the tag and which is used for recording EM and power traces. Thereby, always the same memory location of the UHF RFID tag is used. This memory location is initialized with the value zero before a new chosen 2-byte value is written. Initializing the memory location has the purpose to bring the UHF RFID tag always to the same initial state.

After recording the traces, a hypothetical model is used to map the chosen 2-byte values to hypothetical values that try to predict the side-channel information in the measured traces of the UHF RFID tag. There exist various hypothetical models like the Hamming-weight model or the Hamming-distance model (see Section 3.1.2). We have used the Hamming-weight model for predicting the side-channel information. Taking the 2-byte input values that have been used to obtain the recorded traces results in a hypothesis that is assumed to be correct. Additionally, another several hundred hypotheses are created that are assumed to be wrong. Wrong hypotheses are determined by applying the hypothetical model to randomly chosen values that are different from the 2-byte values that have been used to obtain the electromagnetic traces and the power traces, respectively.

Having all the hypotheses allows to compare them with the traces that have been recorded previously. Comparison is done with the help of statistical methods. A well known statistical method for DEMA attacks and DPA attacks which we have used is the Pearson correlation coefficient (we have given the formula for the Pearson correlation in (3.1) in Chapter 3). The correlation coefficient shows the linear dependency between different values [117]. The higher the absolute value of the correlation coefficient the higher is the linear dependency between the values that are compared. Based on the correlation coefficient, a correlation

trace can be computed for each hypothesis. As mentioned in Section 3.1.2 of Chapter 3, we call a DEMA or a DPA attack successful if only the comparison between the measured traces and the hypothesis that is assumed to be correct leads to significant peaks in the corresponding correlation trace.

5.4.1 Side-Channel Analysis of the UHF Tag Prototype

The UHF tag prototype that we have built and used for side-channel analysis operates semi passively and contains a microcontroller. Compared to a conventional passive UHF RFID tag, the power consumption of the deployed microcontroller is much higher. For any fixed hardware architecture, higher power consumption brings along higher electromagnetic emanation. As shown in the following, we have conducted contactless measurements in the near field and in the far field of our tag prototype.

Results of Near-Field Measurements

For the near-field measurements of our tag prototype, we have placed a small probe (the LF B 3 and the RF B 3-2 probes) on top of the microcontroller to gather its direct emissions. Main part of the electromagnetic field that is emanated by the UHF tag prototype's microcontroller is located in the frequency range of some hundreds of megahertz. Since the strong RF signal of the UHF RFID reader is located around 868 MHz, the RF signal can be easily suppressed by applying a low-pass filter. There are two possible ways for low-pass filtering: directly with the help of the digital-storage oscilloscope during the measurement acquisition, or via software in an additional preprocessing step before performing the DEMA attack.

Suppressing the strong RF signal by using the digital-storage oscilloscope results in electromagnetic traces with smaller amplitudes. As a consequence, a higher input sensitivity can be selected at the digital-storage oscilloscope which increases the accuracy of the measurements. Figure 5.9 shows the result of a successful DEMA attack on the UHF tag prototype in the near-field during the execution of a *Write* command. Recording 1 000 individual electromagnetic traces has lead to a maximum absolute value of 0.63 for the correlation trace of the correct hypothesis. Low-pass filtering has been directly done on the digital-storage oscilloscope during measurement acquisition. For comparison, Figure 5.10 shows the result of the same DEMA attack by doing low-pass filtering of the electromagnetic traces in software. In this case, the maximum absolute value of the correlation trace reduces to about 0.21 (due to lower input sensitivity).

Results of Far-Field Measurements

Besides analyzing the emanation of the UHF tag prototype in the near field, we have also done analysis work in the far field. As mentioned in Section 5.3.2, we have concentrated on measuring the emanation of UHF RFID tags around the

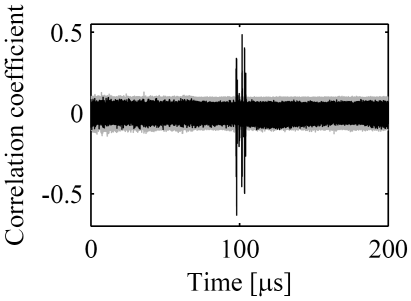


Figure 5.9: Result of the DEMA attack on the UHF tag prototype by doing low-pass filtering directly on the digital-storage oscilloscope.

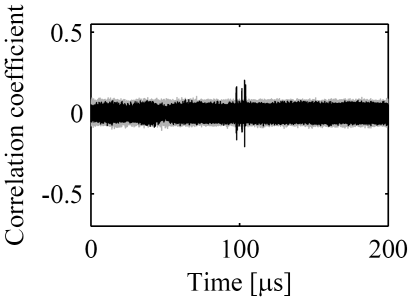


Figure 5.10: Result of the DEMA attack on the UHF tag prototype by doing low-pass filtering via software in an additional preprocessing step.

carrier frequency of the RF signal of about 868 MHz that contains the *parasitic backscatter*. With our measurement strategy we could not detect any data-dependent emanation of the UHF tag prototype in the far field. We assume that the semi-passive operation of the tag prototype (supplied by a battery) and the components between microcontroller and antenna (voltage regulator, large storage capacitors) prevent that enough data-dependent leakage propagates through the *parasitic backscatter*, which we have tried to detect with our far-field measurements.

5.4.2 Side-Channel Analysis of Passive UHF RFID Tags

In contrast to our UHF tag prototype, passive UHF RFID tags have a power consumption of only some microwatts. With our measurement equipment we have not been able to directly measure the electromagnetic field that is emanated by the microchip of a passive UHF RFID tag (*i.e.* by directly placing a small probe like the LF B 3 or the RF B 3-2 on the microchip). Hence, we could only use the *parasitic backscatter* reflected by the tag antenna to detect data-dependent emanation of the microchip. As mentioned before, *parasitic backscatter* occurs when the power consumption of a UHF RFID tag modulates its backscatter [141]. An important aspect of the backscatter of UHF tags is that it can be detected via a simple dipole antenna within several meters. This makes SCA attacks based on the *parasitic backscatter* highly critical from a security point-of-view. For the passive UHF RFID tags we have not only conducted contactless measurements in the near field and in the far field, but also contact-based measurements. Results of the measurements are presented in the following.

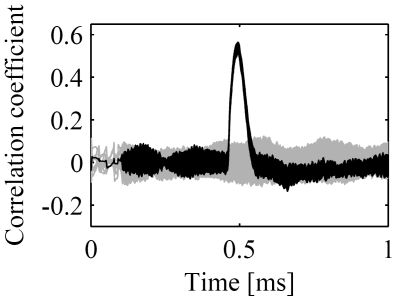


Figure 5.11: Result of the DEMA attack on a passive UHF RFID tag in the near field.

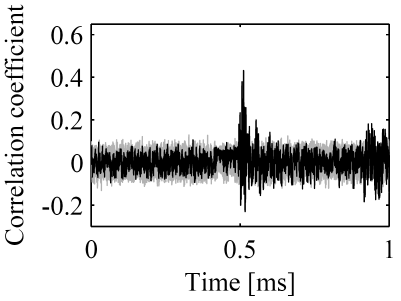


Figure 5.12: Result of the DEMA attack on a passive UHF RFID tag from a different tag vendor in the near field.

Results of Near-Field Measurements

Using the *parasitic backscatter* of passive UHF RFID tags in the near field has allowed to perform successful DEMA attacks. When using a near-field probe, its placement toward the passive UHF RFID tag that is examined is an important factor for the success of the DEMA attack. Favorable placement of the near-field probe stronger attenuates the RF field that is emitted by the antenna of the UHF RFID reader. The stronger the RF field is attenuated the less measurements are necessary for a successful DEMA attack. We have achieved the best results by placing the RF R 400 near-field probe about 3-5 cm away from the tag antenna. This also underlines that we have really measured the *parasitic backscatter* not the direct emissions of the tag's microchip.

In this way we have been able to perform successful DEMA attacks by measuring less than 100 electromagnetic traces. Figure 5.11 shows the result of a DEMA attack on a passive UHF RFID tag by using 1 000 measurements. In order to ensure that this is not a phenomenon of a specific tag vendor, we have tested passive UHF RFID tags from various tag vendors. Figure 5.12 shows the result of the same DEMA attack by using a passive UHF RFID tag from a different tag vendor. Although the two correlation traces in Figure 5.11 and Figure 5.12 are quite different, both illustrate that there is a strong data dependency.

Results of Far-Field Measurements

For the passive UHF RFID tags we have done the same measurements in the far field than for our UHF tag prototype. In contrast to the UHF tag prototype, the passive UHF RFID tags that we have examined show data dependent emanation also in the far field. Thereby, we have analyzed the electromagnetic field with a self-made dipole antenna at various distances of the passive UHF RFID tags, starting from 20 cm up to 1 m.

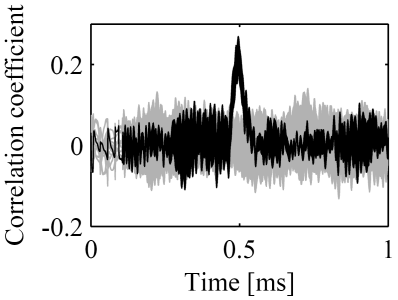


Figure 5.13: Result of the DEMA attack on a passive UHF RFID tag at a distance of 20 cm using 1 000 EM traces.

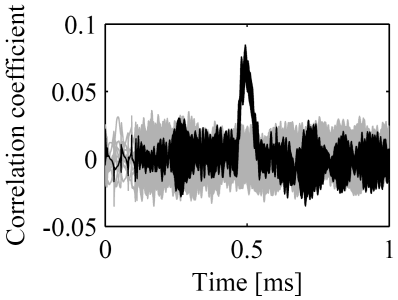


Figure 5.14: Result of the DEMA attack on a passive UHF RFID tag at a distance of 1 m using 10 000 EM traces.

All our DEMA attacks in the far field of the passive UHF RFID tags have been successful, even at a distance of 1 m. Figure 5.13 shows the result of a DEMA attack on a passive UHF RFID tag at a distance of 20 cm using 1 000 measurements. Regardless of the distance, the peaks in the resulting correlation traces always look similar. For comparison, Figure 5.14 shows the correlation traces of the same passive UHF RFID tag at a distance of 1 m. The difference when the distance increases is the maximum absolute value of the correlation coefficient. Figure 5.13 shows a maximum absolute value of the correlation trace of 0.27 which decreases to 0.08 in Figure 5.14. As a consequence, the number of measurements must be increased to clearly identify the data dependency at greater distances. The correlation traces in Figure 5.14 have been obtained by using 10 000 measurements.

Results of Contact-Based Measurements

Another measurement technique that we have applied for detecting data-dependent leakage of passive UHF RFID tags are contact-based measurements. Figure 5.15 shows the photo of a microchip that is separated from the tag antenna and connected to the reader via a shielded cable. Also with this measurement technique, successful attacks have been possible. Figure 5.16 depicts the correlation traces of a DPA attack (since we are now conducting power measurements) using 1 000 measurements. The correlation peak in the correct hypothesis looks quite similar to the one in Figure 5.11, which results from the same tag. Also the maximum absolute value of the correlation coefficient is quite the same. This illustrates that contact-based measurements are an interesting alternative to contactless measurements with an EM probe. However, contact-based measurements require a modification of the tag.

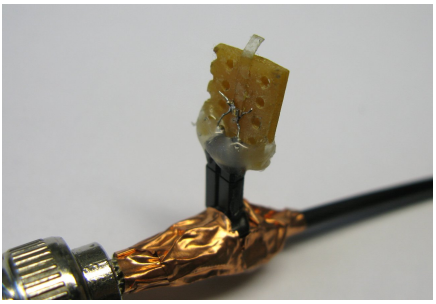


Figure 5.15: Photo of a microchip that is separated from the tag antenna and connected to the reader via a shielded cable.

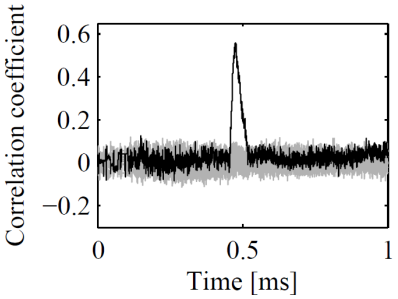


Figure 5.16: Result of the DPA attack on a passive UHF RFID tag using a contact-based measurement technique.

5.5 Summary

In this chapter we have shown the susceptibility of UHF RFID tags to side-channel analysis. We have analyzed a self-made UHF tag prototype and commercially available passive UHF RFID tags from various tag vendors. By using a contactless measurement setup, we have gathered the EM emissions of the self-made UHF tag prototype and the commercially available RFID tags in the near-field as well as the far field. For the commercially available RFID tags, we have also utilized a contact-based measurement setup. A summary of the side-channel analysis results that we have achieved is presented in Table 5.1. Whereas the UHF tag prototype that operates semi passively shows only data-dependent emanation in the near field (using direct emissions of the microcontroller), passive UHF RFID tags show data-dependent emanation in the far field too. We have performed successful DEMA attacks in the far field of passive UHF RFID tags at distances up to 1 m by measuring their *parasitic backscatter*. However, increasing the number of acquired measurements should allow to realize successful DEMA attacks at greater distances as well. Also the contact-based measurements that we have applied on the passive UHF RFID tags have been successful.

Current low-cost UHF RFID tags do not use cryptographic protection and furthermore store no secret that could be the aim of such attacks. Hence, this work has no practical relevance for current RFID products. Nevertheless, it was our goal to investigate the side-channel leakage and to determine the susceptibility of future UHF RFID tags to this class of attacks. Our results clearly show that once cryptographic functionality is added to UHF RFID tags, countermeasures against SCA attacks need to be applied.

Table 5.1: Summary of the side-channel analysis results.

Device under test	**Contactless measurement**		**Contact-based measurement**
-	**Near field**	**Far field**	-
Tag prototype	Successful DEMA attacks using direct emissions	DEMA attacks failed	-
Commercially available tags	Successful DEMA attacks using *parasitic backscatter*	Successful DEMA attacks using *parasitic backscatter*	Successful DPA attacks

6

Fault Analysis of Low-Cost RFID Tags

The second kind of implementation attack that we have applied on commercially available low-cost RFID tags is fault analysis which we describe in this chapter. As previously mentioned, fault analysis is an implementation attack that tries to reveal secret information of a device by provoking an abnormal behavior. Such an abnormal behavior can either be achieved by manipulating the input signals, the operating environment, or even the device itself.

We have applied global as well as local fault-injection methods on passive low-cost RFID tags that are commercially available. The tags are in form of so-called adhesive labels where the tag antenna a printed on a flexible carrier material. We have analyzed HF and UHF tags from various tag vendors. The analyzed tags neither include cryptographic security nor countermeasures. We have used the write operation of the tags to their internal non-volatile memory for conducting fault attacks, since this operation is highly critical in terms of power consumption and execution time (*i.e.* causes high power consumption and takes quite a long time). In that way, we have used different fault-injection methods to interrupt or disturb the write operation executed by the tags. The fault-injection methods that we have applied are: temporarily antenna tearing, electromagnetic interferences, and optical inductions. Temporarily antenna tearing has been achieved by shortly interconnecting the antenna pins of the tag. For generating electromagnetic interferences a high-voltage generator has been used. Optical inductions have been applied with the help of a laser diode that has either been used to directly illuminate the tag chip (global fault injection) or has been focused on the tag chip with a microscope (local fault injection). All evaluated tags have shown vulnerabilities to fault analysis. We have been able, for example, to prevent writing of data, interrupt writing of data, and writing of faulty data without being detected by reader or tag.

Similar to the side-channel analysis attacks presented in Chapter 5, almost no information have been available about fault analysis on RFID tags when we started our work. Again, fault analysis has been a well researched topic to attack cryptographic devices like smart cards [11, 12, 19, 105, 159, 172, 178], but information on contactless devices such as RFID tags has been missing. Our fault-analysis results on commercially available RFID tags have been published at the CHES 2008 conference [80]. It is a joint work with Michael Hutter and Jörn-Marc Schmidt and has been the first published paper that focuses on fault analysis on passive RFID tags and that further presents practical examples. Most of the information given in this chapter relies on the work presented in this paper.

The reminder of this chapter is organized as follows. Section 6.1 describes protection mechanisms of passive RFID tags. Fault-analysis techniques that are suitable for passive RFID tags are presented in Section 6.2, followed by the description of the evaluated tags and the conducted fault analyses in Section 6.3. In Section 6.4 we give information about the deployed measurement setups, and Section 6.5 provides the fault-analysis results that we have achieved. The chapter ends with a summary in Section 6.6.

6.1 Protection Mechanisms of Passive RFID Tags

Contactless devices like passive RFID tags have different requirements than contact-based devices. Passive RFID tags have to cope with surrounding noise and strongly varying reader-field strengths, *i.e.* they have to properly work close to the reader (very strong field) and at a certain distance from the reader (rather weak field). Surrounding noise can disturb the communication channel between reader and tag. Hence, error-correction mechanisms like cyclic redundancy checks (CRC) and parity bits are typically integrated to detect corrupted data in the communication [50, 93]. When tags are passively supplied, they have to be prepared that the reader field and thus also their power supply is instantaneously teared off. Especially operations like writing of data to the tag's internal non-volatile memory are highly critical to unexpected power-supply losses since they require rather high power consumption and have long execution times (in the range of several milliseconds). In order to prevent such effects, RFID tags can integrate so-called anti-tearing mechanisms [14]. There, data is first written to a temporary buffer (also non-volatile memory) before it is written to the final destination in the memory (*e.g.* EEPROM or flash). When the power supply is interrupted during a write operation, inconsistencies in the temporary buffer and the final destination are detected at next power up and resolved by restoring the data from the temporary buffer. An alternative method to detect inconsistencies in the internal memory of a tag is to append, for example, a CRC to each data word. However, both methods lead to increased memory requirements and longer write times.

Another protection mechanism of passive tags is to limit the access to certain tag resources like special configuration parameters and memory regions with sensitive data. Often, tag resources are only protected by means of a password that has to be transmitted whenever access needs to be granted. EPC Generation 2 tags [50] for example use two passwords: an access password for limiting the access to parts of the tag memory and a kill password for permanently deactivating the tag. Since password-based approaches are highly vulnerable to replay attacks, they are only a rather weak protection mechanism. More-advanced tags use challenge-response approaches and symmetric as well as asymmetric cryptography that provide much better protection. Examples are Mifare [137], SecureRF [166], or CryptoRF [14] tags. However, such approaches are much more expensive and thus typically omitted by low-cost RFID tags.

6.2 Fault-Analysis Techniques Suitable for Passive RFID Tags

Before applying fault-analysis techniques, having preliminary knowledge about the general structure of RFID tags is advantageous. When manufacturing RFID tags, mainly two approaches are used to connect the tag chip with the antenna: direct-chip attach and chip-strap attach. As the name says, direct-chip attach takes the small tag chip itself and directly mounts it onto the antenna. This approach requires precise handling of the tag chip. For the chip-strap attach method, the tag chip is first mounted on a small interconnect adhesive or printed circuit board (PCB) called *strap* to ease further handling of the tag chip. Afterwards, the *strap* is then mounted onto the antenna. Figure 6.1 shows the cross section of a tag where the chip is mounted via direct-chip attach onto the antenna. Materials used for the antenna are mainly copper and aluminum that are either etched or printed. Carrier material for the antenna is typically a Polyethylene Terephthalate (PET) film (especially in case of adhesive labels). Often, also a special ink layer is inserted between tag chip and antenna [66].

As described in Chapter 3, fault-analysis techniques can be applied globally on the whole tag chip as well a locally on parts of the tag chip (*e.g.* only on a specific part of the memory). Further, fault-analysis techniques can be either non-invasive, semi-invasive, or invasive. Non-invasive techniques leave the chip untouched, whereas semi invasive and invasive techniques require to decapsulate the chip. Invasive attacks also electrically contact the decapsulated chip. According to this classification scheme, modifying a tag by separating the chip from the antenna and supplying it with external signals is still a non-invasive technique. Further, when the tag is realized as adhesive label where the surface of the tag chip is only covered by a transparent carrier layer, decapsulating the tag chip might not even be necessary. In the following we give a short overview of the fault-injection techniques that seem to be suitable for attacking passive RFID tags.

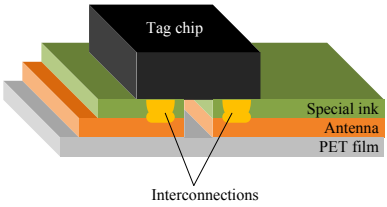


Figure 6.1: Cross section of a tag where the chip is mounted onto the antenna via direct-chip attach.

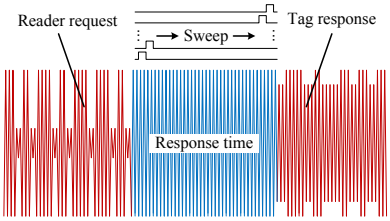


Figure 6.2: Indication of the response time that is used by the tag for processing the write operation.

6.2.1 Temperature Variations

When increasing the operating temperature of passive RFID tags, properties of the tag circuit (especially of the analog part) change. Experiments with commercially available tags have shown that writing data into the internal tag memory is no longer possible around temperatures of 180 °C. Further increasing the temperature leads to a complete break down of the communication between reader and tag, *i.e.* the tag remains silent and does not answer to reader requests. When stressing tags with high temperatures, it is advantageous to separate the tag chip from the antenna to prevent deformation of the carrier film. Temperature variations can only be applied in a global manner.

6.2.2 Supply Voltage and Clock Variations

The tag chip has typically only two input pins. These two pins are connected to the antenna and provide the tag chip with the power supply and potentially also with the clock signal. For generating supply-voltage variations, the input pins of the tag chip can be temporarily interconnected. No power is provided to the tag chip as long as the input pins are interconnected. We call this effect temporarily antenna tearing, which originates from card tearing that is used to attack contact-based smart cards [75]. Whether clock variations can be applied to disturb the behavior of passive tags depends on the tag type itself. HF tags mainly extract their clock signal from the carrier signal of the reader field which they receive via their antenna. In such a case, glitches in the carrier signal of the reader (*e.g.* by temporarily increasing the carrier frequency) could be generated to affect the behavior of a tag. For UHF tags, clock variations might be rather difficult to achieve, since they often use their own clock signal derived from an internal oscillator circuit. Both supply-voltage variations and clock variations are global fault-injection techniques.

6.2.3 Electromagnetic Interferences

Passive RFID tags are designed to cope with varying field strengths (*e.g.* when they are located close to the reader antenna) and thus have integrated special protection circuits to prevent overloading of the analog front-end. However, the protection circuits only work to a certain reader-field strength. As reported in [2], strong EM pulses can be utilized to completely destroy a tag. The frequency range for which a tag is susceptible mainly depends on the tag antenna. Consequently, UHF tags that are receptive to frequencies around 900 MHz are considered to be more sensitive to electromagnetic interferences than for example HF tags that operate at much lower frequencies [23].

6.2.4 Optical Inductions

In order to apply optical inductions, intervisibility to the analyzed areas of the tag chip is required. When analyzing RFID tags that are manufactured as adhesive labels, no special decapsulation procedure is necessary. Intervisibility to the tag chip can be easily achieved by scratching off for example the PET film and by carefully removing the adhesive via chemicals. In cases where the tag chip is only covered by a transparent PET film, no further modification of the tag is required to apply optical inductions. An advantage of optical inductions is that they cannot only be applied globally on the whole chip, but also in a local manner by focusing on parts of the chip.

6.3 Description of Evaluated Tags and Conducted Fault Analyses

We have evaluated the susceptibility of commercially available RFID tags to fault analysis. The evaluated tags are passive low-cost tags operating either in the HF range or the UHF range. HF tags are using the ISO 15693 [89] standard and UHF tags the ISO 18000-6C [92] standard (EPC Generation 2 standard). Data rate of the HF tags has been 26.48 kbps for reader-to-tag and tag-to-reader communication. Data rate of the UHF tags has been 26.67 kbps for reader-to-tag communication and 40 kbps for tag-to-reader communication. Since none of the tags provide cryptographic operations, we have selected the writing to the internal non-volatile memory of the tags for inducing faults instead. Writing to the non-volatile memory is a critical operation in terms of power consumption and execution time.

Inducing a fault always followed the same principle. First, an appropriate write command has been sent to the tag via the RFID reader. For HF tags we have used a TAGscan multi ISO reader from Tagnology and for UHF tags an A828EU compact reader from CAEN. While the tag is executing the write command, induction of a fault has been triggered. As illustrated in Figure 6.2, induction of the fault occurs during the so-called response time of the tag (compare Figure 5.8 in Chapter 5 where we used a similar time range for the DEMA

attacks). The response time is defined as the time between the end of the reader command and the beginning of the tag response. Length of the response time depends on several factors like data rate, protocol, and writing time of the tag memory. For the evaluated tags, we have observed response times in the range of a few milliseconds (writing time itself took only a few hundred microseconds). Since we had no information at which point in time the write operation actually takes place, we have conducted automatic fault-injection sweeps. Hence, we have varied the point in time where the fault has been induced step-by-step over the whole response time. Additionally, we have also varied the duration of the fault injection to have a precise control over the intensity of the induced fault. As fault-injection methods, power and clock variations, electromagnetic interferences, and optical inductions have been applied. In the next section, details about the measurement setups for the different fault-injection methods are provided.

6.4 Measurement Setups for Fault Analysis

Different measurement setups have been used for injecting faults on passive RFID tags. The setups mainly consist of a PC, an RFID reader, a tag prototype, the fault-injection device, and the device under attack which is the analyzed tag. Central element of the measurement setup is the PC that controls all other components. By running MATLAB scripts on the PC, fault injections can be conducted in an automated way. For HF measurements, an HF reader with a maximum field strength of about 400 mW has been chosen. UHF measurements have used a UHF reader with a field strength of about 60 mW. In order to provide appropriate trigger events for fault injection, we have placed a tag prototype inside the reader field together with the device under attack to eavesdrop the reader-to-tag communication. The tag prototype is a semi-passive tag that consists of a PCB with an integrated antenna and an analog front-end as well as a digital part with a programmable microcontroller. We have used separate tag prototypes for analyzing HF and UHF tags (details about the deployed tag prototypes can be found in Section 7.1). The tag prototype is connected to the PC via a serial interface and can be configured to release a trigger event with a defined offset time and a defined duration time (with a resolution of about 300 ns). This allows to perform precise fault-injection sweeps over the whole response time of the analyzed tag. The trigger signal from the tag prototype is connected to the concerning fault-injection device. Depending on the measurement setup, different fault-injection devices are deployed. For temporarily antenna tearing we have used an optocoupler circuit. For electromagnetic interferences a high-voltage generator has been utilized. Optical inductions have used a laser diode (and a microscope when performing local fault injections). In the following, we describe the different fault-injection measurement setups for global and local fault analysis in more detail.

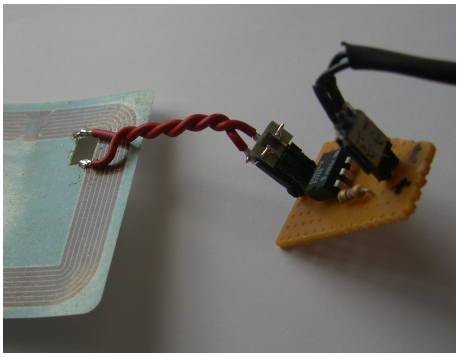


Figure 6.3: Picture of the optocoupler circuit that has been inserted between tag chip and antenna.

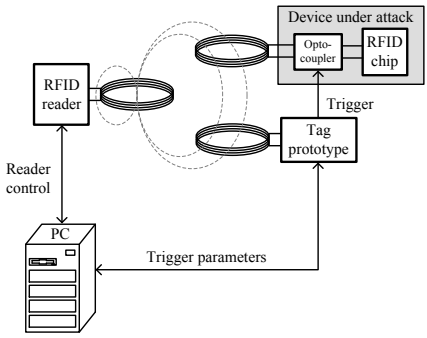


Figure 6.4: Schematic view of the measurement setup for performing antenna-tearing attacks using an optocoupler that is placed between tag antenna and chip.

6.4.1 Measurements Setups for Global Fault Injections

Three different measurement setups have been used for global fault injections. An advantage of global fault injections is that no detailed knowledge of the geometric structure of the analyzed chip is required. Fault injections are only applied to the whole chip and not to parts of it. Hence, no expensive equipment for precisely handling and probing the chip is required. The only important aspect is accurate triggering of the fault injection, which is typically rather easy to achieve.

Temporarily Antenna Tearing

We have separated the tag chip from its antenna for conducting fault injections via temporarily antenna tearing (in a similar way as shown in [31]). As depicted in Figure 6.3, an optocoupler circuit has been inserted between tag chip and antenna. The optocoupler allows to temporarily interconnect the two antenna pins and gets its input signal from the tag prototype described above. By using an optocoupler, tag prototype and antenna circuit are galvanically isolated. This has the advantage that damage of the tag prototype due to high voltages induced in the tag antenna is prevented, and further that feedback from the tag prototype to the antenna circuit is minimized (*e.g.* to avoid de-tuning of the antenna circuit). A schematic view of the measurement setup that we have used for temporarily antenna tearing is shown in Figure 6.4. The PC controls RFID reader, tag prototype, and the device under attack. As aforementioned, both tag prototype and device under attack are placed inside the reader field. For the

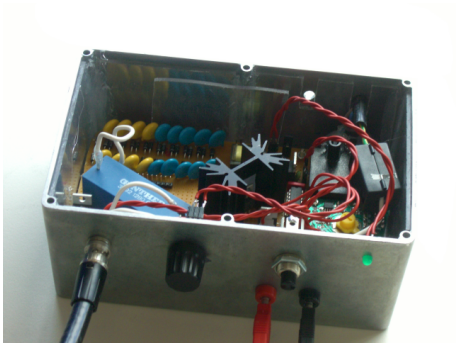


Figure 6.5: Picture of high-voltage generator with the cover of the case removed.

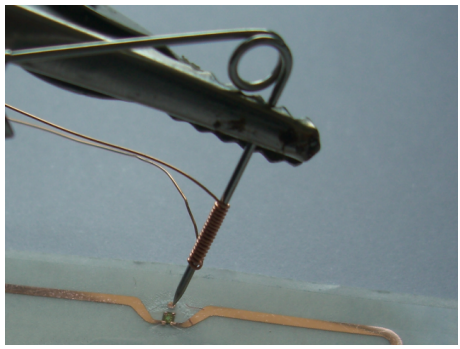


Figure 6.6: Probe coil with needle that is directly placed above the tag chip.

UHF measurements, tag prototype and device under attack have been placed about 10 cm away from the reader antenna, for the HF measurements they have been placed close to the reader antenna.

Electromagnetic Interferences

For generating electromagnetic interferences, a high-voltage generator has been built that can produce voltages up to 18 kV. The high-voltage generator has to be handled with caution, since high voltages can be very dangerous. In the first stage of the high-voltage generator, a square-wave signal with an amplitude of approximately 100 V is generated. This square wave is then stepped up to a high voltage in the second stage by using a DC voltage converter and a charge-pump circuit. In order to prevent damage of electronic devices in the proximity of the measurement setup, we have placed the high-voltage generator inside a shielded aluminum case that has been connected to earth ground. A picture of the high-voltage generator with the cover of the case removed is shown in Figure 6.5. The output of the high-voltage generator is connected to a probe coil over a high-voltage shielded cable. The probe coil is wound around a needle and is directly placed above the tag chip as depicted in Figure 6.6. The rest of the measurement setup is similar to the one used for temporarily antenna tearing. When the high-voltage generator receives a trigger signal from the tag prototype, a fast-changing current pulse is generated within the coil. This fast-changing current pulse induces eddy currents into the chip, which can be used to influence the behavior of the tag chip. In contrast to temporarily antenna tearing, electromagnetic interferences require no modification of the tag.

Optical Inductions

The third measurement setup that we have used for global fault injections uses a simple laser diode for producing optical inductions. The laser diode has a

optical output power of 100 mW and operates at a wavelength of 785 nm. We have placed the laser diode directly above the tag chip as illustrated in Figure 6.7. When laser light hits the surface of the chip, a current is induced (Optical Beam Induced Current) that causes transistors to switch. The laser diode is controlled by a small electronic circuit that uses the trigger signal from the tag prototype. All other components of the measurement setup are the same as in the setups described above. As mentioned in Section 6.2, many RFID tags that come as adhesive labels use a transparent PET film as carrier for the antenna. Hence, no further modification of such tags is required to apply optical inductions, since the tag chip is only covered by the transparent film that is permeable for the laser light. In that way, optical inductions provide a rather convenient way to induce global faults and are much less dangerous then the aforementioned electromagnetic interferences that require a special high-voltage generator.

6.4.2 Measurement Setup for Local Fault Injections

In contrast to global fault injections that affect the whole tag chip, local fault injections allow to concentrate on parts of the chip for inducing a fault (*e.g.* a dedicated data block in the memory). We have used optical inductions to conduct local fault injections. The deployed measurement setup is similar to the one that has been used for global fault injections via optical inductions, except that an additional microscope is utilized. The microscope is responsible for focusing the laser beam and accurately positioning the tag chip. We have mounted the laser diode (the same diode is used for global fault injections) on the camera port of the microscope. The laser beam of the diode is first parallelized by a collimator lens and then focused by the optical objective of the microscope, which has a magnification of 50 diameters. Figure 6.8 shows a picture of the measurement setup where the laser beam is focused on the chip of an HF tag. We have placed the tag above the reader antenna, together with the tag prototype that is deployed for generating the trigger signal.

6.5 Fault-Analysis Results

After the description of the measurement setups in the section above, we now provide the fault-analysis results that we have achieved with these setups. We have injected faults during writing to the internal memory of passive low-cost tags. HF as well as UHF tags from different tag vendors have been analyzed. All evaluated tags have been susceptible to fault injections during writing to memory.

Mainly, we have observed five fault types during fault injection. The different fault types are listed in Table 6.1. Every tag has its own writing characteristic, *i.e.* start time of writing, write duration, writing strategy (*e.g.* erase before write) and further also shows different sensitivity to fault injections. Hence, offset time and duration of the trigger signal for successfully inducing a fault has been different for individual tags. However, after determining appropriate trigger-

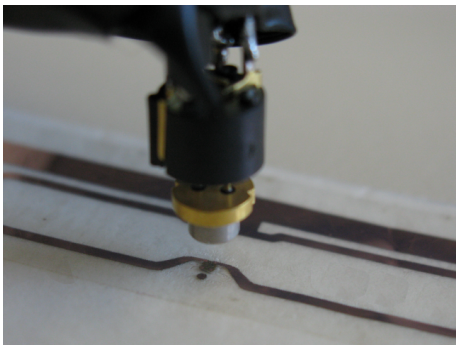


Figure 6.7: The laser diode is directly placed above a tag chip that is only covered by a transparent PET film.

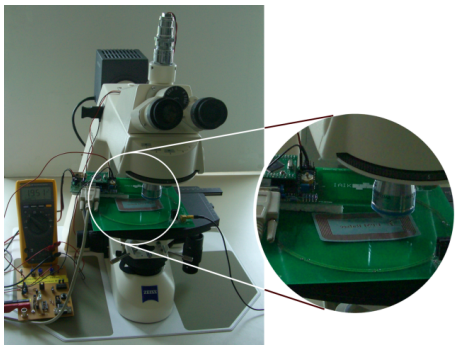


Figure 6.8: Measurement setup for local fault injections via optical inductions using a microscope.

signal parameters, fault injections have been easily reproducibly and successful for all analyzed tags. Using automatic fault-injection sweeps helped finding such appropriate trigger-signal parameters within rather short time. The first two fault types that we have observed are *Unconfirmed Lazy Write* and *Unconfirmed Successful Write*. During an *Unconfirmed Lazy Write*, the tag neither writes the new value to its internal memory nor does it send a response after the write operation. The *Unconfirmed Successful Write* on the other hand writes the new value to its internal memory but also provides no response after writing. Both fault types can also occur during normal tag operation (*e.g.* tag does not receive enough power from the field) and thus are typically somehow handled by the RFID communication protocol. The ISO 18000-6C protocol [92], for example, uses a time-out mechanism to detect non-responsive tags and allows tags to send error messages that indicate whether they have enough power to complete a write operation or not.

Table 6.1: Overview of the different fault types with the resulting EEPROM values after the write operation and the estimated thread level.

Fault type	**EEPROM value**	**Threat level**
Unconfirmed Lazy Write	old	low
Unconfirmed Successful Write	new	low
Unconfirmed Faulty Write	influenced by adversary	medium
Confirmed Lazy Write	old	medium
Confirmed Faulty Write	influenced by adversary	high

The other three fault types that we have observed are much more critical and typically are not considered by RFID communication protocols. During an *Unconfirmed Faulty Write*, a different value is written to the internal memory

than specified by the write operation. No response is sent by the tag. The value that is written is not random and can be largely influenced by the trigger parameters (offset time and duration) of the fault injection. When the tag performs a *Confirmed Lazy Write*, a response is sent by the tag indicating that the write operation has been successful, but not write operation takes place and the old value remains in the internal memory. The last and most critical fault type that we have obtained is the *Confirmed Faulty Write*. Similar to the *Unconfirmed Faulty Write*, a different value is written to the internal memory that can be influenced by the trigger parameters of the fault injection, but the tag also confirms that the write operation has been successful by sending an appropriate response.

In the following, we give more details about the results that we have achieved with global as well as local fault-injection techniques.

6.5.1 Global Fault Injections

We have successfully conducted global fault injections using temporarily antenna tearing, electromagnetic interferences, and optical inductions. Regardless of the applied fault-injection technique, similar results have been obtained. In order to thoroughly analyze the tags automatic fault-injection sweeps have been performed by varying offset time and duration of the trigger signal for the fault injection. Hereafter, we describe first the influence of the duration of the trigger signal on the achieved results, followed by the influence of the offset time.

When the duration of the trigger signal and thus also the duration of the fault injection has been chosen too short, no influence on the write operation of the tag has been observed. Selecting the duration of the fault injection sufficiently long has led to the effect that the tags have performed a reset and thus did not respond to the write command from the reader. The minimum fault-injection duration that is required to cause a reset depends on several factors like, tag type, strength of the reader field, reading distance between reader and tag, and fault-injection technique. When for example a stronger reader field is used, longer fault-injection durations have to be selected to force a reset. The fault-injection duration has only been relevant for temporarily antenna tearing and optical inductions. There, a fault-injection duration of about 100 µs has been sufficient to force a reset. For electromagnetic interferences, no variation of the fault-injection duration has been possible, since the electromagnetic-discharge process is very fast and completes within several nanoseconds. However, depending on the distance between probe needle and tag chip, a single discharge pulse has been enough to reset the tag.

The second parameter that we have varied during fault injection is the offset time. In that way we have been able to sweep through the whole response time of the tag. Depending on the offset time that has been selected to reset the tag via a fault injection, three different fault types have been obtained as illustrated in Figure 6.9. Choosing a rather short offset time has led to an *Unconfirmed Lazy Write*. The tag performs a reset before the actual write operation inside the tag starts, the old value is still present in the memory. Selecting a large

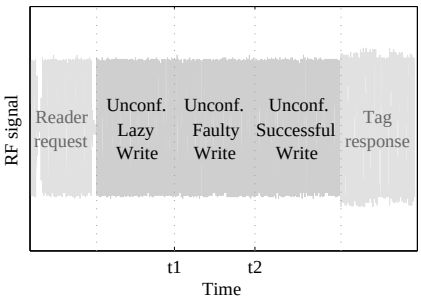


Figure 6.9: Overview of the different fault types that occurred during global fault injection.

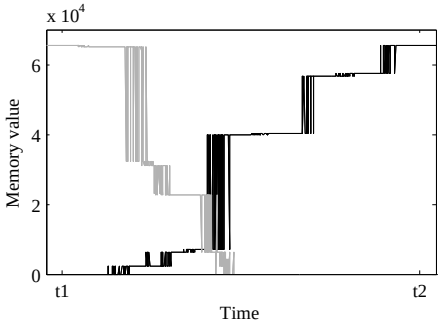


Figure 6.10: Memory content during *Unconfirmed Faulty Write* at different points in the response time when writing the values $0x$FFFF (black trace) and $0x0000$ (gray trace).

offset time has resulted in an *Unconfirmed Successful Write*. The tag performs a reset after the write operation inside the tag has taken place, the new value is stored in the memory. However, when properly selecting the offset time such that the fault injection occurs exactly during the write operation of the tag, an *Unconfirmed Faulty Write* can be achieved. As shown in Figure 6.10, the faulty values that are written to the tag memory can be largely influenced by the offset time of the fault injection. The black trace presents the result of a fault-injection sweep where a 16-bit memory location has been first initialized by writing zero ($0x0000$) to it, followed by writing a value with all bits set to one ($0x$FFFF) to it during which the fault-injection has been provoked. The gray trace has been obtained by using the opposite values for the write operations. First, the memory location has been initialized with a value containing only ones, followed by writing zero to it. As the two traces clearly point out, data is written serially to the memory (bit-by-bit) within the time range $t1$ to $t2$. The larger the step in the trace, the more left is the position of the affected bit in the 16-bit memory location. An interesting observation is that the bits are not sequentially written in ascending order (*i.e.* from the least-significant bit to the most-significant bit) or in descending order (*i.e.* from the most-significant bit to the least-significant bit), rather another (fixed) sequence is used. Different sequences have been observed for different tags.

With the automatic fault-injection sweeps, parameters like the start time, the write duration, and the write strategy (*e.g.* whether a separate erase cycle is used or not) of a the write operation can be easily deduced within minutes. These parameters can be deployed to characterize tags (fingerprinting).

Experiments have shown that optical inductions can be utilized to achieve even more powerful fault types. By using optical inductions, we have also ob-

tained the fault types *Confirmed Lazy Write* and *Confirmed Faulty Write* where the tag acknowledges the write operation by sending a response. In order to induce such faults, duration of fault injection (*i.e.* time where laser diode illuminates the chip surface) has to be adjusted very precisely. When the fault-injection duration is adjusted properly, the tag has still enough power to send the tag response, but not enough power to complete the write operation. Depending on the utilized offset time, a *Confirmed Lazy Write* is achieved when inducing the fault before the write operations starts and a *Confirmed Faulty Write* when inducing the fault during the write operation. These faults are rather powerful, since no indication is given by the tag that the write operation is incomplete or has failed. Further, such faults could even be used to bypass anti-tearing mechanisms as described in [14] where data is temporarily stored by inducing multiple faults (can be done because the tag performs no reset), or to prevent incrementing/decrementing of a counter value that limits for example the number of authentications a tag is allowed to perform.

6.5.2 Local Fault Injections

Optical inductions are not only a powerful technique for inducing global faults, but can also be used to generate local fault injections. By utilizing a microscope to focus and precisely position the laser beam, we have induced local faults in different regions of the tag chips. The achieved fault types mainly depend on the location of the fault injection and on the intensity of the laser beam. All the five fault types from global fault injection have also been obtained with local fault injection.

When the intensity of the laser beam is chosen too high or the diameter of the beam is too large, the tag performs simply a reset and sends no response. Hence, *Unconfirmed Lazy Write*, *Unconfirmed Successful Write*, and *Unconfirmed Faulty Write* can be generated with this method. Properly adjusting the laser beam intensity has allowed us to provoke also *Confirmed Lazy Write* as well as *Confirmed Faulty Write*. Especially the memory-control logic has turned out to be a good location for fault injection. Our experiments have further shown that precise positioning and focusing of the laser beam are much more important for local fault injection than accurate timing. We have generated for example *Confirmed Faulty Write* without accurately triggering the fault injection, just by focusing the laser beam on the proper region of the chip. This makes local fault injections easier to conduct, which comes at expense of higher equipment cost for the inductions.

6.6 Summary

In this chapter we have evaluated the susceptibility of passive low-cost tags to fault analysis. We have analyzed HF and UHF tags from various tag vendors. All analyzed tags have shown vulnerability to fault injections. Global as well as local fault injections have been applied. Global fault injections have been conducted

by using temporarily antenna tearing, electromagnetic interferences, and optical inductions. For local fault injections, optical inductions have been used. Aim of the fault injections has been the writing of data to the internal memory of the tags. Five different fault types have occurred during the experiments: *Unconfirmed Lazy Write*, *Unconfirmed Successful Write*, *Unconfirmed Faulty Write*, *Confirmed Lazy Write*, and *Confirmed Faulty Write*. The first three fault types are obtained by resetting the tag at different points in time within the response time. We have been able to generate these fault types with all applied fault-injection techniques, although electromagnetic interferences turned out to be difficult to reproduce (mainly due to inaccurate timing behavior of our measurement setup). The fault types *Confirmed Lazy Write* and *Confirmed Faulty Write* are more powerful since the tag performs no reset and thus can still send a response to the write operation. We have achieved these fault types only with optical inductions, globally as well as locally. Global fault injections require precise timing, whereas local fault injections presuppose proper positioning and focusing of the laser beam. Local fault injections provide the highest success rate but require also more expensive equipment (microscope). A summary of the different fault types that have occurred together with their fault-reproducibility rate is given in Table 6.2.

None of the evaluated tags have included cryptography nor fault-analysis countermeasures. However, aim of the work presented in this chapter has been to find out whether fault analysis poses a potential thread for passive RFID tags. If once cryptography is integrated into such tags, proper measures have to be taken to prevent these kind of attacks. Our results have shown that low-cost equipment suffices to write faulty values to the internal memory of tags. The faulty values are not random, but can be largely influenced by an adversary by properly injecting the fault.

Table 6.2: Summary of the occurred fault types and their fault-reproducibility rate.

	Antenna tearing	**Electromagnetic interferences**	**Optical inductions**	
Fault type	*global*	*global*	*global*	*local*[a]
Unconfirmed Lazy Write	$> 95\,\%$	$< 10\,\%$	$> 95\,\%$	$> 95\,\%$
Unconfirmed Successful Write	$> 95\,\%$	$< 10\,\%$	$> 95\,\%$	$> 95\,\%$
Unconfirmed Faulty Write	$> 95\,\%$	$< 10\,\%$	$> 95\,\%$	$> 95\,\%$
Confirmed Lazy Write	—	—	$> 90\,\%$	$> 95\,\%$
Confirmed Faulty Write	—	—	$> 90\,\%$	$> 95\,\%$

[a]Local optical inductions require that the laser beam is properly focused and positioned onto the tag chip.

7

Evaluating the Effectiveness of Randomization as a Countermeasures for RFID Devices

As we have shown in the previous chapters, not only contact-based devices but also contactless devices like passive RFID tags are highly susceptible to implementation attacks. In order to prevent implementation attacks, countermeasures have to be integrated. In this chapter, we evaluate the effectiveness of a randomization countermeasure on algorithmic level to make low-cost RFID tags less susceptible to side-channel analysis (SCA) attacks. The chapter afterwards concentrates on the evaluation of the detached power-supply countermeasure that is aimed for protecting passive UHF RFID tags against parasitic-backscatter attacks.

For protecting cryptographic devices against side-channel analysis attacks based on power and EM measurements, two basic countermeasure approaches are deployed: hiding and masking [117]. A very efficient way of implementing hiding, especially for low-resource devices like RFID tags, is to randomize the execution of the algorithm. This means that the performed operations of the algorithm occur at different points in time in each execution. Randomization can be done by shuffling and by randomly inserting dummy cycles (see Section 4.1.2). The reason why randomization is very cost efficient in terms of hardware resources is that the implementation is mainly done in the control logic. Moreover, spending additional clock cycles for randomizing the execution of the algorithm is convenient since the data rates used in RFID systems are rather low.

In this chapter we apply several SCA attack techniques to an implementa-

tion of the Advanced Encryption Standard (AES) [131] that has randomization countermeasures integrated and compare them with each other. For the attacks we use differential frequency analysis (DFA) and preprocessing techniques such as filtering and trace integration. Today's commercially available low-cost RFID tags do not have cryptographic algorithms with randomization countermeasures implemented. In order to perform and analyze the SCA attack techniques, we have used semi-passive RFID tag prototypes for the HF and the UHF frequency range as target of evaluation. In these prototypes it is possible to implement the AES algorithm with randomization countermeasures in software. The prototypes give also full control over the configuration of the randomization countermeasures, which is advantageous for the evaluation. Our results show that DFA is a powerful technique, especially when analyzing the electromagnetic emissions of RFID devices. The work presented in this chapter has been published at the WISA workshop 2009 [152], and has been jointly conducted with Michael Hutter and Martin Feldhofer.

The following topics are covered by the remainder of this chapter. In Section 7.1 we describe the HF and UHF tag prototypes that have been used for evaluating the effectiveness of the randomization countermeasure. Section 7.2 deals with noise in SCA measurements and discusses potential techniques that lower the impact of noise and that ease the attacking of hiding countermeasures. Details about the randomized AES implementation are provided in Section 7.3. A description of the deployed measurement setup is given in Section 7.4, and results are presented in Section 7.5. We close the chapter with a short summary in Section 7.6.

7.1 Description of the RFID Tag Prototypes

In order to evaluate the effectiveness of the randomization countermeasure, we have utilized two different RFID tag prototypes. Utilizing prototypes provides many advantages. They can be used to demonstrate new applications and protocols, making an invention more informative and imaginable. Prototypes are also suitable for identifying weaknesses more easily by modifying and testing the device in real terms. In cryptographic systems, prototypes allow the analysis of side channels by measuring, for example, the electromagnetic emanation. This chapter focuses on such analyses by using prototypes that implement security mechanisms.

We have used two RFID tag prototypes. One prototype operates in the HF frequency range at 13.56 MHz and one prototype works in the UHF frequency range at 868 MHz. Both devices have been assembled using discrete components. In Figure 7.1, a picture of the two prototypes is shown. They principally consist of an antenna, an analog front-end, a microcontroller, a clock oscillator, a serial interface, a JTAG interface, and a power-supply connector. Both devices differ in their antenna design, the analog front-end, the clock source, and the software that runs on the microcontroller (*i.e.* protocol implementation). The remaining components are the same. As a microcontroller, the ATmega128 [15]

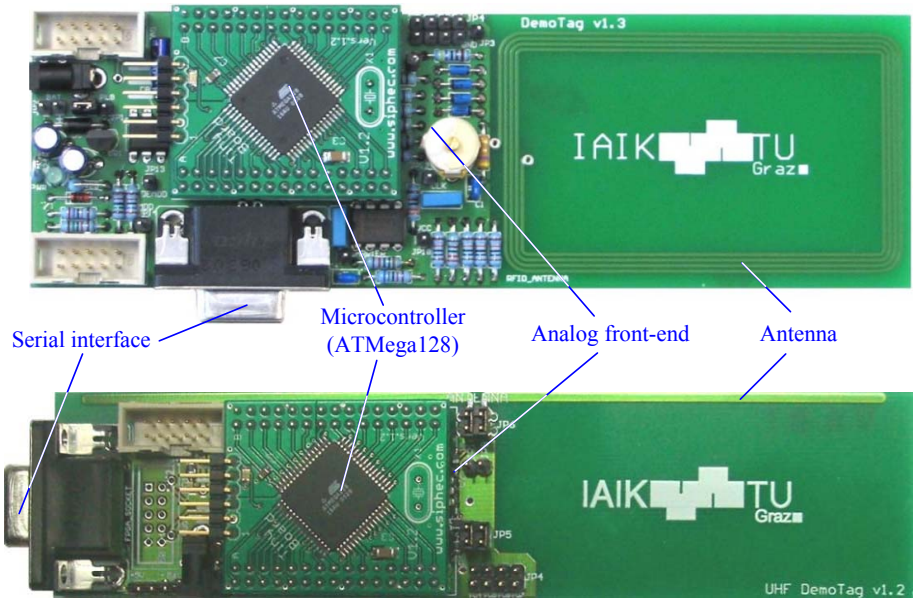


Figure 7.1: Picture of the HF (top) and the UHF (bottom) tag prototype.

has been used, which manages reader requests and tag responses by following the specification of the used RF communication protocol. The microcontroller is able to communicate with a PC over a serial interface. It furthermore supports In-System Programming (ISP) and has a JTAG interface for debug control and system programming. Both devices operate semi passively where the microcontroller is powered by an external power source, typically a battery, while the RF communication is done passively without any signal amplification. In the following, implementation details of the HF and the UHF tag prototype are given.

7.1.1 HF Tag Prototype

The HF tag prototype uses a self-made antenna according to ISO 7810. It consists of a coil with four windings that allows the communication with a reader over the air interface. The antenna is tuned to resonate at a carrier frequency of 13.56 MHz, which is realized by a matching RLC circuit. This circuit narrows the frequency range and can also be considered as a band-pass filter that passes the carrier frequency but attenuates unwanted and spurious frequencies. The matched signals are then preprocessed by an analog front-end that is used to transform the analog signals into the digital world. First, the signals are rectified using a bridge rectifier. Small-signal Shottky diodes have been assembled that provide low voltage drops and low leakage currents. Second, the voltage is regulated by a Zener diode. At the third stage, a comparator is used to identify reader modulations. The output of the comparator is then connected to the

microcontroller that rises an interrupt and starts the receiving process. The microcontroller is clocked by a 13.56 MHz quartz crystal that has been assembled on board. For sending data from the tag to the reader, a load modulation circuit is available that consists of a shunt and a transistor. The microcontroller triggers the transistor that switches the shunt and thus modulates the tag response.

The tag prototype can communicate using several protocol standards. It implements ISO 15693, ISO 14443 (type A and B), ISO 14443-4 and ISO 18092. The software is written in C while parts have been implemented in assembly language due to timing constraints. Moreover, it implements a user-command interface that allows easy administration over the serial interface. For our experiments, we have used the ISO 14443-A protocol standard [90] and have included some proprietary commands that implement a simple challenge-response protocol. First, the reader sends 16 bytes of plaintext to the prototype. The prototype encrypts the plaintext using the AES. Second, the reader retrieves the ciphertext and verifies the encrypted result. Furthermore, we have implemented a command to set different randomization parameters for the AES encryption. These parameters are used to randomize the encryption process that is commonly used as a countermeasure for side-channel analysis.

7.1.2 UHF Tag Prototype

The second tag prototype operates in the UHF frequency range. In contrast to the HF tag prototype it uses a half-wave dipole antenna consisting of two wires directly integrated to the layout of the printed circuit board (PCB). The antenna, whose length is about 150 mm, is optimized for a frequency of 868 MHz and it is connected to the analog front-end. Like for the HF tag prototype, an adjustable capacitor is placed in parallel to its antenna. This capacitor is used for matching the antenna to the input impedance of the analog front-end. Signals that are received by the antenna are first rectified by a charge-pump rectifier. This rectifier performs demodulation and voltage multiplication all at once. Special detector diodes, which have a low voltage drop and are constructed to operate up to some GHz, are used in the rectifier circuit. Subsequently, signals are filtered and passed to a comparator before feeding them to the microcontroller. For tag-to-reader communication, a backscatter-modulation circuit is provided within the analog front-end. This circuit, which works similar to the one used by the HF tag prototype, uses a transistor to switch an impedance (shunt and capacitor) in parallel to the tag antenna. A 16 MHz quartz crystal is assembled on board in order to generate the system clock for the microcontroller.

The UHF tag prototype supports the ISO 18000-6C standard (EPC Generation 2 standard [50]) which is the most widespread protocol in the UHF frequency range. Implementation of the protocol is done in software on the microcontroller. The software for the UHF tag prototype is also mainly written in C while time-critical routines are directly realized via assembly language. Also the same challenge-response protocol has been implemented that allows encryption of received data, as well as a dedicated command to adjust the parameters for the AES randomization.

7.2 Noise in Side-Channel Analysis Measurements

As discussed in Section 4.1.2, hiding can be used as a countermeasure to make side-channel analysis attacks less effective. Hiding can be done in the amplitude dimension as well as in the time dimension. Hiding aims to lower the signal-to-noise ratio (SNR) of the measurements by either increasing the noise or by lowering the data-depended part of the signal. However, the SNR of the measurements is not only effected by countermeasures such as hiding, but also by inherent noise sources and inaccurate trigger mechanisms. In the remainder of this section, we describe how noise can occur in RFID measurements, followed by a discussion of potential techniques to lower the impact of noise in measurements and to ease the attacking of hiding countermeasures.

7.2.1 Noise in RFID Measurements

Especially when evaluating the side-channel analysis resistance of RFID systems, noise in measurements is a concern. Noise can occur in both the amplitude dimension and in the time dimension. Noise makes side-channel analysis of RFID systems typically more difficult to conduct than in case of contact-based devices.

Noise in the Amplitude Dimension

Measuring the side channels that leak from RFID devices is a challenging task. RFID readers emit a very strong field in order to allow a certain reading range. This field is necessary to power the tags, to allow a communication, and in most cases also to provide a clock signal to the tags. However, the field interferes and perturbs the measurement of the weak side-channel emissions. In addition, if the reader field and the clock signal of the tag differ in their frequency, a superposition of signals can be perceived. This results in periodic rises and falls in the amplitude of the measured EM traces. Measurements on HF RFID tag prototypes, whose clock frequency differ from 13.56 MHz, are a typical example where the reader field interferes the measurement of interesting side-channel emissions. Measurements on UHF tags that often include their own oscillators are another example. The internal clock allows the communication with multiple reader frequencies that are used in different countries (868 MHz in Europe, 915 MHz in USA, or 950 MHz in Japan). In all cases, the variations in the amplitude dimension essentially lower the SNR and thus make the measurement of side channels harder to perform.

Noise in the Time Dimension

Additional noise can also emerge in the time dimension, where traces are misaligned due to the absence of adequate trigger events. Especially in RFID environments, triggering is often performed on the communication instead of the

measured emanation. For example, the end of the last reader command before executing the targeted algorithm can be used to trigger the measurement. This trigger signal does not always appear at the same position in time which leads to misaligned traces and thus to additional noise.

7.2.2 Techniques to Lower the Impact of Noise and to Ease the Attacking of Hiding Countermeasures

There exist various techniques based on trace preprocessing that help to increase the performance of SCA attacks in presence of measurement noise and hiding countermeasures. The most obvious and commonly used preprocessing technique is filtering. By applying different filters, it is possible to reduce noise that originates from narrow-band interferers such as RFID readers. Filtering of these perturbing signals helps to evade noise in the amplitude dimension. Though this requires knowledge of the appropriate filter parameters to preserve data-dependent information in the traces.

Misaligned traces that result from noise in the time dimension (*e.g.* inaccurate trigger events) and/or from hiding countermeasures (*e.g.* shuffling) can be preprocessed with pattern-matching techniques that try to realign the traces. The difficulty of this technique is to find an appropriate pattern in the traces for the realigning process. Especially in case of hiding countermeasures where the misalignment of the traces is typically much larger than in case of inaccurate trigger events, finding such patterns might be difficult.

An alternative approach to deal with misaligned traces is the integration of power or EM traces. Specific points in time are summed up before performing the attack. In practice, only points are chosen that exhibit a high side-channel leakage. These points form a kind of comb or window that can be swept through the trace in order to obtain the highest correlation. This technique is often referred to as *windowing* [117]. However, it is evident that this technique implies the knowledge of certain points in time where the leakage of information is high. If no knowledge of this leakage is available, it shows that the performance of this attack is rather low due to the integration of unimportant points.

Another related technique for handling misaligned traces uses the fast Fourier transform (FFT) to bring traces into the frequency domain. Instead of performing differential power analysis or differential electromagnetic analysis in the time domain, the analysis is performed in the frequency domain. Since the FFT is time-shift invariant, the time delays introduced by the side-channel analysis countermeasures or by inaccurate trigger events are removed in the frequency domain. Further advantage of this technique is that traces that are interfered by the reader field are of no concern. Such an analysis is also referred as Differential Frequency Analysis (DFA) which has been first mentioned by Gebotys *et al.* [63, 64] in 2005. There, the authors successfully applied DFA to attack cryptographic algorithms running on a personal digital assistant (PDA) device. Another approach that uses the frequency domain for handling misaligned traces has been presented by Homma *et al.* [73] in 2006. They have been able to dimin-

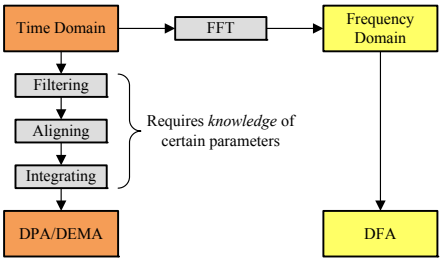


Figure 7.2: Overview of the preprocessing steps necessary for DEMA and DPA as well as DFA attacks.

Figure 7.3: Principle of shuffling used in the randomized AES implementation.

ish the displacement between traces by using a so-called phase-only correlation after transformation to the frequency domain.

Figure 7.2 illustrates the necessary preprocessing steps for conducting DEMA and DPA attacks as well as DFA attacks in case of traces that are misaligned and covered by additional noise. In this chapter, all three discussed types of preprocessing techniques are analyzed in terms of their efficiency. These preprocessing techniques are applied on EM measurements that have increased noise in both amplitude and time dimension. The increased noise is caused by an interfering RFID reader and by a randomized AES implementation that is described in the following.

7.3 Description of the Randomized AES Implementation

In our experiments, an AES implementation with 128-bit key length (AES-128) similar to the one presented in [72] has been used that offers hiding in the time dimension. First, the implementation allows to choose additional rounds that are randomly executed either at the beginning or the end of the actual algorithm. Second, it allows the shuffling of bytes $b0$ to $b15$ within the AES state. There, the sequence of the columns and the sequence of the rows can be randomized as shown in Figure 7.3. In order to set specific randomization parameters during our experiments, we have implemented a custom command that can be sent over the air interface. These parameters define the number of dummy rounds and the number of shuffling operations. In particular, it is possible to define the sequence of the columns as well as the sequence of the rows within the AES state. If no dummy rounds are inserted and all bytes of the state are shuffled, 16 different positions can be taken over time for one state operation. Regarding side-channel analysis, the correlation coefficient through randomization is then reduced linearly by a factor of 16. The number of necessary traces to succeed

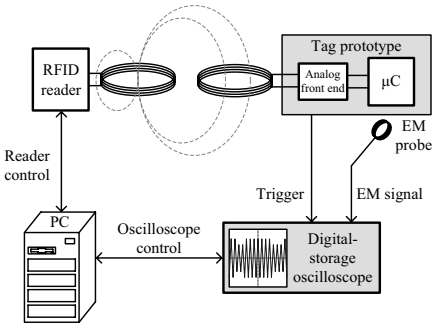


Figure 7.4: Schematic view of the general measurement setup used to gather the EM emissions of the tag prototypes.

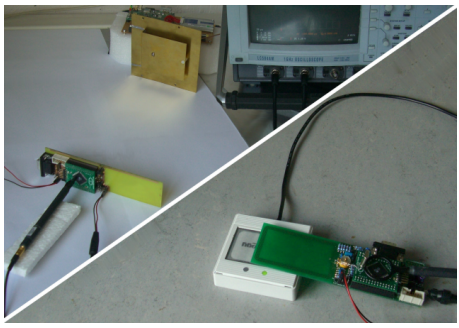


Figure 7.5: Picture of the measurement setup using UHF (upper left) and HF (lower right) RFID tag prototypes.

an attack increases roughly by a factor of $16^2 = 256$. However, the quadratic influence is only correct when no preprocessing method like windowing or DFA is applied [117]. When applying for example windowing techniques to attack an implementation with a shuffling countermeasure, the correlation coefficient is no longer reduced linearly but only with the square root (*i.e.* reduced by a factor of $\sqrt{16} = 4$ when shuffling the state over 16 positions). Consequently, the number of traces required for a successful attack only increases linearly and not quadratically (in our example the number of traces increases by a factor of 16 instead of 256). This clearly illustrates that the effort for a successful attack is significantly lowered when proper preprocessing methods are applied.

7.4 Measurement Setup

The measurement setup used for our experiments is shown in Figure 7.4. It comprises different devices such as a PC, a standard RFID reader, a digital-storage oscilloscope, the tag prototype, and a measurement probe. The RFID reader and the digital-storage oscilloscope are directly connected to the PC that controls the overall measurement process. MATLAB is running on the PC and is used to apply the preprocessing techniques and to conduct the side-channel analysis. The RFID reader communicates with the tag prototype via the air interface. For the HF tag prototype, the ISO 14443-A protocol has been used while the ISO 18000-6C protocol has been used for the UHF tag prototype. The HF tag prototype has been placed directly upon the reader antenna. The UHF tag prototype has been placed 30 cm away from the UHF reader. Two channels of the digital-storage oscilloscope (*LeCroy LC584AM*) are used in our experiments. One channel is connected to the trigger pin of the tag prototype, the other channel is connected to the measurement probe. Signals have been sampled with 2 GS/s. Both tag prototypes have been programmed to release a trigger event whenever a new AES encryption is started. This trigger event causes the

oscilloscope to record the EM emissions of the tag prototype using magnetic near-field probes. We have used two probes from *EMV Langer* which is the RF R 400 for the HF measurements and the RF B 3-2 for the UHF measurements. Figure 7.5 shows a picture of the measurement setup for the HF and one for the UHF tag prototype.

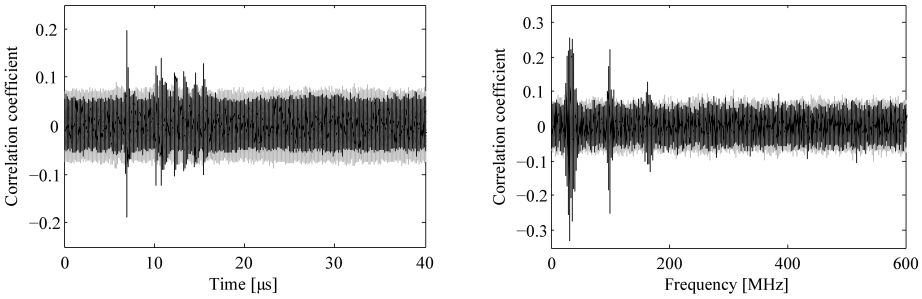
7.5 Results

In this section, the results of the performed side-channel attacks on our RFID tag prototypes are presented. Attacks have been performed on the electromagnetic emissions of the HF and the UHF tag prototype. The target of all attacks has been the first key byte of AES. As a power model, the Hamming weight has been used.

First, we have analyzed traces that contain noise in the amplitude dimension. For this, we have measured EM emissions of our prototypes that are interfered by unsynchronized reader signals. Note that no randomization of the AES state is enabled in this experiment. In order to perform attacks on such kind of polluted traces, we investigated two different preprocessing approaches. The first approach applies filtering techniques to suppress the interfering noise of the reader. The second approach applies an FFT before performing the DFA attack. We compare both techniques in their practical efficiency and performance. Second, we show results of attacks that have been performed on traces that contain both noise in the amplitude dimension and misalignment. For this experiment, we have enabled the randomization of the AES implementation which is commonly used in practice to counteract against side-channel attacks. For this scenario, we have applied trace-integration techniques by windowing. We also compare the results with the results obtained by DFA.

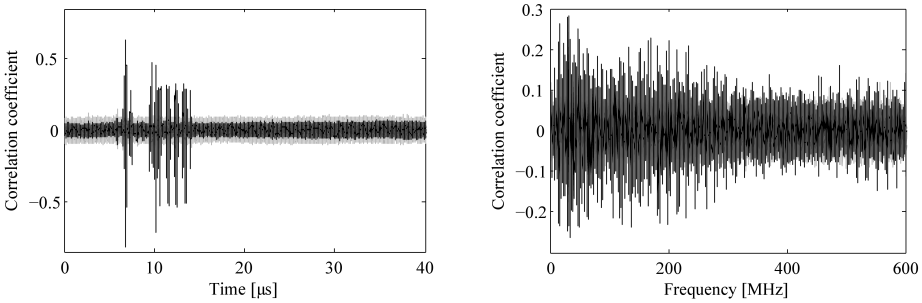
7.5.1 Results with Noise in the Amplitude Dimension

At first, we focus on typical measurements in RFID environments. The additional noise in the EM traces is caused by readers that interfere the EM measurement of RFID devices. In our experiment, we consider the scenario where the clock signal of our prototype and the reader carrier are desynchronized. This is already the case for our UHF tag prototype which operates at 16 MHz and which communicates with an 868 MHz reader. For the HF prototype, we have used a 13.56 MHz quartz crystal that is assembled on board. This quartz crystal is also unsynchronized with the communicating 13.56 MHz reader. Both devices have been placed inside the reader field, which interfered the measurement due to additional noise. After the acquisition of 2 000 traces, we have performed filtering techniques to circumvent the interferer and to decrease the noise at this juncture. For the HF prototype, a bandstop filter has been designed using MATLAB that filters the 13.56 MHz carrier. For the UHF prototype, a low-pass filter has been used that passes all frequencies below 200 MHz. We have performed a filtered DEMA attack and a DFA attack using FFT.



(a) (b)

Figure 7.6: Result of the filtered DEMA attack (a) and DFA attack (b) on the HF tag prototype when introducing noise in the amplitude dimension.



(a) (b)

Figure 7.7: Result of the filtered DEMA attack (a) and DFA attack (b) on the UHF tag prototype when introducing noise in the amplitude dimension.

In Figure 7.6(a), the result of the filtered DEMA attack for the HF tag prototype is shown. The correct key hypothesis is plotted in black while all other key hypotheses are plotted in gray. The correct key hypothesis leads to a correlation coefficient of 0.20. Figure 7.6(b) shows the result of the DFA attack. Three peaks in the electromagnetic spectrum are clearly discernable, which represent high data-dependent frequency emissions. The highest absolute correlation coefficient has been 0.33 and occurred at a frequency of around 33 MHz.

In Figure 7.7(a), the result of the filtered DEMA attack is presented that has been performed on the UHF tag prototype. A maximum absolute correlation coefficient of 0.63 has been obtained for the correct key hypothesis. Figure 7.7(b) shows the result of the DFA attack. As opposed to the results of the HF tag prototype, many peaks occurred up to a frequency of about 600 MHz. The highest correlation that has been obtained is 0.28.

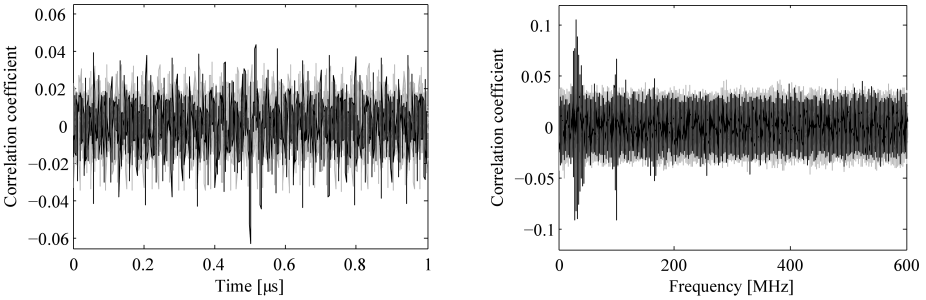
For the UHF tag prototype, the results show a higher correlation coefficient compared to the results of the HF prototype. This is explained by the fact that

our UHF measurement setup provides a higher SNR. On the one hand, a different EM probe has been used for the measurement that allows the probe to be drawn nearer to the surface of the chip (the RF B 3-2 probe from *EMV Langer* has been used for the UHF measurements and the RF R 400 for the HF measurements). On the other hand, our experiments have shown that the UHF reader produces lower noise compared to the HF reader. However, when the result of the filtering technique and the result of the DFA are compared to each other, it shows that the DFA attack leads to a higher correlation in noisier environments while it is less effective in measurements where a low noise source is present.

7.5.2 Results with Noise in the Amplitude Dimension and Activated Shuffling

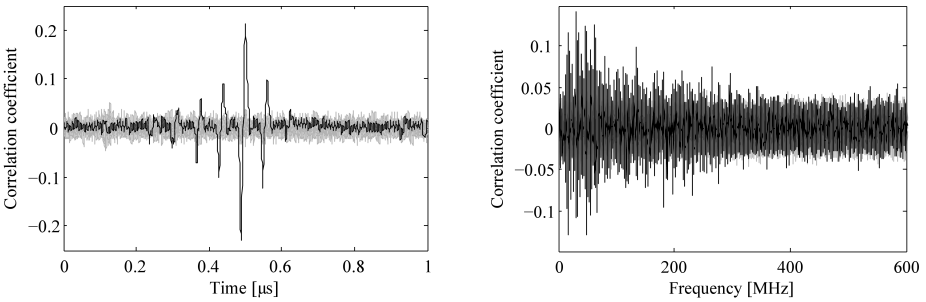
Next, we consider the scenario where a side-channel countermeasure is enabled on the tag side. In addition to the noise of the reader, we have activated the randomization countermeasure of the AES. As stated in Section 7.3, we are able to shuffle all bytes within the AES state. This leads to 16 different positions in time where a byte may be processed during one round. Nonetheless, the results of our experiments have shown that for the HF tag no significant correlation has been obtained for the case where we have preprocessed the traces using the trace integration (windowing) technique. By performing the attack in the frequency domain using DFA, we successfully revealed the correct key byte. Hence, we decided to reduce the number of shuffling bytes to 8 for the HF-measurement scenario in order to succeed the attack in both cases. The attacks for the UHF-measurement scenario, in contrast, have been successful when randomizing all 16 bytes of the AES state. We performed software filtering as described in the section above to reduce the noise of the RFID reader for the DEMA attacks in the time domain. For each experiment, 10 000 traces have been captured.

The attack using windowing as a preprocessing technique has been performed as follows. We summed up 100 points in time which showed the highest correlation in a previously performed standard DEMA attack. This defines an integration window that involves points with high data-dependent information. For a better visualization of the window matching, we have further implemented an automatic sweep that slides the window from the beginning of the trace to its end. At each position in time, all points of the window are summed up and a DEMA attack is performed afterwards. This results in a correlation trace where a peak occurs in time when the window fits best the specified data-dependent locations. Using such a windowing approach requires that at least one successful attack (without using windowing) has been performed in advance. This can be interpreted as a characterization phase that allows to significantly lower the effort for attacking consecutive devices. In Figure 7.8(a), the result of the attack on the HF tag prototype is shown where we have zoomed only into the interesting region in time. A peak is observable which has a maximum absolute correlation coefficient of 0.06. In Figure 7.8(b), the result of the performed DFA attack is given. Note that neither filtering nor other trace-alignment techniques



(a) (b)

Figure 7.8: Result of the windowing attack (a) and DFA attack (b) on the HF tag prototype when introducting noise in the amplitude dimension and shuffling 8 bytes of the AES state.



(a) (b)

Figure 7.9: Result of the windowing attack (a) and DFA attack (b) on the UHF tag prototype when intrducing noise in the amplitude dimension and shuffling 16 bytes of the AES state.

have been applied before. Two peaks are discernable that arise at about 30 MHz and 100 MHz. These data-dependent frequencies are the same as those we have already obtained in the previous experiment (see Figure 7.6(b)). The highest correlation coefficient is 0.10.

After that, we have focused on our UHF tag prototype. We have applied the same integration technique as used for the HF tag prototype. In contrast to the attack on the HF tag prototype where 8 bytes have been randomized, now 16 bytes have been shuffled within the AES state. Figure 7.9(a) shows the trace-integration result of the UHF tag prototype. A maximum absolute correlation coefficient of 0.23 has been obtained. In Figure 7.9(b), the result of the performed DFA attack is shown again without using any filtering or trace-alignment techniques. There, a maximum correlation coefficient of 0.14 is obtained.

By taking a closer look at our results, it becomes clear that DFA poses

a powerful and easy to use preprocessing technique that is able to reveal the secret key of our RFID tag prototypes. DFA provides not only high correlation in noisy environments but can also be successfully applied against randomization countermeasures without having any knowledge of either interfering frequencies nor data-dependent locations.

7.6 Summary

In this chapter, we have presented results of DEMA and DFA attacks on HF and UHF RFID tag prototypes. We have addressed the issue of traces that contain additional noise and that are misaligned. These traces are interfered by the reader field, which results in increased noise in the amplitude dimension. In addition to that, we have investigated a randomized AES implementation in software that hides the side-channels leakage in the time dimension. We have performed DEMA attacks in combination with preprocessing techniques such as filtering and trace integration (windowing), as well as DFA attacks. A summary of the results that have been achieved is presented in Table 7.1. Our experiments prove that DFA is a powerful attack technique that allows a fast and time-invariant analysis even in environments where traces are covered by noise and misaligned due to randomization. Filtering techniques, in contrast, need the knowledge of the noise-source frequency and might also suppress interesting leakages. Applying integration techniques is a time-consuming task that requires the knowledge of the data-dependent locations to design an appropriate integration window. Moreover, if the degree of randomization is increased, the number of windowing points has to be increased as well. We conclude that DFA offers many advantages especially when neither knowledge of the device nor possibilities of noise reduction are given. All side-channel attacks performed on the RFID tag prototypes with the randomized AES implemented in software have been successful by applying DFA. This also clarifies that RFID devices that are using randomization as a countermeasure suffer from this kind of attack. The effort for attacking commercially available RFID tags is assumed to be higher, since they will have their cryptographic algorithm and the countermeasure realized in dedicated hardware. Nevertheless, combining randomization with other countermeasure approaches as proposed in [117] might be a good solution to provide a higher degree of security.

Table 7.1: Summary of the side-channel analysis results that have been obtained by performing DEMA and DFA attacks on the tag prototypes.

Attack technique	**Scenario**	**HF tag prototype**	**UHF tag prototype**
-	-	[Correlation]	[Correlation]
DEMA[a]	Noise in amplitude	0.20	0.63
DFA		0.33	0.28
DEMA[b]	Noise in amplitude	0.06	0.23
DFA	and shuffling	0.10	0.14

[a]Filtering has been applied before applying the DEMA attack.
[b]Filtering and trace integration (windowing) has been applied before applying the DEMA attack.

8

Evaluating the Detached Power Supply as Side-Channel Analysis Countermeasure for Passive UHF RFID Tags

The results presented in Chapter 5 clearly point out that side-channel analysis is a serious concern for passive RFID tags that operate in the ultra-high frequency (UHF) range. Due to the so-called *parasitic backscatter* that has been discovered by Oren and Shamir [141] in 2007, differential electromagnetic analysis (DEMA) attacks on passive UHF RFID tags can be conducted from a distance of one meter and more [147]. In order to make UHF tags less vulnerable to parasitic-backscatter attacks, suitable countermeasures have to be integrated. Unfortunately, integrating countermeasures usually also increases power consumption, chip size, and design complexity of a device. Since passive low-cost tags have only limited power available (to achieve a reasonable read range) and must not consume too much chip area to stay competitive in price, proper selection of the countermeasures is very important.

A countermeasure recently proposed by Shamir [167, 168] for protecting UHF tags from parasitic-backscatter attacks is the detached power supply. It is a hardware countermeasure based on hiding at the architectural level. The decoupling of the power consumption is accomplished by using two capacitors. At any time, one capacitor powers the digital circuit of the tag and the other capacitor is charged by the tag's analog front-end. By periodically switching the capacitors, energy is transferred from the analog front-end to the digital circuit without a direct physical connection. The simplicity and the fact that its application requires no extensive modification of existing chip designs makes it interesting for RFID tags. When we started our research, no information has been available

about the efficiency of the detached power-supply countermeasure. Although there has existed a practical implementation of a similar approach by Corsonello *et al.* [39] using a three-phase charge pump, results illustrating how well this countermeasure prevents side-channel analysis have been missing.

In this chapter, we evaluate the efficiency of the detached power supply and discuss its suitability for protecting passive UHF tags from parasitic-backscatter attacks. The work has been published at the CT-RSA conference 2009 [148] and has been the first article that presents practical results of the detached power-supply countermeasure with respect to side-channel analysis. We have implemented the detached power supply with discrete components and applied it to a smart card with an unprotected software version of the Advanced Encryption Standard (AES). Using a smart card has been necessary since commercially available passive UHF tags do not yet contain standardized cryptographic algorithms. By performing DPA attacks, we have analyzed a basic version of the detached power supply and an enhanced version which uses an additional discharge phase. The results have shown that even the enhanced version of the detached power supply is still vulnerable to power analysis because of the non-ideal properties of the analog switches. Moreover, it has turned out that there is a strong side-channel leakage at the I/O pin of the smart card, regardless whether the detached power has been used or not. In order to address this problem, we suggest a simple countermeasure that prevents the side-channel leakage at output pins. Finally, an estimation is provided about the required capacitor size and the power-consumption overhead caused by integrating the detached power supply into a passive UHF tag.

This chapter is organized as follows. Section 8.1 describes the principle of the detached power supply. Section 8.2 illustrates the practical implementation of the countermeasure and the measurement setup. The side-channel analysis results are presented in Section 8.3. A suggestion for a simple countermeasure to prevent side-channel leakage at output pins is shown in Section 8.4. In Section 8.5, the costs of integrating the detached power supply into passive UHF tags is discussed. The chapter closes with a summary in Section 8.6.

8.1 Description of the Detached Power Supply

Initially, the detached power supply has been intended to protect smart cards from power analysis [167]. However, the countermeasure is much more suitable for preventing parasitic-backscatter attacks on UHF tags [168]. First, the detached power supply does not protect from side-channel analysis that uses the direct emissions of the device. Second, manipulating the capacitors (*e.g.* interconnecting the pins of the capacitors) makes the countermeasure completely useless. In case of the parasitic-backscatter attack, where a passive attacker remotely measures the power reflected from a tag, both arguments are of no importance. Moreover, the power consumption of passive tags is much lower compared to smart cards. The lower power consumption allows capacitors to be smaller and they can directly be integrated into the chip of the tag.

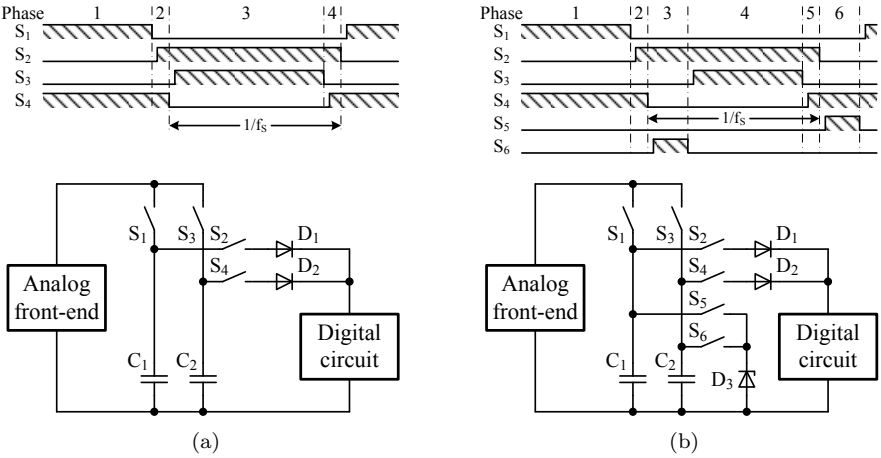


Figure 8.1: Sequence diagrams comprising the states of the switches during the particular phases and a schematic overview of the circuits used for the basic version (a) and the enhanced version (b) of the detached power supply.

In the following, the principle of the detached power supply is explained in more detail. After the description of the basic version, an enhanced version with an additional discharge phase is presented.

8.1.1 Basic Version of the Detached Power Supply

The basic version of the detached power-supply [168] comprises: two capacitors, four switches, and two diodes. Figure 8.1(a) presents a schematic diagram and a time lapse of the switches' states. Shaded areas in the sequence diagram indicate the intervals in which a certain switch is closed. A complete cycle consists of four phases. In the first phase the switches S_1 and S_4 are closed and S_2 and S_3 are opened. This causes C_1 to be charged by the analog front-end, while C_2, which has been charged in a previous phase, powers the tag's digital circuit. During the second phase, which is rather short in time, S_1 is opened and S_2 is closed. Now, the digital circuit is powered by both capacitors. The diodes D_1 and D_2 prevent charge equalization between the two capacitors. In the next phase, S_4 is opened and S_3 is closed. The digital circuit is powered by C_1 and C_2 is charged by the analog front-end, the opposite way around than in the first phase. In the fourth and last phase, which is again a short phase where both capacitors power the digital circuit, S_3 is opened and S_4 is closed. After the fourth phase, the cycle is completed by opening S_2 and closing S_1 and starts from the beginning. The toggling between C_1 and C_2 is done at the switching frequency f_S.

Switching between particular phases can be assigned to a fixed number of clock cycles or triggered by the voltage of the discharging capacitor when reaching a certain threshold. Regardless of the switching strategy, an attacker can

still get information about the total amount of charge consumed by the digital circuit during a discharge phase. This information can be minimized by selecting larger capacitors allowing to make the discharge phase longer.

8.1.2 Enhanced Version of the Detached Power Supply

In order to remove the remaining information leakage present in the basic version of the detached power supply, it is suggested in [167] to discharge each capacitor to a fixed voltage level before reconnecting it to the analog front-end. This enhancement requires two additional switches and a voltage-limiting element for properly discharging the capacitors. An example for a simple voltage-limiting element is a Zener diode. Figure 8.1(b) shows the schematic overview of the circuit and a sequence diagram comprising the states of the switches during the particular phases. In contrast to the basic version, the enhanced version needs six instead of four phases for a complete cycle. During each of the two additional phases, one capacitor is switched in parallel to the Zener diode D_3 to get further discharged to a predefined voltage level. All other phases are the same as in the basic version. In the third phase C_2 is connected to the Zener diode via S_6, in the sixth phase C_1 via S_5. As a result, the charge stored in the concerning capacitor is no longer related to the charge consumed by the digital circuit while it has been supplied by the capacitor. Theoretically, this makes power-analysis attacks completely useless.

8.2 Implementation of the Detached Power Supply and the Measurement Setup

Evaluating the effectiveness of the detached power supply requires its implementation and furthermore application to a physical device. Since passive UHF tags that are commercially available do not yet have integrated standardized cryptographic algorithms, we have decided to use an unprotected smart card instead. This allows to draw conclusions for passive UHF tags, because the countermeasure operates independently of the circuit it protects. Both the simple and the enhanced version of the detached power supply have been implemented as an analog circuit using discrete components. The switches of each circuit are handled by a microcontroller. For better flexibility, the microcontroller is connected to a PC via a serial interface to adjust certain parameters like the switching frequency or activation/deactivation of the detached power supply. Figure 8.2 illustrates how the detached power supply and the smart card are combined to protect against power analysis. The detached power supply is integrated into the supply line of the smart card. All other pins like clock, reset, and I/O are left untouched.

The utilized smart card runs at a clock frequency of 3.57 MHz and consumes about 5 mA at 5 V. A software version of the AES using a key length of 128 bits (AES-128) is implemented on the smart card. The AES is a block cipher

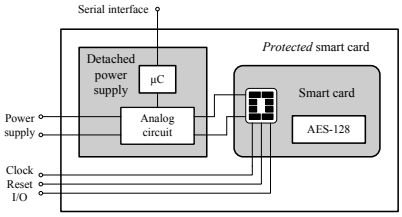


Figure 8.2: Overview of the smart card protected with the detached power-supply countermeasure.

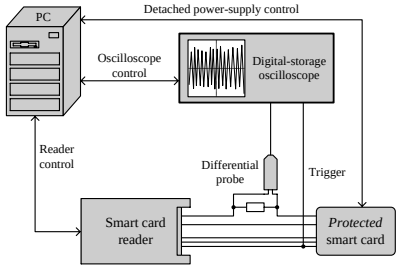


Figure 8.3: Measurement setup used to determine the power consumption of the protected smart card.

operating on 128-bit blocks of input data. Encrypting a single block of data takes the smart card less than 3 800 clock cycles.

Crucial components of the detached power supply are the analog switches which should ideally have: high switching speed, low cross talk, high off isolation, and low on resistance. After testing several switches, it has pointed out that universal serial bus (USB) high-speed multiplexers are a good solution since they have excellent electrical properties. The multiplexers can operate up to some hundreds of MHz, have an off isolation of about 100 dB at 100 kHz, and have a maximum on resistance of 10 Ω.

Standard ceramic types with a value of 0.1 µF have been selected for the capacitors. Taking into account the smart card's average power consumption of about 5 mA and allowing the capacitors to be discharged from 5 to 3.5 V, results in a minimum switching frequency of about 36 kHz. This means that a single 0.1 µF capacitor can power the smart card over a time of approximately 100 clock cycles.

Further components of the circuit are low voltage-drop Shottky diodes that prevent charge equalization between the two capacitors while they are temporarily switched in parallel to the smart card. A Zener diode has been chosen as voltage-limiting element for the enhanced version of the detached power supply. The breakdown voltage of the Zener diode depends on the deployed switching frequency of the detached power supply and needs to be below the minimum voltage reached by the capacitor after being discharged by powering the smart card.

Besides implementing the detached power supply and applying it to a suitable cryptographic device, building a measurement setup is necessary for the evaluation process. Figure 8.3 shows a measurement setup that allows automated acquisition of the power traces. Main components of the measurement setup are: the protected smart card, a smart-card reader, a PC, a differential probe, and a digital-storage oscilloscope. The smart-card reader, the digital-storage oscilloscope, and the microcontroller that is responsible for managing the switches of the detached power supply are controlled by a MATLAB script running on the

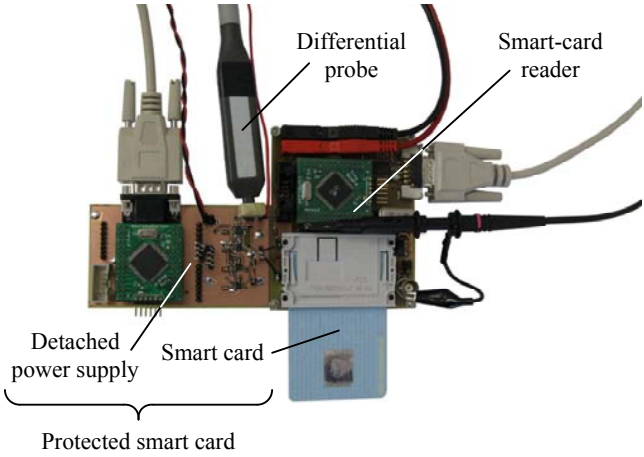


Figure 8.4: Photo of the actual measurement setup containing the protected smart card, a smart-card reader, and a differential probe (the differential probe is connected between smart-card reader and protected smart card).

PC. A photo of the actual measurement setup containing the protected smart card, the smart-card reader, and the differential probe is presented in Figure 8.4.

The measurement cycle for retrieving a single power trace always follows the same scheme. After receiving an appropriate command from the smart-card reader, the protected smart card starts encrypting the incoming data block. When the encryption begins, the protected smart card releases a trigger event that causes the digital-storage oscilloscope to record a power trace. Determining the power consumption is achieved by measuring the voltage drop across a $1\,\Omega$ resistor in the supply line of the protected smart card with a differential probe. The measurement cycle is finished by transferring the power trace to the PC, where further analysis work is conducted.

8.3 Results of the Side-Channel Analysis

This section presents the results obtained by analyzing the power consumption of the smart card equipped with the detached power supply described in Section 8.2. DPA attacks have been carried out using the Pearson correlation coefficient r to detect linear dependencies between the measured power consumption and the formed hypotheses (see (3.1) in Chapter 3). As power model, the Hamming-weight model has been selected. In that way, attacking the first key byte of the AES implemented on the smart card without using any countermeasures, has led to a correlation coefficient of 0.55 for the correct hypothesis. Based on a given r, the rule of thumb stated in [117] allows to determine the number of power traces n that is needed for a successful attack with high probability ($> 99.99\%$):

$$n = 3 + 8\frac{3.719^2}{ln^2(\frac{1+r}{1-r})} \tag{8.1}$$

Substituting 0.55 for r in (8.1) computes to about 75. Hence, 75 power traces are required on average for a successful DPA attack on the smart card. This value is later used as a reference to better quantify the impact on the number of needed power traces when the countermeasure is activated. Consecutively, the results of the DPA attacks applied to the smart card with activated detached power supply are presented. Both basic and enhanced version of the countermeasure are examined and compared with each other.

8.3.1 Results of the Basic Version of the Detached Power Supply

After analyzing the side-channel leakage of the smart card itself, measurements with the basic version of the detached power supply as countermeasure have been conducted. Various switching frequencies starting from 640 kHz to 36 kHz have been examined. It has shown that analyzing power traces resulting from measurements with activated detached power supply is costlier since they require additional preprocessing steps. An example of such a power trace is given in Figure 8.5. The traces are afflicted with a variable offset, making it necessary to align them vertically. Moreover, in our implementation the switching of the capacitors is controlled by an extra microcontroller and occurs independently from the operation of the smart card. This makes standard DPA attacks based on power traces highly inefficient. However, transforming the power traces from the time domain into the frequency domain as suggested by Gebotys [63], has solved this problem as well.

When applying the preprocessing techniques described above, power-analysis attacks have been successful if the basic version of the detached power-supply countermeasure has been activated. As expected in theory, decreasing the switching frequency has lowered the side-channel leakage. Using a switching frequency of 100 kHz, which equals to an integration over about 36 clock cycles of the smart card's power consumption, has led to a maximum r of 0.1 for the correct hypothesis. Selecting 36 kHz, which is the lowest possible switching frequency for our setup equaling to an integration over 100 clock cycles, has resulted in a maximum r of 0.04. According to (8.1), approximately 2 800 measurements are necessary on average for an attack when integrating over 36 clock cycles and more than 17 200 measurements if integrating over 100 clock cycles.

A side-channel leakage that is not decreased by lowering the switching frequency could be the indicator for inadequate analog switches. We have observed for example that analog switches with an insufficiently large off isolation at higher frequencies make the detached power-supply countermeasure useless. Data dependencies modulated on harmonics of the smart card's clock frequency pass the analog switches more or less unhampered. Hence, the choice of suitable switches

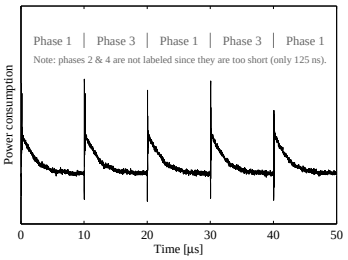


Figure 8.5: Power trace of the protected smart card with the basic version of the detached power supply using a switching frequency of 100 kHz.

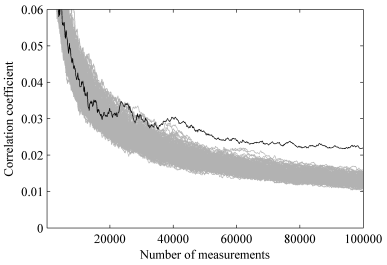


Figure 8.6: Correlation coefficient as a function of the number of measurements for the enhanced version of the detached power supply using a switching frequency of 100 kHz.

is essential. The switches that we have finally used have an off isolation of about 100 dB at 100 kHz.

8.3.2 Results of the Enhanced Version of the Detached Power Supply

Evaluating the efficiency of the enhanced version of the detached power supply is of much more interest. As a vulnerability to power analysis of the basic version is already stated in theory, it is not so for the enhanced version. The microcontroller that is responsible for activating the analog switches of the detached power supply has been configured such that an additional discharge phase of 10 clock cycles is introduced. During this phase, one of the two capacitors is further discharged to a fixed voltage level before reconnecting to the power supply to get rid of the remaining side-channel leakage.

Two switching frequencies have been examined, 100 kHz and 36 kHz. By applying the same preprocessing and analysis techniques as described in Section 8.3.1, a maximum correlation coefficient r of 0.022 has been obtained for a switching frequency of 100 kHz. According to (8.1), this corresponds to 57 100 measurements needed for a successful attack with high probability. Figure 8.6 shows the correlation coefficient as function of the number of measurements. The correct hypothesis is printed in black, incorrect hypotheses are printed in gray. Compared to the basic version of the detached power supply using the same switching frequency, the maximum correlation coefficient is reduced from 0.1 to 0.022 and the number of needed power traces is increased from 2 800 to 57 100. A certain vulnerability to side-channel analysis is still present. However, when decreasing the switching frequency from 100 kHz to 36 kHz, the maximum-achievable correlation coefficient is lowered as well. We have limited the number of measurements per experiment to 100 000 and therefore have not been able to

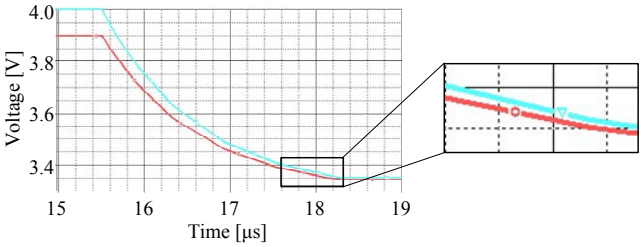


Figure 8.7: Screenshot of the two discharge curves that have been obtained with the computer simulation.

perform a successful attack on the smart card when using a switching frequency of 36 kHz.

The remaining side-channel leakage is explained with the non-ideal properties of the analog switches. In contrast to ideal switches whose resistance is assumed to be zero when activated, our analog switches have a maximum on resistance R_{ON} of about 10 Ω. This resistance acts in series to the capacitor during the discharge phase. Consequently, the discharge speed of the capacitor is limited by the time constant τ which is the product of R_{ON} and the capacitance of the capacitor. The time constant indicates the time required to discharge a capacitor to about 37 % of its initial charge. A shorter time constant is achieved by selecting more appropriate switches with lower R_{ON} or by using smaller capacitors.

We have verified the influence of R_{ON} by performing simulations with a computer program that is intended for analyzing the behavior of electronic circuits (a SPICE program). In the simulations, two identical capacitors, one charged to 4.0 V and the other to 3.9 V, are discharged over a 10 Ω resistor to a constant voltage of 3.3 V. The discharge time is selected to be equal to an interval of 10 clock cycles of our smart card. After discharging the capacitors, one capacitor has a remaining voltage of 3.352 V and the other of 3.344 V. Figure 8.7 shows a screenshot of the two discharge curves. Hence, the initial voltage difference of 100 mV has not completely disappeared, but has been reduced to 8 mV. This clarifies that even under ideal conditions as they can be found in the simulation, a non-zero R_{ON} has a noticeable effect.

Besides the influence of the R_{ON} of real switches, there is another aspect that needs to be considered when using the detached power supply to protect a cryptographic device against power analysis. As already mentioned in [167], data-dependent information can not only leak through the power line of a device but also through its I/O pins. We have measured the voltage variations at the I/O pin of our smart card while encrypting data to evaluate whether such attacks pose a threat or not. The deployed smart card operates conform to the ISO 7816 standard [86] and uses one pin for serially receiving and transmitting data in a half-duplex mode. During the processing of data, the smart card keeps its I/O pin in high-impedance mode (tri-state). We have detected rather strong data-

dependent leakage at the smart card's I/O pin, regardless whether the detached power supply has been activated or not. A maximum correlation coefficient r of about 0.15 has been obtained by only measuring the voltage variations at the I/O pin and by consecutively applying a DPA attack.

8.4 A Suggestion for Preventing Side-Channel Leakage at Output Pins

As the results of the previous section illustrate, decorrelating the power consumption of a device alone is not enough. Due to coupling effects on the chip, data-dependent information can also leak by the device's I/O pins. An example for such coupling effects is crosstalk, where the switching activity of one wire influences the signal of neighboring wires [101]. There exist various techniques for minimizing crosstalk like reducing the wire lengths, and increasing the distance between critical signal wires or introducing additional ground planes between them. These measures need to be applied during the chip-design phase and probably increase the design complexity and the resulting chip size. In addition, with shrinking CMOS technology the impact of the crosstalk effect is increased [197].

The digital circuit of UHF tags requires at least one output pin which controls a so-called backscatter circuit that is used to transmit data to the reader. This backscatter circuit is connected in parallel to the antenna and varies the power consumption of the tag which in turn influences the power reflected by the antenna. In contrast to the parasitic backscatter, this is an intended effect. Consequently, side-channel leakage coupled into the output pin can propagate through the backscatter circuit to the tag's antenna. As proposed in [167], the most obvious solution to avoid the side-channel leakage is to temporarily disconnect or to ground the output pin during the execution of cryptographic operations. However, as demonstrated in the work of Schmidt *et al.* [165] there is still a significant amount of side-channel information leaking through I/O pins even if they are disconnected (*i.e.* brought into a high-impedance state) or connected to ground.

A decoupling principle similar to the detached power-supply approach can be applied to address the side-channel leakage at the digital circuit's output pin. By using the output-decoupling principle shown in Figure 8.8, data present at the output pin of the digital circuit is transmitted to the backscatter circuit without direct physical connection at any time. Figure 8.8 also illustrates how to combine the detached power supply and the decoupling principle to protect passive UHF tags. The components required for the decoupling of the output are the switches S_1 and S_2, the capacitors C_1 and C_2, and a Schmitt trigger T_1. In a first phase, S_2 is opened and S_1 is closed. This causes C_1 to be charged to the voltage at the output pin of the digital circuit. The voltage across C_2 is buffered by the Schmitt trigger and converted to a distinct digital voltage level at its output. In a second phase, which is rather short in time, S_1 is

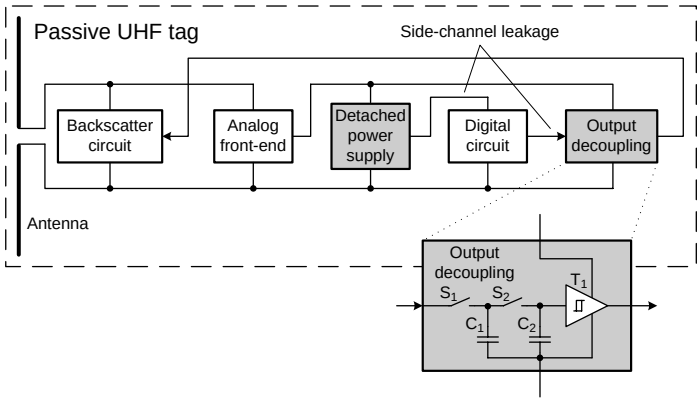


Figure 8.8: Schematic of the decoupling principle for the output pin and its combination with the detached power supply to protect passive UHF tags.

opened and S_2 is closed. Since the value of C_2 is much smaller than that of C_1, the resulting voltage across the parallel connection of the two capacitors is approximately equal to the initial voltage across C_1. After the second phase, the cycle is completed and continues from the beginning. The duration of a cycle defines the maximum-achievable data rate and needs to be selected properly to meet the requirements of the application.

In terms of additional hardware costs, the decoupling of the output pin is rather cheap. The size of the capacitors and the switches can be kept small because of the low currents, and the Schmitt trigger only requires a handful of additional transistors. Another advantage of this approach is its simplicity. No extensive redesign of the tag chip is required to address potential coupling effects at the output pin of the digital circuit.

8.5 Discussing the Costs of Integrating the Detached Power Supply into Passive UHF Tags

Integrating the detached power supply into passive UHF tags introduces additional costs in terms of chip size and power consumption that need to be considered. The low power consumption of passive UHF tags, which is typically in the range of some microamperes, allows to deploy small capacitors for the detached power supply. For a rough estimation of the required capacitor size, we have taken the experimental results from [113], describing the digital circuit of a passive UHF tag with an integrated AES module. As stated by the authors, their implementation on a 0.18 µm CMOS process has a power consumption of about 2.6 µA at a supply voltage of 1.8 V. The clock frequency of the AES module is around 420 kHz. When integrating over 100 clock cycles and allowing the capacitors to be discharged to a voltage of 1 V, each of the two capacitors for the detached power supply needs to have a value of at least

0.78 nF. Although capacitors in the size of 1 nF are not unusual for current UHF tags [51], integrating another two capacitors of that size directly into the chip of a tag could be too costly. On-chip capacitors have the advantage that it is more difficult for an attacker to manipulate them. However, in order to protect the tags from parasitic-backscatter attacks, which are conducted remotely and without modification of the tag, cheaper external capacitors can be deployed as well.

Another important aspect for passive UHF tags is the increased power consumption caused by the detached power supply. Real switches have an on resistance that is different from zero and charging a capacitor always results in extra thermal power loss. When using the enhanced version of the detached power supply, the power loss is further increased due to the additional discharge phase of the capacitors. Generally, the power loss of a switched-capacitor circuit is proportional to its output current and inversely proportional to the switching frequency and the size of the capacitors [125]. Consequently, the power loss introduced by the detached power supply is reduced when selecting larger capacitors, increasing the switching frequency, and using a device with low power consumption. The requirement to utilize high switching frequencies conflicts with the results in Section 8.3, which illustrate that the switching frequency needs to be lowered to obtain a better resistance against power analysis. For the example of the passive UHF tag with the integrated AES module mentioned before, we have calculated a power loss of approximately 22 % when integrating over 100 clock cycles. There is a square-root relation between the power consumption of the tag and the maximum read range [41]. Doubling the power consumption of the tag results in a read range that is decreased by a factor of $\sqrt{2}$. Hence, a power loss of 22 % reduces the read range to approximately 88 % of the original value without detached power supply.

8.6 Summary

In this chapter, we have evaluated the effectiveness of the detached power-supply countermeasure to prevent power analysis and discussed its suitability for protecting passive UHF tags from parasitic-backscatter attacks. It is the first work that presents concrete results about the efficiency of this countermeasure with respect to side-channel analysis. DPA attacks have been applied to a smart card protected with a basic version of the detached power supply and an enhanced version that uses an additional discharge phase. A summary of the side-channel analysis results is given in Table 8.1. Due to the non-ideal properties of the deployed analog switches, even the enhanced version shows a susceptibility to power analysis. Moreover, we have depicted that the side-channel leakage of I/O pins poses a serious problem when utilizing the detached power supply. In order to address this issue, a simple decoupling principle for output pins has been presented. Additionally, we have provided an estimation concerning the required capacitor size and the power-consumption overhead introduced by integrating the detached power supply into a passive UHF tag.

Table 8.1: Summary of the side-channel analysis results obtained with basic and enhanced version of the detached power-supply countermeasure.

Countermeasure	**Integration interval**	**Measurement result**	**Required measurements**
-	[Cycles]	[Correlation]	[Traces]
None	-	0.55	75
Basic version	36	0.10	2 800
	100	0.04	17 200
Enhanced version	36	0.022	57 100
	100	< 0.017	$> 100\,000$

We conclude that the detached power supply significantly reduces the side-channel leakage in the power consumption of a cryptographic device, if the integration interval is sufficiently long and the utilized analog switches have adequate properties. Using this countermeasure to protect passive UHF tags from parasitic-backscatter attacks is feasible. However, longer integration intervals also increase the power loss caused by the detached power supply. A higher power loss results in reduced read ranges of the tags. Combining the detached power supply with other countermeasures, for example on algorithmic level, is indispensable if more sophisticated attacks need to be prevented as well. Such attacks involve manipulating the capacitors and measuring the direct emissions close to the tag chip.

This chapter closes the first part of this thesis that deals with implementation attacks on low-cost RFID tags as well as the evaluation of suitable countermeasures. In the second part of this thesis, implementation aspects of low-cost RFID tags are covered.

Part II

Hardware-Implementation Aspects of Low-Cost RFID Tags

9 Design of Digital Hardware Circuits

After the implementation attacks and countermeasure evaluations in the first part of this thesis, we now concentrate on implementation aspects of low-cost RFID tags in hardware. We start with preliminary information about the design of digital hardware circuits and emphasize the requirements of low-cost tags in this chapter. In the subsequent chapters, we present a flexible tag architecture that is suitable to integrate advanced tag functionality like file access and security features to future low-cost tags. We further show that a combined implementation of protocol and cryptographic algorithm on a limited microcontroller is highly advantageous in terms of resource usage.

During the last decades, integrated hardware circuits have become more and more popular and are an integral part of our daily life. Hardware circuits are not only found in personal computers and laptops, but also in cars, domestic appliances and in any device that has communication capabilities. Continuous migration to smaller process technologies has not only allowed to dramatically increase the transistor count but also to lower the power consumption of hardware circuits. Especially the continuously lowered power requirement of the circuits has pushed the development of contactlessly communicating devices such as smart phones, sensor nodes, and RFID tags.

For designing digital hardware circuits mainly two basic approaches are used: the general-purpose approach and the special-purpose approach. The general-purpose approach bases on hardware circuits like microprocessors that provide a fixed set of functionality (*i.e.* the instruction set of a microprocessor). Customization of the microprocessor for a concerning application is done through program development which provides high flexibility. The special-purpose approach on the other hand, involves design of a dedicated hardware circuit for a more specific application. This hardware-based concept is less flexible and

causes longer development times, but allows optimizing a design towards a certain goal, for example: low area, low power consumption, low energy consumption, or high throughput. Reducing the hardware overhead is desirable for cost-sensitive high-volume products that aim for minimum chip area. Achieving low power consumption or low energy consumption is important for passively powered devices (*e.g.* RFID tags) and battery-operated devices (*e.g.* smart phones), respectively.

9.1 Design Cycle

Today's digital hardware designers are faced with two challenges, increasing circuit complexity and shorter design cycles. Following Moore's Law, transistor count of hardware circuits doubles about every 18 months. This prediction has been formulated more than 40 years ago and is still adhered by semiconductor industry by migrating towards smaller and smaller process technologies. Most recent microprocessors, for example, have already reached a transistor count of one billion and more. Circuit complexity grows faster than productivity of designers and efficiency of electronic design automation (EDA) tools increase. This has opened a so-called "design gap" over the years. In order to close this gap, design of VLSI circuits has been brought to a higher abstraction level. Designing a circuit at a higher abstraction level also addresses the requirement of shorter design times to improve cost effectiveness.

A good overview of the different abstraction levels and design perspectives of digital hardware circuits is provided by the Y-diagram illustrated in Figure 9.1. The Y-diagram has been introduced by Gajski and Kuhn [60] in 1983 and has its name from the three axes that are arranged in a y-shape. Each axis relates to a different design perspective. The three design perspectives are: behavioral perspective, structural perspective, and geometric perspective. Behavioral perspective focuses on the functionality of a circuit, whereas structural perspective describes the interconnection of different blocks within it. Geometric perspective deals with the arrangement of the components, including the final layout of a circuit. Concentric circles indicate the various abstraction levels, which are: system level, architectural level, register-transfer level, logic level, and electrical level. Starting from highest abstraction level at the outermost circle, the various development steps of a hardware circuit are passed through when moving towards the center of the diagram, marking the final outcome of the design (*i.e.* layout of the circuit).

When moving towards the center of the diagram to reach the design goal, the level of detail increases continuously. Different perspectives can be used for entering lower abstraction levels and changing between perspectives is possible as well. Behavioral perspective is the most-suitable domain for describing digital hardware designs with high complexity. Consequently, most of today's designs use the behavioral perspective as starting point. First step is creating a software model that implements the specification of the system and that allows exploring different algorithm variants. This first software model also eases communica-

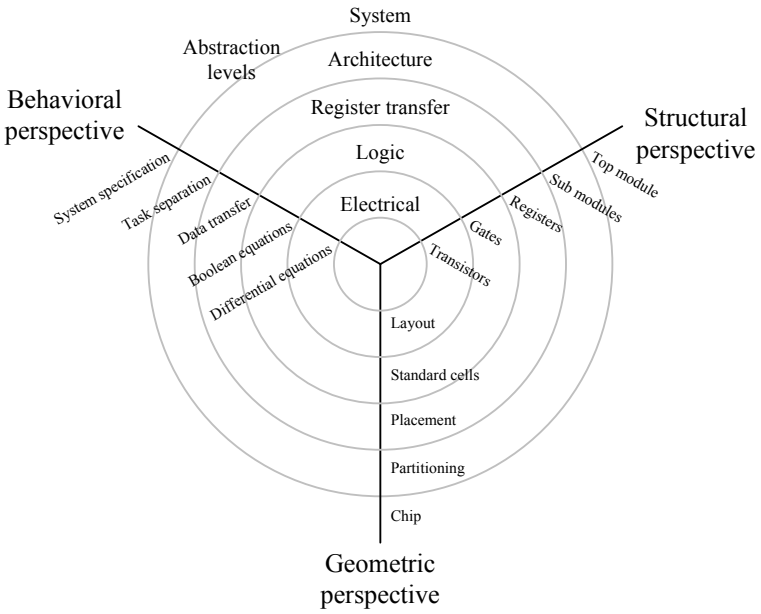


Figure 9.1: Y-diagram according to Gajski and Kuhn [60] showing the different design perspectives and abstraction levels of digital hardware circuits.

tion among design teams and enables concurrent development of hardware and software components in the following (important to shorten the overall development time). Next step is finding an appropriate architecture that is reflected by a cycle-accurate high-level model. When the architecture is fixed, hardware description languages (HDLs) like VHDL [82] and Verilog [83] are deployed to transfer the high-level model into a register-transfer level representation. The combined use of HDLs and EDA tools for circuit synthesis allows an automated transformation from behavioral perspective to structural perspective. Outcome of this step is a netlist that contains a circuit representation with logic gates, flip flops, and the appropriate wire connections.

The following steps after netlist creation relate to the so-called back-end design where the structural perspective is left and the geometric domain is entered. Automated tools are again applied to deduce a standard-cell representation and the layout of the design. During back-end design, various verification techniques are utilized to ensure proper operation and manufacturability of the circuit. Verification techniques comprise for example, design-rule checks, electrical-rule checks, layout-versus-schematic checks, timing verification, and simulation of power consumption. With the layout of the circuit, the final design step (tape out) is reached and data can be sent to a semiconductor manufacturer, where the microchips are produced.

Following this top-down approach gives a good understanding of the involved steps of state-of-the-art digital hardware design. Implementing a circuit within

behavioral perspective through HDLs and deploying automated tools for further processing eases not only handling of circuit complexity, but brings also more flexibility. A circuit in HDL representation can be easily mapped to different process technologies and targets by using circuit-synthesis tools. This significantly shortens the time required for migrating a design to a new process technology and allows also first-level tests on field-programmable gate array (FPGA) prototypes.

Continuously testing the functionality of a design within all abstraction levels is an important aspect of modern design methodology. Required test data is typically derived from the high-level model and repeatedly used for tests on lower abstraction levels. When a test fails, designers can immediately step back and fix the problem. This allows detection of issues as early as possible, following the first-time-right concept to launch products on time.

When building digital hardware circuits that contain security-relevant components, functional tests alone are no longer enough. As shown in the first part of this thesis, additional considerations have to be taken into account like evaluating the resistance of the implementation against side-channel analysis and fault analysis. Such evaluation tests are mainly conducted after chip production on first prototype samples, but also during design phase. Power-simulation results of the circuit can be used to deduce first information about side-channel resistance of a design. Another example are side channel and fault attacks on FPGA prototypes that contain a synthesized version of the design.

9.2 Design Space

Digital hardware circuits can be designed towards different optimization goals, depending on the targeted application. Typical optimization goals are: high throughput, low area, low power consumption, and low energy consumption. Optimization can be conducted on different abstraction levels. The higher the abstraction level the larger is the impact of the optimization techniques and the lower the required effort. Optimizing a design at system level or at architectural level is therefore more promising than optimizing it for example on logical level. Various metrics are used to quantify the effectiveness and the influence of a certain optimization measure. Widely used metrics among others are: chip area, throughput, execution time, maximum clock frequency, latency, and average power consumption.

Optimization at system level typically involves finding more suitable protocols or looking for alternative algorithms that lead to the same result but provide advantageous behavior in terms of computation time or resource usage. A good example is the representation of the substitution box (S-box) used in the AES. The S-box is a non-linear operation that is applied on a single byte of data. Hence, the result of the S-box operation can be precomputed for all possible 2^8 input values and stored in a look-up table. This will result in an area requirement of more than $1\,000\,GEs$ when implementing the look-up table with standard cells. However, the S-box operation can also be realized by calculating

the multiplicative inverse in the finite field $GF(2^8)$ followed by an affine transformation (see [131] for more details). Using combinatorial logic to calculate the S-box operation in that way, leads to an area requirement of 300 GEs. This is less than a third of the value required by the look-up table approach. Achieving such an area saving through optimization at lower abstraction levels is hardly possible.

Architecture is another abstraction level that has significant potential to optimize a design towards a certain direction. Well-known optimization techniques at architectural level are: functional decomposition, pipelining, and parallel computation [96]. Functional decomposition aims for breaking a complex function into smaller subfunctions that can be computed sequentially. This method is most effective when the subfunctions compute similar operations that allow to reuse a single hardware unit that decreases the overall chip area. Execution time remains roughly the same, since the shorter critical path allows a higher maximum clock frequency, which compensates the increased number of required clock cycles.

Pipelining is another effective optimization method at architectural level. The data path of a function is cut into smaller parts (ideally of equal length) by inserting storage elements called pipeline registers. This shortens the critical path and leads to a higher maximum clock frequency. For computing the result of one data item, as many clock cycles are required as there are pipeline stages. However, once the whole pipeline is filled, the result of a data item is computed with every clock cycle. It is important to note that this works only if there are no recursive data dependencies, since they would prevent the pipeline from getting filled. Pipelining is very efficient because a marginal increase of chip area that is introduced by adding pipeline registers, results in a significant computational speed up.

Computing operations in parallel is the opposite of functional decomposition. Instead of reusing components to reduce chip area, additional hardware modules are introduced to lower computation time. Trading chip area for speed is some kind of brute-force approach and is used if other measures like pipelining are not applicable (*e.g.* if low latency is required). In contrast to pipelining, the critical path of a design is not shortened and therefore increasing the clock frequency is not possible. Chip-area requirements increase significantly and relate to the degree of parallelism.

An overview of the ideal impact of all three optimization techniques within the design space is given in Figure 9.2. Functional decomposition and pipelining are efficient approaches to decrease chip area and execution time of a design, respectively. Both techniques significantly lower the area-time product. Parallel execution of operations increases chip area to lower execution time, by keeping the area-time product roughly constant. Especially functional decomposition is of interest when designing hardware circuits for low-cost RFID tags, since chip area is a highly limited resource in such designs.

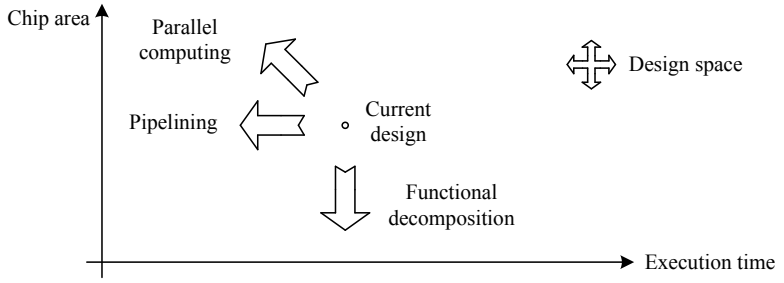


Figure 9.2: Ideal impact of functional decomposition, pipelining, and parallel computation on design space.

9.3 Testability

Testing is an important activity not only during development of a digital hardware circuit but also after its manufacturing. Typically, not all microchips that have been manufactured are working properly. This has various reasons, for example, varying process parameters during production or imperfections of material and masks. The yield, which is the ratio between the number of working chips and the overall number of manufactured chips, should be as high as possible to maximize the profit. In order to separate faulty chips from working chips, tests have to be applied.

Releasing a faulty chip causes tremendous costs. Imagine the following simple example: A company manufactures 100 000 chips, and sells them at the price of 1 $ per chip. We assume that one percent of the chips (*i.e.* 1 000 chips) are faulty. When the faulty chips are immediately detected after production through tests before they get sold, costs of 1 000 $ will arise. When the faulty chips get detected after they have been sold and soldered on a board, costs will already result in 50 000 $ if repairing a malfunctioning board costs 50 $. Even worse, when the failing parts get detected after integration into a whole system, costs will boost to 1 000 000 $ when repairing a non-working system costs 1 000 $. This simple example clearly emphasizes the need of detecting faulty parts as early as possible after production.

In order to get confidence about proper operation of a microchip after production, reliable tests are necessary. For realizing such reliable tests, the underlying test concepts that are used have to be planned and included already at design time of a hardware circuit. This so-called "design-for-test" approach integrates additional test structures to a circuit to allow fast and comprehensive analysis of a chip after production. The more internal details of a chip can be accessed, the more comprehensive tests can be conducted, lowering the chance that malfunctioning parts remain undetected.

A powerful and widely used test concept are scan chains that provide access to the values internally stored in the flip flops of a digital hardware circuit. For cryptographic devices, giving access to internal values can be problematic. As

shown in the work of Yang *et al.* [194, 195], test structures based on scan chains can be easily used to mount attacks against cryptographic devices. In order to prevent such attacks, test structures of security-related devices are typically deactivated after successfully testing the chip (*e.g.* by blowing a fuse) or even totally removed by cutting them off [105].

An alternative to scan-chain approaches are built-in self tests (BISTs). The National Institute of Standards and Technology (NIST) suggests to use BISTs instead of scan-chains for cryptographic devices [129]. For a BIST, necessary test data and test cases are generated within the evaluated microchip. The only information that is returned after conducting the tests is whether all tests have been passed or not. This is advantageous from a security point of view but comes at cost of a lower fault-detection rate, since comprehensive tests as with scan-chain approaches are not possible. Moreover, generating test data within the microchip causes significant hardware overhead.

9.4 Requirements for Passive Low-Cost RFID Tags

When designing digital hardware circuits for passive low-cost RFID tags, two important aspects have to be taken into account: chip area and power consumption. Chip area has an impact on the economics of the design. The larger the chip area the more expensive will be the resulting tag. However, when tags provide additional features (*e.g.* by integrating security functionality), even higher tag prices are acceptable. Power consumption on the other hand is limited from a technical point of view. Passive tags obtain their power supply directly from the RF field, which limits the maximum available power at a certain distance from the reader antenna. Hence, increasing the power consumption leads to shorter read ranges.

9.4.1 Chip Area

Low-cost tags are produced in high volume and have to be cheap to be profitable. Among other factors, chip area of a design has a strong influence of the resulting costs. The whole size of a low-cost tag chip is typically around 20 000 GEs (one gate equivalent (GE) is the silicon area required by a two-input NAND gate), including analog part and digital part [140]. A design with a larger chip area consumes not only more silicon on the wafer, but also increases the so-called edge effect and lowers the fabrication yield [96]. A wafer is a thin circular slice that is made of single-crystal silicon which serves as substrate for chip manufacturing. Due to the rectangular shape of the chip layout and the circular shape of the wafer, silicon around the circumference of the wafer is inevitably wasted, which is named edge effect. The larger the chip layout, the higher is the impact of the edge effect and the more silicon is wasted. Another aspect that is influenced by the size of the chip layout is fabrication yield. As mentioned above, the yield is the ratio between working chips and manufactured chips. As wafers have defects

included that are mostly randomly distributed, the probability that a single chip on the wafer includes a defect gets higher when the chip area increases.

Various techniques at different abstraction levels are used to minimize the chip area of a design. As described in Section 9.2, optimization at higher abstraction levels (*e.g.* system level and architectural level) has the most impact. In that way, proper selection of protocol, data coding, and architecture are very important for a compact design. A well-known optimization technique at architectural level is for example functional decomposition, where complex functions are broken into smaller subfunctions that are computed sequentially. For comparison, a standard hardware implementation of the block cipher AES requires about 16 kGEs [116] of chip area. When extensively applying functional-decomposition techniques, area requirement can be reduced to less than a quarter (3.4 kGEs) of it, as shown by Feldhofer *et al.* [57].

Another possibility to reduce the chip area of a design is to step to a more advanced CMOS process technology. When passing over, for example, from a 180 nm technology to a 130 nm technology, the number of CMOS transistors per unit area is roughly doubled. Hence, more functionality can be integrated within the same area. Today's RFID tags are typically fabricated on 180 nm and 130 nm CMOS processes. Within the next one or two years, it is expected that even 90 nm CMOS processes are used for RFID tag production [74]. Most recent CMOS process technologies (*e.g.* 45 nm, 32 nm, 22 nm) as they are used by high-performance processors are not applicable for RFID tag fabrication. Such technologies are still too expensive (especially production of the masks) and integration of non-volatile memories and mixed-signal designs are much more difficult [56].

9.4.2 Power Consumption

The power budget of passive RFID tags is limited and mainly depends on three factors: the coupling method of the RFID system, the RF output power of the reader, and the distance between reader antenna and tag. The two most-prevalent coupling methods are inductive coupling and electromagnetic coupling. Inductive coupling is used by RFID tags operating in the LF and the HF range, whereas UHF tags base on electromagnetic coupling. The relation between the power P_{Tag} that is available at the antenna-pins of the tag and d which indicates the distance between reader antenna and tag, is approximately $1/d^3$ for inductively coupled tags. Hence, with increasing distance d, the available power at the tag rapidly decreases. Electromagnetically coupled tags on the other hand have a $1/d$ relation between distance d and available power P_{Tag} and can thus have much larger read ranges when passively powered [59]. Table 9.1 gives typical values of P_{Tag} for inductively coupled and electromagnetically coupled tags for different distances d. When assuming an efficiency factor of 0.2 for the analog front-end of the tag (*e.g.* due to losses in the rectifier), we can derive the maximum average power-consumption value I_{Dig} of the digital hardware circuit. For a 130 nm CMOS process technology with a supply voltage of $1.2\,V$, this results in average power-consumption values of some milliamps for HF tags close to the

reader antenna and some tens of microamps several meters away from the reader antenna for UHF tags.

Table 9.1: Typical power values of HF and UHF tags at different distances from the reader antenna.

Coupling method	**Distance** d	**Tag power** P_{Tag}	**Circuit current** I_{Dig} **@ 1.2 V**
-	[m]	[mW]	[mA]
Inductive (HF tag, 13.56 MHz)	0.1	29	4.83
	0.5	7.4	1.23
	1	0.43	0.072
Electromagnetic (UHF tag, 868 MHz)	1	4.0	0.667
	3	0.5	0.083
	5	0.16	0.027

For passive RFID tags, lowering the average power consumption is important to achieve reasonable read ranges. In contrast to battery-operated devices, energy efficiency is not a design goal for passive RFID tags. As long as the reader field is switched on, the tag is constantly supplied with power. Further, data rates of RFID systems are rather low, which makes it attractive to use time spreading of computations to lower the average power consumption. The most-effective design strategy to lower the average power consumption is again to optimize on higher abstraction levels and to combine several approaches. On system level for example, unused components can be completely switched off to save power. Functional decomposition can be applied on architectural level to lower the average power consumption (equals time spreading). Pipelining and parallel computing can principally also be used to lower the power consumption, since clock frequency and supply voltage can be lowered when the obtained speed gain is not needed. However, both pipelining and parallel computing are less favorable in terms of chip-area requirement than functional decomposition. On register-transfer level, idling circuit parts can be deactivated by turning off their clock signal via so-called clock-gating cells. Using more-advanced CMOS process technologies for fabrication is another option to lower the power consumption of a design. As mentioned above this measure can only be applied to a certain extent since most-recent CMOS processes are not suitable for RFID tag production.

9.5 Summary

In this chapter we have provided basic information about the design of digital hardware circuits. We have started with a description of the design cycle and discussed different optimization techniques in the design space. Further, the importance of testing within the design of hardware circuits has been pointed out. As a last point, we have focused on the requirements of passive low-cost

RFID tags in terms of chip area and power consumption. Chip area is not limited per se, but has a high impact on the tag costs. However, when tags provide additional features (*e.g.* by integrating security functionality), even higher tag prices are acceptable. Power consumption on the other hand is a technological limitation, since passive tags receive their power from the reader field. Various design techniques at different abstraction levels can be applied to lower both chip area and power consumption.

10

Hardware Implementation of a Flexible Tag Platform for Passive Low-Cost RFID Tags

Designing digital hardware circuits for passive low-cost RFID tags is challenging due to the fierce requirements in terms of chip area and power consumption. As we have pointed out in the previous chapter, such tags are produced in high volume and need to be cheap in price. In order to keep the price low, the chip area must not exceed a certain size. Further, passive tags are supplied by the RF field of the reader, which strongly limits the power consumption of the tag.

Especially passive RFID tags (more than 2 billion items have been sold in 2010) are the enabler for the future *Internet of Things* where every object equipped with a tag has communication capabilities. Even the latest generation of smart phones (e.g. Nexus S, Blackberry Bold 9900) integrate RFID functionality, namely near-field communication (NFC) technology. With this expected spreading of RFID readers many new applications are arising. However, not only reader devices are required for these applications, also the tags' functionality must increase and their design must get more flexible.

Currently, there exist two categories of passive RFID devices. The first category are "stupid" tags which can only transmit their unique identifier (UID) and have a small amount of non-volatile memory. Typically, these are low-cost tags that are implemented with a hardwired finite-state machine (FSM) approach because the design goal is smallest chip size and lowest power consumption [41, 59]. The implementation of this FSM approach is very time consuming and also inefficient when control complexity increases. In the second category are contactless smart cards which can perform complex operations like cryptographic primitives

and protocols. Such devices are programmable in the field and have powerful microcontrollers integrated which are rather expensive in terms of chip area and power consumption [85, 138, 173].

A substantial part of the chip size of current low-cost tags is consumed by the controlling unit that handles the communication protocol. The work of Yu *et al.* [202] states that about 7 500 gate equivalents (GEs) (one GE is the silicon area required by a NAND gate) are consumed for only handling the protocol. Hence, when integrating additional functionality such as file access and security features, controlling effort further increases.

In this chapter we present the implementation of a flexible tag platform that targets on passive low-cost RFID tags concerning its chip size and power requirements but allows increasing functionality due to a programmable solution using an integrated 8-bit microcontroller. This highly optimized microcontroller reduces the overall complexity of protocol handling compared to state machines and makes the design very flexible in terms of reuse of components and protocol adaption. We show how we used the presented flexible architecture to implement an NFC-compatible tag that provides advanced file-access functionality and security features.

All the work that is presented in this chapter and the chapters afterwards that comprises hardware implementation aspects of low-cost RFID tags has been carried out from 2009 to 2011 within the nationally funded project "**Cryp**tographic Protected **T**ags for new RFID **A**pplications" (CRYPTA). During this project, a fully working RFID tag-prototype chip has been designed and manufactured. Project partners have been the semiconductor manufacturer Austriamicrosystems and the RFID software and service provider RF-iT Solutions GmbH. The information about the flexible tag architecture provided in this chapter has been published at the DSD conference 2011 [150] and is a joint work with Martin Feldhofer. Cryptographic unit and EEPROM that are used by the flexible tag platform have been implemented within the CRYPTA project by Michael Hutter and Austriamicrosystems, respectively.

The remainder of this chapter is structured as follows. In Section 10.1, we give an overview about the tag's flexible platform while Section 10.2 describes the functionality of our implementation. The separation of the tasks in hardware and software is depicted in Section 10.3. The components of the flexible tag platform are elaborated in Section 10.4 and we show how to develop code for the microcontroller in Section 10.5. Section 10.6 gives insights into the hardware implementation and provides results. We present the manufactured prototype chip in Section 10.7 and close the chapter with a summary in Section 10.8.

10.1 Overview of the Flexible Tag Platform

The flexible tag platform described in this work is intended for efficiently handling complex control tasks on resource-constrained devices like passive RFID tags. In order to demonstrate the effectiveness of our approach, the digital part of an NFC-compatible tag with advanced file-access functionality and security

features has been implemented.

The design concepts that we use for our flexible tag platform are typically found in contactless smart cards [189, 192]. Such smart cards rely on powerful microcontrollers that are expensive in terms of power consumption and chip area, making them unsuitable for resource-constrained devices. The issue of power consumption can be addressed by replacing the powerful microcontroller with a microcontroller optimized for low power consumption. Several commercially available microcontroller platforms exist that are designed towards minimizing the power consumption, for example, PIC16 from Microchip [122], C8051 from Silicon Labs [170], and MSP430 from Texas Instruments [179]. Microcontrollers like that are often used for sensor-enabled tags with enhanced functionality [3, 119, 126, 198]. However, the resource usage of such microcontrollers regarding chip area is still beyond that what is acceptable for low-cost RFID tags.

First attempts to make low-cost tags more flexible are reported in the work of Yan *et al.* [193] and Yu *et al.* [202]. Both authors use a very application-specific processor for handling the protocol of Electronic Product Code (EPC) Generation 2 tags [50]. Compared to the work of Yan *et al.* and Yu *et al.*, our tag platform is kept more generic and designed towards handling even more complex protocols, as demonstrated with our implementation of an NFC-compatible tag.

Central component of our flexible platform is an efficient 8-bit microcontroller with very limited functionality that handles most of the control tasks on the tag. The design concept of the microcontroller bases on a 4-bit microcontroller published by Feldhofer [53], which has been used there to handle simple control tasks on an ISO 18000 tag. We have not only extended the bit width from 4 to 8 bits, but also completely modified the instruction set (*e.g.* to support instructions that operate on two input registers) and optimized several components towards lower resource usage and lower power consumption (*e.g.* by using latches). Besides the microcontroller, the digital part of the tag consists of: a framing logic (FL), a bus arbiter, an EEPROM, and a cryptographic unit (CU). Figure 10.1 gives an overview of the digital components of the tag. The framing logic does only low-level protocol handling and provides a byte-level interface between the microcontroller and the air interface. The EEPROM of the tag has been designed by Austriamicrosystems and stores numerous files that contain, for example, the UID, user and configuration data. In order to provide security features, a cryptographic unit is used to generate random numbers and for signing and encrypting/decrypting data. Random numbers are generated according to the FIPS 186-2 standard [130] using a pseudo-random number generator based on the SHA-1 algorithm. Signing of data is done with the Elliptic Curve Digital Signature Algorithm (ECDSA) based on the recommended $\mathbb{F}_{p192}$ NIST elliptic curve. For encrypting and decrypting data, the AES with a key length of 128 bits is used. More details about the cryptographic unit can be found in the work of Hutter *et al.* [77]. All components are connected by an AMBA Advanced Peripheral Bus (APB) that is controlled by the bus arbiter. AMBA stands for Advanced Microcontroller Bus Architecture and is a widely deployed bus concept for on-chip communication [13]. In general, the data-bus

width is 16 bits. However, the framing logic uses only 8 bits of the data bus since it operates on byte level. Both the cryptographic unit and the framing logic also share some *direct signals* with the microcontroller. The direct signals reduce the communication overhead for handling time-critical events.

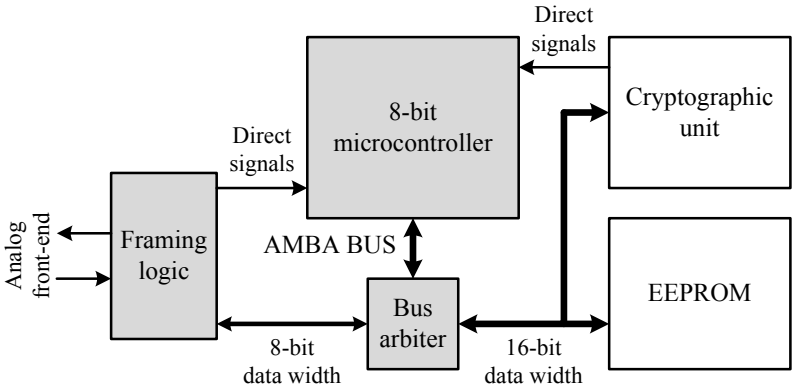


Figure 10.1: Overview of the tag's digital components.

A description of the functionality supported by the NFC-compatible tag is provided in the next section. Details about the flexible tag platform follow afterwards.

10.2 Functionality of the NFC-Compatible Tag

NFC devices operate in the high frequency (HF) range at a carrier frequency of 13.56 MHz and are divided into different types (NFC Type 1 to NFC Type 4). Our NFC-compatible tag is based on the NFC Type 4 specification [134] and uses the ISO 14443A protocol standard. The tag provides basic tag functionality fully compliant to ISO 14443-3 [91] covering tag initialization and anticollision, as well as advanced tag functionality fully compliant to ISO 14443-4 [93] covering a block-transmission protocol. The block-transmission protocol is used to exchange application protocol data units (APDUs) that are specified in ISO 7816-4 [87]. Using such APDUs for data exchange is widely deployed in the area of contactless smart cards. Low-cost RFID tags on the contrary typically have only basic tag functionality that allows participating in an anticollision sequence with a UID and storing some additional information in the EEPROM.

Hereafter, a short overview of basic and advanced functionality of our tag as well as the supported commands is given.

10.2.1 Basic Tag Functionality

Basic tag functionality covers an initialization and anticollision phase which has the purpose to get the UID of all tags available in the reader's field. The com-

mands used in this phases are: request (REQA), wake-up (WUPA), anticollision (AC), SELECT, and halt (HLTA). A REQA or a WUPA command starts the initialization phase. Tags that have been initialized can continue with the anticollision phase. There, multiple AC commands are used by the reader to identify the UID of all available tags. Potential collisions in the tag responses are dissolved with a special binary-search approach (tree-walking). After obtaining the UID of a single tag, a SELECT command is transmitted by the reader, which brings the corresponding tag into an activated state. A tag in activated state can continue with advanced tag functionality or it can be brought into some kind of sleep state by sending a HLTA command to it. Tag-response time during initialization and anticollision phase is rather short (approx. 8.5 µs). This is the reason why the low-level commands are implemented in a dedicated state machine in the framing logic. However, as soon as the tag is in activated state, response time becomes much longer (several milliseconds).

10.2.2 Advanced Tag Functionality

Tags that are in activated state can enter the block-transmission protocol which allows accessing advanced tag functionality. Block transmission is initiated with a request-for-answer-to-select (RATS) command followed by an optional protocol-and-parameter-selection (PPS) command. The RATS command defines basic parameters for block transmission and the optional PPS command allows switching to higher data rates. After setting up all required parameters block transmission can start. Three types of blocks exist: I-blocks, R-blocks, and S-blocks. I-blocks are used to exchange application data between reader and tag. R-blocks are used to give positive (ACK) or negative (NAK) acknowledgement of the last received block. S-blocks contain control information and are deployed to temporarily extent the answer time of the tag (WTX command) or to stop block transmission (DESELECT command).

Application data is carried by I-blocks and is encapsulated via APDUs that consist of several fields: command header (class (CLA), instruction (INS), and parameters (P1 and P2)), command-data length (Lc), command data (DATA), expected response-data length (Le), and response trailer (SW1 and SW2). Not necessarily all fields need to be present in every APDU. When APDUs are too long to fit within a single I-block, they are split into smaller parts and transmitted via multiple I-blocks. This procedure is called chaining and makes the controls tasks on tag side even more demanding.

In order to realize advanced tag functionality, our NFC-compatible tag supports six commands on application level that are transmitted via APDUs. Three commands are related to file-access functionality and another three commands are used for security features. Figure 10.2 provides an overview of the commands with all their parameters.

File-access functionality File-access functionality on the tag is realized with three commands: SELECT_FILE, READ_BINARY, and UPDATE_BINARY.

Command	CLA	INS	P1	P2	Lc	DATA	Le
SELECT_FILE	00h	A4h		00h			
By identifier			00h		02h	File ID	--
By name			04h		07h	File name	--
READ_BINARY	00h	B0h	Offset		--	--	# bytes
UPDATE_BINARY	00h	D6h	Offset		# bytes	Data to write	--
INTERNAL_ AUTHNETICATE	00h	88h		00h			
Using AES			00h		08h	Challenge	10h
Using ECDSA			01h		10h	Challenge	30h
GET_CHALLENGE	00h	84h	00h	00h	--	--	08h
EXTERNAL_ AUTENTICATE	00h	82h	00h	00h	10h	E_K(Challenge)	--

Figure 10.2: Commands for file-access functionality and security features.

SELECT_FILE allows selecting a file by its identifier or its name for further read or write access. READ_BINARY reads data from the selected file, whereas UPDATE_BINARY writes new data to the selected file. Both commands allow specifying the number of bytes that need to be read/written and an offset pointer within the file were the file access starts. Files are stored in the EEPROM of the tag. For our tag, we have defined 12 files that contain: encryption keys for the cryptographic module, configuration parameters of the tag, data that is required for NFC compatibility, and user data. Depending on the file and the configuration parameters of the tag, different read and write access to the files is realized. Files with user data, for example, can be configured such that they are freely accessible. Files that contain configuration data on the contrary, require prior authentication before access is granted.

Security features The security features involve tag authentication and reader authentication and rely on the following three commands: INTERNAL_AUTHENTICATE, GET_CHALLENGE, and EXTERNAL_AUTHENTICATE. Tag authentication can either be done symmetric using the Advanced Encryption Standard (AES) [131] or asymmetric using the Elliptic Curve Digital Signature Algorithm (ECDSA) [132]. For reader authentication only AES is supported. Required operations for the security features, like generating random numbers as well as signing, encrypting, and decrypting data are performed within the cryptographic unit. Tag authentication is achieved with the INTERNAL_AUTHENTICATE command, which uses AES if the parameter byte P1 is 00h and ECDSA if P1 is 01h. The command contains a challenge (randomly generated data) from the reader. When using AES, the challenge is first encrypted by the tag and then transmitted to the reader. Because both reader and tag share a secret key, the reader can verify whether the tag could successfully authenticate or not. When using ECDSA, the tag signs first the challenge with its secret key and transmits it to the reader. With the public key of the tag, the reader can verify the signa-

ture and prove the authenticity of the tag. A similar scenario is used for reader authentication. There, the reader sends first a GET_CHALLENGE command to the tag, which causes the tag to generate a challenge that is transmitted to the reader. The challenge is then encrypted by the reader and sent to the tag with an EXTERNAL_AUTHENTICATE command. The tag can decrypt the encrypted challenge and compare it to the original challenge transmitted to the reader. If both values match, the reader is treated as authenticated. More detailed information about the operations performed by the cryptographic unit can be found in [76, 77]. As mentioned before, reader authentication is also used for file-access functionality. Hence, reading from or writing to certain files (*e.g.* to configuration files) is only granted after the reader has successfully authenticated itself towards the tag. Moreover, because the execution of cryptographic operations like encryption or decryption of data typically requires a significant amount of time, additional waiting-time-extension (WTX) commands are deployed. In that way, the tag obtains enough time to process the received data and to prepare the appropriate response.

10.3 Splitting Functionality into Hardware and Software

Looking at the previous section makes clear that integrating advanced tag functionality results in a significant amount of controlling complexity. Data has to be transmitted from one component to another. Commands that are split into several blocks (*i.e.* chaining of data) need to be reconstructed. Moreover, commands have to be handled according to their parameters, the configuration of the tag as well as the current tag state. A tag platform that is based on a microcontroller can better cope with such increased controlling complexity than a conventional state-machine approach. However, when using a microcontroller, the fierce requirements of passive RFID tags in terms of chip area and power consumption have to be fulfilled. Consequently, only a very simple microcontroller with a small chip size can be deployed. In order to keep the power consumption low, the microcontroller should be clocked with the lowest possible clock frequency.

Processing all control tasks with the microcontroller would result in a high clock frequency due to the short tag-response time during initialization and anticollision phase. In order to address this issue, functionality that is related to the initialization and anticollision phase (basic tag functionality) is directly handled by a dedicated hardware circuit. Since controlling complexity of basic tag functionality is low, implementation in hardware is achievable. Moreover, basic tag functionality is independent of the overlaying application data and consequently does not affect the flexibility of the platform. Advanced tag functionality on the contrary leads to rather high control complexity but has relaxed timing requirements that make an implementation in software on the microcontroller highly favorable. Figure 10.3 illustrates how the tag's functionality is divided

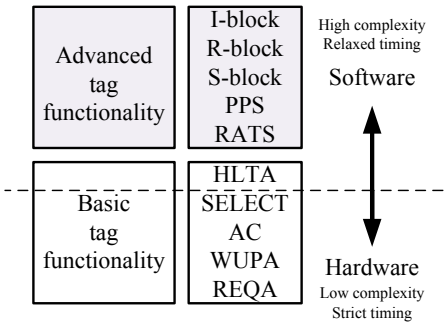


Figure 10.3: Hardware-software separation of basic and advanced tag functionality.

into hardware and software. Commands up to SELECT in the protocol stack are handled in hardware. All other commands are processed in software (note that the HLTA command requires no tag response and can thus also be processed in software).

10.4 Detailed Description of the Flexible Tag Platform

An overview of all the components of the flexible tag platform has already been given in Section 10.1. Hence, only the components that are relevant for handling the control tasks on the tag are described here in a more detail. The relevant components are the framing logic and the microcontroller. The bus arbiter is only a small circuit with very limited functionality and thus omitted for simplicity.

10.4.1 Framing Logic

The framing logic is some kind of serial-to-parallel interface that handles basic tag functionality. Figure 10.4 sketches the architectural overview of the framing logic with the following main blocks: receive-and-transmit (RxTx) unit, control unit, and AMBA interface. The RxTx unit is the interface between the serial data signals of the analog front-end and the parallel data signals of the control unit. Additionally, the RxTx unit receives a clock signal from the analog front-end, which is used to extract a bit-clock signal that is provided to the microcontroller. Hence, for a default data rate of 106 kbit/s, the resulting bit-clock signal has a frequency of 106 kHz. Incoming serial data from the analog front-end is first sampled, decoded into bits, transformed to byte data, and checked for integrity (parity bits and CRC). Incoming byte-level data from the control unit is first appended with a checksum, encoded, and then transmitted bit by bit. The RxTx unit is also responsible for proper timing of the tag response, which needs to be transmitted within certain time slots. The control unit steers the

RxTx unit as well as the AMBA interface and handles also the initialization and anticollision phase of the tag (up to SELECT command). All data received after a SELECT command are no longer handled by the control unit and are directly forwarded to the AMBA interface instead. The AMBA interface places this data into a so-called first-in first-out (FIFO) buffer that is accessed by the microcontroller over the AMBA bus. The FIFO buffer can store up to six bytes and decouples the communication between control unit and microcontroller. When data coming from the microcontroller needs to be transmitted by the framing logic it is first placed in the FIFO buffer and then forwarded by the control unit to the RxTx unit.

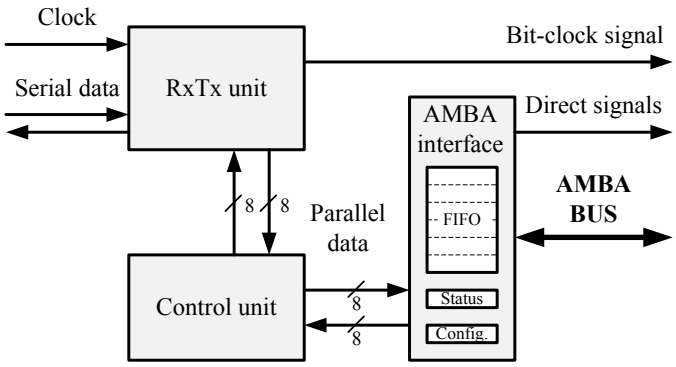


Figure 10.4: Overview of framing-logic architecture.

The AMBA interface connects the framing logic with the AMBA APB bus. Although the data width of the AMBA bus is 16 bits, only the lower 8 bits are used by the framing logic since it operates on byte level. The AMBA interface also contains a status register that provides information about the internal state of the framing logic (*e.g.* indicates framing error of incoming data) and a configuration register that allows the microcontroller to adjust some parameters of the framing logic. Both registers can be accessed by the microcontroller via the AMBA bus. Moreover, to give full control over the framing logic to the microcontroller, several commands can be sent to the framing logic via the AMBA bus. The commands allow, for example, starting and stopping the operation of the framing logic, interrupting the reception of data, and initiating transmission of data. Besides the AMBA bus, some additional direct signals are used between framing logic and microcontroller. The direct signals give information about the actual number of utilized bytes in the FIFO buffer and indicate whether the last byte of a command has been received or not (end-of-frame indication).

10.4.2 8-Bit Microcontroller

The architecture of the 8-bit microcontroller targets on low chip area and low power consumption for replacing conventional state machines that make a design inflexible and modifications very costly. In comparison, contactless smart

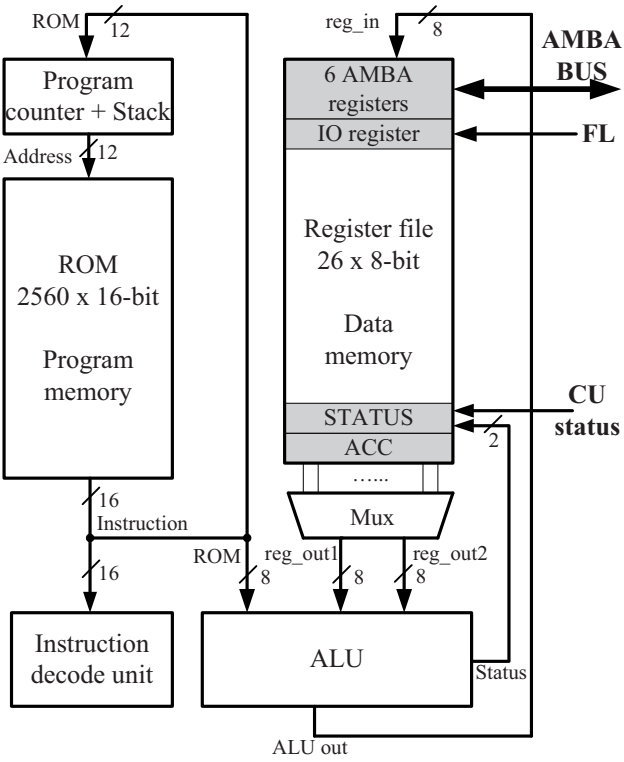


Figure 10.5: Overview of microcontroller architecture.

cards have often 32-bit controllers integrated that can perform rather expensive computations [85, 139, 173]. Our implemented microcontroller combines the advantages of hardwired state machines and complex controllers. It keeps the design programmable while consuming only a limited amount of hardware resources. The controller is the central element in the tag design as it steers all other modules via a memory-mapped AMBA APB bus or direct interfacing. Moreover, it is fully synthesizable for standard-cell technology but using an integrated program ROM macro is also possible.

An overview of the microcontroller architecture is depicted in Figure 10.5. The design uses a Harvard architecture which has the advantage that the separated 8-bit data memory and the 16-bit program memory can have different word sizes. The microcontroller supports 31 instructions which can be divided into logical operations (AND, OR, XOR, MOV, ROT, SHIFT), arithmetic operations (ADD, SUB, INC, DEC) and control-flow operations (GOTO, CALL, RET, conditional branching). A detailed overview of all instructions with name, number of clock cycles, and a short description is given in Table 10.1. In order to reduce overhead no interrupts are supported which means that polling has to be implemented when waiting for an event.

Table 10.1: Overview of the instruction set used by the low-resource 8-bit microcontroller. Each instruction is listed with its type, the name, the number of cycles, and a short description.

Type	Name	Cycles	Description
Constant	AndLF	1	Logical AND register and constant
	OrLF	1	Logical OR register and constant
	MovLF	1	Move constant to register
	XorLF	1	Exclusive OR register and constant
Branch	GOTO	2	Unconditional branch
	CALL	2	Subroutine call
	BZ	1/2	Branch if zero
	BNZ	1/2	Branch if not zero
Micro	MICRO	N	MICRO instruction for cryptographic unit
Register to register	MovFF	1	Move register to register
	AndFF	1	Logical AND registers
	XorFF	1	Exclusive OR registers
	AddFF	1	Add two registers
Conditional branch	BTC	1/2	Skip next instruction if bit is cleared
	BTS	1/2	Skip next instruction if bit is set
	BWC	1	Wait until bit is cleared
	BWS	1	Wait until bit is set
Work register	AddLW	1	Add constant to ACC
	SubLW	1	Subtract constant from ACC
	RetLW	2	Move constant to ACC and leave subroutine
Register to target	OrWF	1	Logical OR ACC and register
	RotlWF	1	Rotate left through carry
	RotrWF	1	Rotate right through carry
	ShlWF	1	Shift left through carry
	ShrWF	1	Shift right through carry
	DecWF	1	Decrement register
	IncWF	1	Increment register
	DecTWF	1/2	Decrement register and branch if zero
	IncTWF	1/2	Increment register and branch if zero
Others	RET	2	Return from subroutine
	NOP	1	No operation

The main components of the microcontroller are the register file, the program counter, the program memory, the arithmetic-logic unit (ALU) and the instruction decode unit. The register file contains the data memory and consists of 26 8-bit registers. Although potentially 32 registers are addressable we reduced the size to 26 (minimum number of registers required for handling the protocol) which reduces the overall chip size and emphasizes the flexibility of our approach. The register file contains a set of general-purpose registers for storing variables and the internal state and special-purpose registers. These special-purpose reg-

isters (accumulator (ACC), status register (STATUS), 6 AMBA registers, IO register) are used for advanced data manipulation, status information like carry or the external status of a device, the AMBA bus access, and for the direct access of information from the framing logic. The AMBA APB bus uses six registers whereas four are data registers which allow fast access of external modules of various bit size (8-bit and 16-bit), one is the AMBA address register, and one controls the bus access.

Instructions are executed within a two-stage pipeline that consists of a fetch and a decode-execute step. First, the instruction that is addressed by the 12-bit program counter is loaded from the program ROM into the instruction decode unit. Then the instruction is decoded by the instruction decode unit and executed by the ALU. Finally the program counter is updated. The program counter contains a call stack that allows up to four recursive subroutine calls. All instructions are executed within a single clock cycle, except control-flow operations which require two clock cycles. The ROM contains the program of up to 4 096 instructions and is realized as look-up table in hardware. The ROM is also flexible where we instantiate only 2 560 instructions in the current design.

10.5 Design Flow for Code Development

The code development for the microcontroller is the central aspect which makes our design approach very flexible. We have implemented a self-written tool chain that provides instruction-set simulation and assembler functionality. An overview of the design flow for code development is depicted in Figure 10.6. The program itself is written in assembler style but uses Java syntax that is actually a Java file. This avoids that we have to write parsing functionality and we can use Java for preprocessing, constant definitions and the like. Both the instruction-set simulator (ISS) and the assembler use a common instruction-set architecture definition (also based on Java). Further controller configuration that defines for example the available number of registers, the stack depth etc. are used in the simulator only.

The simulator additionally allows to integrate models of IO modules and other components like cryptographic circuits. The simulation itself provides features like single-step mode and gives access to the internal state of the microcontroller. This makes debugging and testing of the program very convenient. The IO stimuli data that model the incoming data are first processed by the IO module and the response can be written to external files which allows a comparison of expected and actual data on protocol level. Furthermore, the simulator provides statistical data on the simulation run. This is information about the execution time, which instructions are used how often, which parts of the code are never executed and many more. Whenever the developed program is working in the simulator we use the assembler tool for code generation. The assembler is used to transform code from assembly language to a binary representation based on the instruction set. It also dissolves addresses of labels that are used for branching operations. As the first and most important output it generates

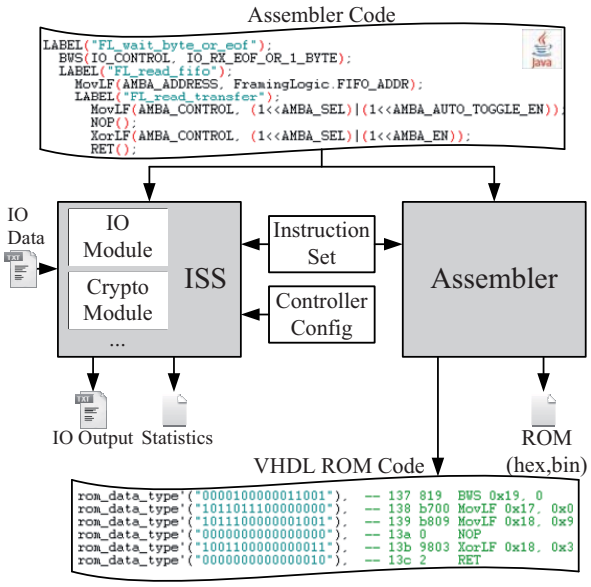


Figure 10.6: Design flow for program development.

an HDL code of the ROM as a look-up table which can be subsequently used for synthesis or for HDL simulation. Furthermore, it provides the data in a hex-file format and in a representation used for ROM macro implementation. Figure 10.6 also shows a tiny code example of the assembler code which is then translated to the shown VHDL code and a binary representation.

10.6 Implementation Results

We have implemented our flexible tag platform in VHDL and designed it towards low-resource usage and low power consumption (*i.e.* by heavily using clock gating). In the following, details about the ROM code for the microcontroller is given, information about area requirement and power consumption of the flexible tag platform are provided, and results are compared with related work.

10.6.1 ROM Code for the Microcontroller

The program of the microcontroller has been first developed and evaluated with the instruction-set simulator described in Section 10.5. After verifying the correct operation of the program, the assembler was used to transform the assembly code into VHDL ROM code. This ROM code is directly integrated into the VHDL model of the microcontroller. Proper operation of the whole flexible tag platform has been further verified through simulations with Cadence NC Sim and through tests on an FPGA prototype with real RFID-reader devices.

Figure 10.7: FPGA prototype communicating with an NFC-enabled mobile phone.

Figure 10.7 shows a photo of the FPGA prototype communicating with an NFC-enabled mobile phone. A detailed description of the FPGA prototype can be found in the work of Feldhofer *et al.* [54]. The final ROM code for the microcontroller that implements all the tag functionality described in Section 10.2, contains 1261 instructions (equals 2 522 bytes of code). Subroutine calls are used whenever possible to keep code size small. Table 10.2 shows the distribution of the ROM code with respect to tag functionality. Most instructions of the ROM code, about 40 %, are only used for handling the block-transmission protocol. Nearly 25 % of the instructions are utilized for generic subroutines that provide a basic set of functions that are reused multiple times (*e.g.* routines for accessing the AMBA bus). File management and security features require about 26 % and 9 %, respectively.

Table 10.2: Distribution of ROM code with respect to tag functionality.

Tag functionality	**Code size**	
	[Instructions]	[%]
Generic subroutines	312	24.7
Block transmission	499	39.6
File management	331	26.3
Security features	119	9.4
Total	**1 261**	**100.0**

10.6.2 Chip Area and Power Consumption

Two requirements have to be fulfilled when designing circuits for low-cost RFID tags: chip area and power consumption. In order to verify the suitability of our flexible tag platform for this design target, synthesis for a 0.35 µm CMOS process technology was performed and power simulations of the microcontroller were conducted (EEPROM and cryptographic unit are not considered here).

Table 10.3: Synthesis results of the flexible tag platform (excluding EEPROM and cryptographic unit).

Component	Chip area	
	[GEs]	[%]
Microcontroller	**7 248**	**72.3**
Program counter	495	4.9
ALU	202	2.0
Register file	1 693	16.9
Instruction decode unit	248	2.5
Program ROM	4 610	46.0
Framing logic	**2 449**	**24.4**
RxTx module	1 035	10.3
Control unit	761	7.6
AMBA interface	653	6.5
Bus arbiter	**319**	**3.2**
Total	**10 016**	**100.0**

Synthesis has been done with Cadence RTL Compiler using a semi-custom design flow. Table 10.3 presents the synthesis results. The chip area for each component is given in terms of gate equivalents (GEs) and in percentage of the total chip area. The whole flexible tag platform requires a chip area of about 10 kGEs (note that EEPROM and cryptographic unit are not considered here). As expected, most of the area, around 70 %, is consumed by the microcontroller. The framing logic follows with about a quarter of the total chip area. The bus arbiter is the smallest component and consumes only around 3 % of the total chip area. By far largest sub component is the program ROM of the microcontroller with more than 4 600 GEs. However, when using a ROM macro instead of a look-up table, this value can be further decreased. Another large block is the register file of the microcontroller with a size of about 1 700 GEs. As depicted in Figure 10.8, the register file uses latches for the general-purpose registers instead of flip flops, saving around 300 GEs of chip area. The area requirement of the register file can be further decreased by utilizing a RAM macro for the general-purpose registers.

Power simulations of the microcontroller were conducted with Synopsys Nanosim. Figure 10.9 presents results of a power simulation (power consumption I and mean power consumption I_mean) while the microcontroller is handling two reader requests with advanced tag functionality. The simulations show a mean power consumption of about 10 µA for the 0.35 µm CMOS process technology when operating the microcontroller with a supply voltage of 2 V and a clock frequency of 106 kHz. This clock frequency equals the bit-clock signal that is provided by the framing logic when a default data rate of 106 kbit/s is used. When higher data rates are selected, the power consumption increases accordingly (linearly with data rate). The overall power consumption of the microcontroller for

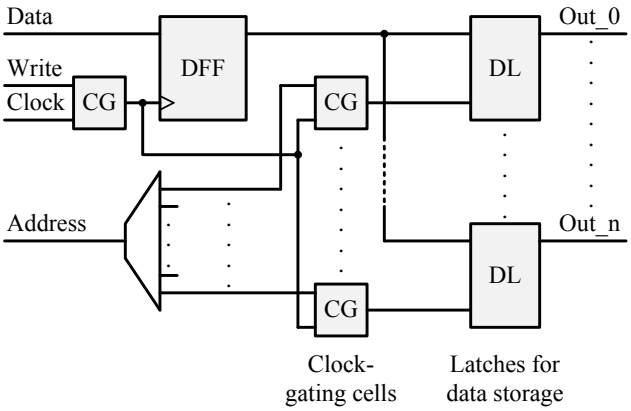


Figure 10.8: Latch-based register file for the general-purpose registers of the microcontroller to reduce chip area.

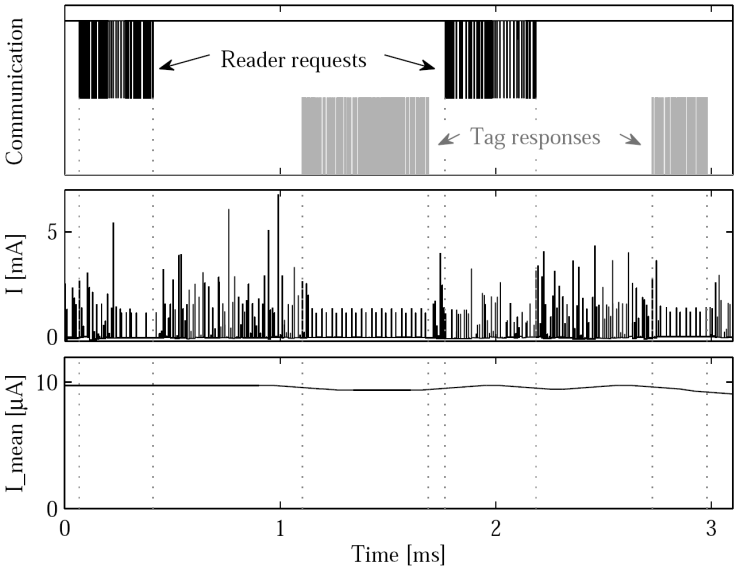


Figure 10.9: Simulated power consumption I (middle plot) and mean power consumption I_mean (bottom plot) of the microcontroller together with the communication between reader and NFC-compatible tag (top plot).

the flexible tag platform is already quite low due to low-power design techniques like clock gating. Figure 10.10 shows the schematic of a clock-gating cell that reduces the toggle activity and hence the power consumption of the enclosed flip flop. However, the value of the overall power consumption can significantly be decreased by moving towards a more advanced CMOS process technology (*e.g.* 0.18 µm or 0.13 µm).

In order to demonstrate the effectiveness of our flexible tag platform, a small part of the control functionality is moved from the ROM code of the microcontroller to the control unit of the framing logic. The control functionality that is moved comprises processing of the commands: HLTA, RATS, and PPS. Additionally handling the three commands within the state machine of the control unit increases the chip area of the framing logic by 210 GEs. On the other hand, removing the 46 instructions from the ROM code that are responsible for handling the three commands, decreases the chip area of the microcontroller by 130 GEs. Consequently, the overall chip size is increased by 80 GEs. This small example further highlights that a microcontroller-based approach can better cope with complex control functionality than a conventional state machine.

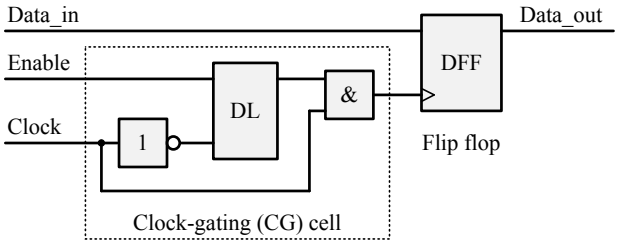


Figure 10.10: Schematic of a clock-gating cell to reduce toggle activity and power consumption of the enclosed flip flop.

10.6.3 Comparison with Related Work

Both chip area and power consumption of our flexible tag platform are within a range that make such an approach realistic for integration in low-cost RFID tags. Comparing our results with existing published work is difficult since authors often give only a vague description of their designs regarding implementation details (*e.g.* whether dedicated RAM or ROM macros are used) and provided functionality. Moreover, different target technologies are used that make a fair comparison even more difficult.

The work of Yan *et al.* [193] and Yu *et al.* [202] describe both the implementation of an EPC Generation 2 tag with an application-specific processor. The design of Yu *et al.* contains also a lightweight cryptographic unit and targets a 0.18 µm process. Overall size of the design is about 10 kGEs. However, none of the authors gives information concerning the resource usage of the control structure as well as the application-specific processor. Moreover, required control complexity of the utilized protocol is assumed to be much lower than in case of our NFC-compatible tag. Abrial *et al.* [5] present a contactless smart-card implementation that uses an asynchronous 8-bit microcontroller. No information about the utilized chip area is provided. Power consumption of the microcontroller core is rather high (several mA). The work of Piguet *et al.* [145] comes up with a microcontroller architecture that is comparable with the implementation in our design. They describe three microcontroller cores with different level

of functionality. The smallest microcontroller core with only one 8-bit register requires a chip area of 1 150 GEs (our microcontroller core without register file consumes about 950 GEs). Another 8-bit microcontroller with low-resource usage is introduce by Grünbacher *et al.* [67]. The microcontroller was intended for early smart-card designs and has been implemented on an FPGA with a 32x8-bit RAM macro (no program ROM), leading to a chip area around 4 300 GEs. Our microcontroller on the contrary only requires around 2 700 GEs without program ROM.

Comparison with published work shows that our flexible tag platform is highly competitive with respect to resource usage. This is emphasized by the fact that our design uses neither RAM nor ROM macros and thus still provides significant optimization potential.

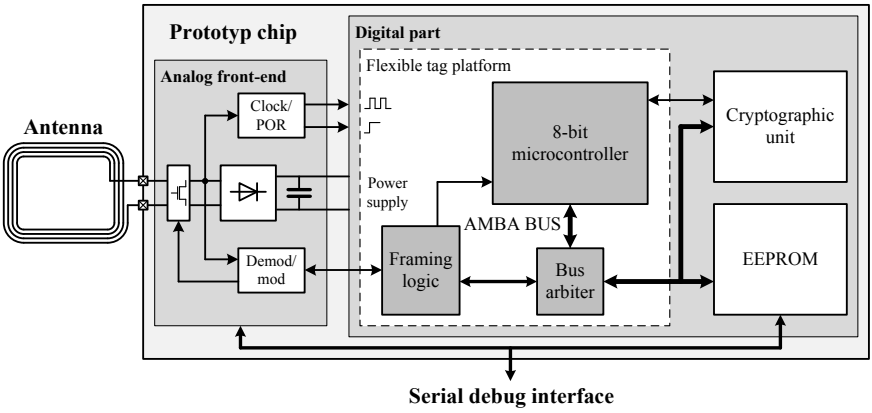


Figure 10.11: Schematic overview of the tag-prototype chip architecture.

10.7 Integration into the CRYPTA Tag-Prototype Chip

The flexible tag platform that we have previously described has been used within the CRYPTA project for implementing an RFID tag-prototype chip in silicon. The prototype chip has been manufactured on a multi-project wafer (MPW) using the 0.35 µm CMOS process technology C35b4 from Austriamicrosystems. The chip consists of an analog front-end (designed by Austriamicrosystems) and a digital part. Figure 10.11 gives a schematic overview of the tag-prototype chip's architecture. The digital part contains the flexible tag platform with the 8-bit microcontroller, the framing logic, and the bus arbiter, as well as the cryptographic unit and a 256 × 16-bit EEPROM. For ease of testability, a small serial debug interface has also been added that allows detailed analysis of analog front-end and EEPROM (*e.g.* reading/writing arbitrary values from/to EEPROM). The analog front-end extracts the power supply for the digital part

from the RF field and provides a power-on-reset (POR) signal as well as a clock signal. Moreover, the analog front-end is responsible for demodulating and modulating the data. The framing logic of the flexible tag platform that is used by the prototype chip has been designed by Austriamicrosystems and is similar to the one described in this work. A photo of the manufactured chip is shown in Figure 10.12. Most of the chip area is consumed by the standard-cell part that contains the flexible tag platform and the cryptographic unit (digital part excluding EEPROM). The standard-cell part has an area requirement of about 40 kGEs, 12.3 kGEs for the flexible tag platform (the deployed framing logic is larger than the one described in this chapter) and 27.7 kGES for the cryptographic unit.

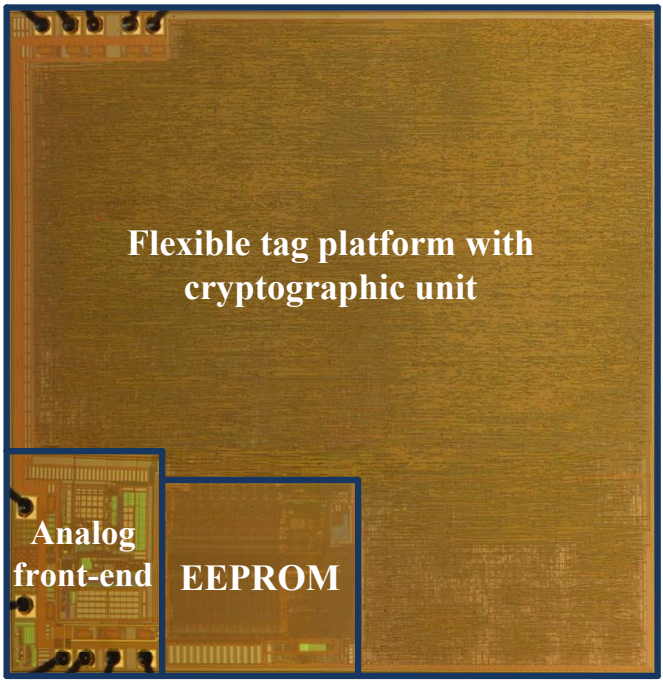


Figure 10.12: Photo of the RFID tag-prototype chip.

After production, the chip has been integrated into a ceramic package and soldered on a small printed circuit board (PCB) to allow tests with real-world RFID-reader devices. The PCB contains an antenna with 4 windings that is connected to the analog front-end of the chip. An adjustable capacitor is used for matching of antenna and analog front-end. The serial debug interface is accessible via a 10-pin connector. Figure 10.13 shows a photo of the PCB with the packaged chip. The tag operates fully passive. Extensive tests with different RFID-reader devices have pointed out that the prototype is working reliably. Two bugs that have been detected during the tests are planned to be addressed and resolved with a future metal change (*i.e.* the chip layout is changed by

modifying the wiring of the metal layers) to have a fully working prototype chip. When considering the complexity of the whole tag, having only two bugs is a very good achievement. This is mainly a result of the thorough simulations that have been conducted on various abstraction levels during the design phase.

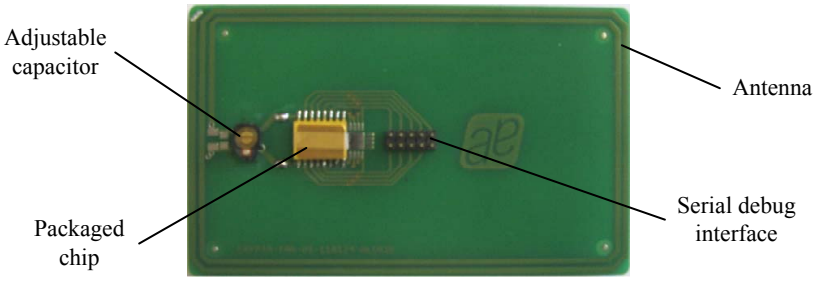


Figure 10.13: Photo of the PCB with the packaged chip.

10.8 Summary

In this chapter, we have presented a flexible platform for implementation of passive RFID tags that is optimized for low chip area and low power consumption. It allows efficiently handling complex control tasks which we demonstrate with a hardware design of an NFC-compatible tag that provides advanced file-access functionality and security features. The NFC-compatible tag has been manufactured as chip in silicon on a 0.35 µm CMOS process technology. Central component of the tag is a low-resource 8-bit microcontroller that consumes less than 10 µA at the target frequency of 106 kHz. For easier program development, a tool chain has been developed for the controller using Java (instruction-set simulator and assembler). The chip area of the control part of our flexible tag platform (*i.e.* without cryptographic unit and EEPROM) is about 10 kGEs when using standard cells. Further optimization of the design is possible when integrating special ROM or RAM macros. The results clearly point out that implementation of passive low-cost RFID tags is feasible using our flexible tag platform.

11

Implementation of Symmetric-Key Algorithms on a Low-Resource 8-Bit Microcontroller

During the last years, a lot of effort has been made by the research community to bring cryptographic security to RFID tags. The most prominent attempts among others are for example symmetric-key schemes like the Advanced Encryption Standard (AES) [55, 69], or public-key schemes like Elliptic Curve Cryptography (ECC) [20, 183]. All these attempts use dedicated hardware modules that are highly optimized for a specific cryptographic algorithm. Moreover, they do not consider the increased controlling effort that comes along with adding security to RFID tags.

When adding for example the security service tag authentication by using a simple challenge-response protocol, several tasks have to be accomplished by the control unit of a tag. First, the control unit of the tag needs to generate random data and combine it with the challenge from the RFID reader. Second, random data and challenge have to be provided to the cryptographic hardware module and processing of data has to be started. Finally, the processed data needs to be transferred to the RFID reader. Other security services like mutual authentication or secure key update are even more demanding in terms of controlling effort.

As we have shown in Chapter 10, using a flexible tag architecture based on a low-resource 8-bit microcontroller is advantageous to cope with increased controlling effort. The flexible tag architecture allows not only faster integration of new functionalities but also fulfills the fierce requirements of passive low-cost RFID tags in terms of power consumption and chip size. Moreover, when using

the microcontroller for control tasks, it seems reasonable to reuse it also for computing cryptographic algorithms. This reuse enables a better utilization of resources like the memory, which can help to reduce the chip size of the tag and in turn also to lower the costs.

In this chapter, we extend the 8-bit microcontroller used by the flexible tag platform (*e.g.* adapt instruction set, increase register file) and implement six symmetric-key algorithms on it. The symmetric-key algorithms are: the Advanced Encryption Standard (AES), NOEKEON, Present, the Scalable Encryption Algorithm (SEA), the Extended Tiny Encryption Algorithm (XTEA), and Trivium. The performance of our implementations is evaluated and compared with results from implementations on other dedicated microcontroller platforms. Finally, the hardware costs that are added due to the implementation of the cryptographic algorithms are compared with the costs of stand-alone hardware modules. The results clearly show that the implementations on our microcontroller have lower costs than the dedicated hardware modules. For example, AES encryption and decryption comes at cost of less than 3 000 GEs on our microcontroller.

Most of the information provided in this chapter has been published at the SAC conference 2010 [151] which is a joint work with Hannes Groß and Martin Feldhofer. The implementation results of NOEKEON have been published at the SecureComm conference 2011 [149] which is a joint work with Martin Feldhofer.

The remainder of this chapter is organized as follows. Section 11.1 describes the extensions that have been integrated into the 8-bit microcontroller. In Section 11.2, a short overview of the selected algorithms is given, followed by the implementation results of the algorithms in Section 11.3. Discussion of the results with respect to passive RFID tags is done in Section 11.4. A summary is given in Section 11.5.

11.1 Extension of the 8-Bit Microcontroller

We have extended the low-resource 8-bit microcontroller that is described in Section 10.4.2 to allow the implementation of more-complex programs and to obtain even higher flexibility. More precisely, we have increased the maximum number of supported registers in the register file from 32 to 64. The ROM has been divided into pages, where each page has a size of 512 bytes. Addressing a page is done via an additional program-counter register (PCH) that is located in the register file. Up to 256 pages can be addressed, resulting in a maximum ROM size of 128 kB. For our implementation we have only used 4 bits of the PCH register, limiting the maximum ROM size to 4 kB. An overview of the extended microcontroller architecture is shown in Figure 11.1. Increasing the number of registers and dividing the ROM into pages has required to adapt the instruction set. Further, we have integrated additional instructions that allow indirect addressing of registers (MovIFF) as well as indirect branching (GOTOI) and indirect subroutine calls (CALLI). Especially indirect subroutine calls are interesting for efficiently implementing look-up tables (which are used *e.g.*

by cryptographic algorithms such as the AES). The overall number of instructions has increased from 31 to 36. Table 11.2 presents a list with the adapted instruction set. New or adapted instructions in the table are highlighted in gray.

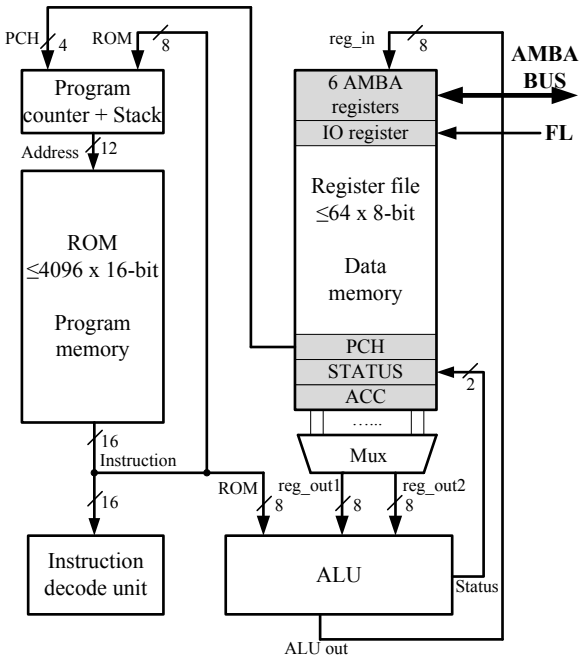


Figure 11.1: Overview of the extended microcontroller architecture.

Synthesizing the microcontroller for a 0.35 µm CMOS process technology leads to a chip area of 4 701 GEs, excluding the ROM. By far largest part is the register file with 3 677 GEs (for 64 x 8-bit). We have used latches for the general-purpose registers, which saves around 900 GEs compared to a pure flip-flop approach. Detailed synthesis results of all components of the microcontroller are given in Table 11.1. Power simulations with Synopsys Nanosim have shown that the power consumption of the microcontroller has not changed much and is still around 10 µA at the target frequency of 106 kHz.

When using the microcontroller for handling the control tasks of the tag, it seems reasonable to reuse it also for computing the cryptographic algorithms. In order to address this issue, different cryptographic algorithms have been implemented on the microcontroller. The following sections contain a short description of the selected cryptographic algorithms and evaluate their implementations with respect to execution time and code size.

Table 11.1: Synthesis results of the extended microcontroller with 64 x 8-bit register file excluding the ROM.

Component	**Chip area**	
	[GEs]	[%]
Program counter with call stack	465	9.9
ALU	257	5.5
Register file (64 x 8 bit)	3 677	78.2
Instruction decode unit	302	6.4
Total	**4 701**	**100.0**

11.2 Overview of the Selected Cryptographic Algorithms

Several cryptographic algorithms that could be interesting for RFID applications have been selected for implementation on the microcontroller. Main selection criteria are adequate security of the algorithm and moderate resource usage. We have selected five block ciphers and one stream cipher for evaluation. The block ciphers are: AES, NOEKEON, Present, SEA, and XTEA. The stream cipher is Trivium. Table 11.3 provides an overview of the selected algorithms and compares some of their properties like key size, block size, and number of rounds.

11.2.1 AES

The Advanced Encryption Standard (AES) is the successor of the Data Encryption Standard (DES) and was introduced by the National Institute of Standards and Technology (NIST) in 2001 [131]. AES uses a so-called substitution-permutation network (SPN) and works on a fixed block size of 128 bits. Three key lengths are supported: 128 bits, 192 bits, and 256 bits. In this work we only focus on the 128-bit key version (AES-128). The internal state of AES is organized as a matrix of 4×4 bytes. Within each round of AES, four operations are applied on the internal state: a byte substitution with an 8-bit S-box, a shift-row transformation, a mix-columns transformation, and an XOR with the actual round key. AES-128 uses 10 rounds. For deducing the rounds keys from the main key, a simple key schedule is applied that mainly uses some XOR operations and the 8-bit S-box. Decrypting data requires inversion of the byte substitution, the shift-row transformation, the mix-columns transformation, and the key schedule. Thus, adding decryption functionality significantly increases the size of an AES implementation. AES is considered as very secure since the algorithm is widely deployed and well researched by the cryptographic community. In 2000, Ferguson *et al.* [58] have published an attack on the 128-bit key version that applies on 7 out of 10 rounds. A key-distinguisher attack on 8 rounds has been presented by Gilbert *et al.* [65] in 2009. Recently, Bogdanov *et al.* [24] have published an attack on the full version of AES-128 that is slightly faster

Table 11.2: Overview of the adapted instruction set used by the low-resource 8-bit microcontroller. Each instruction is listed with its type, the name, the number of cycles, and a short description (new or adapted instructions are highlighted in gray).

Type	Name	Cycles	Description
Constant	MovLF	1	Move constant to register
	XorLF	1	Exclusive OR register and constant
Branch	GOTO	2	Unconditional branch
	CALL	2	Subroutine call
	GOTOR	2	Relative unconditional branch
	CALLR	2	Relative subroutine call
	GOTOI	2	Indirect unconditional branch
	CALLI	2	Indirect subroutine call
	BZ	1/2	Branch if zero
	BNZ	1/2	Branch if not zero
Register to register	MovFF	1	Move register to register
	MovIFF	1	Indirect move register to register
	AndFF	1	Logical AND registers
	XorFF	1	Exclusive OR registers
	AddFF	1	Add two registers
Conditional branch	BTC	1/2	Skip next instruction if bit is cleared
	BTS	1/2	Skip next instruction if bit is set
	BWC	1	Wait until bit is cleared
	BWS	1	Wait until bit is set
Work register	AndLW	1	Logical AND ACC and constant
	OrLW	1	Logical OR ACC and constant
	AddLW	1	Add constant to ACC
	SubLW	1	Subtract constant from ACC
	RetLW	2	Move constant to ACC and leave subroutine
Register to target	BC	1	Clear selected bit of register
	BS	1	Set selected bit of register
	RotlWF	1	Rotate left through carry
	RotrWF	1	Rotate right through carry
	ShlWF	1	Shift left through carry
	ShrWF	1	Shift right through carry
	DecWF	1	Decrement register
	IncWF	1	Increment register
	DecTWF	1/2	Decrement register and branch if zero
	IncTWF	1/2	Increment register and branch if zero
Others	RET	2	Return from subroutine
	NOP	1	No operation

than brute force. However, as underlined by the authors, this attack is only of theoretic interest and does not compromise the security of AES in practise.

Table 11.3: Comparison of the selected cryptographic algorithms.

Algorithm	Key size [bits]	Block size [bits]	Number of rounds per block
Block ciphers			
AES	128	128	10
NOEKEON	128	128	16
Present	80	64	31
SEA	96	96	93
XTEA	128	64	64
Stream ciphers			/128 bits
Trivium	80	-	128

11.2.2 NOEKEON

NOEKEON is a symmetric-key block cipher and was designed by Joan Daemen, Michaël Peeters, Gilles Van Assche, and Vincent Rijmen in 2000 [42]. The cipher is based on a substitution-linear transformation network where both block size and key size are fixed to 128 bits. Similar to AES, NOEKEON consists of a simple round function that is repeatedly applied. The round function is iterated 16 times and consists of five operations: XOR with a round constant, Theta, Pi1, Gamma, and Pi2. All operations only rely on simple bit-wise Boolean operations and cyclic shifts. The operation Theta also involves an XOR with the working key, which is either the initial cipher key itself (direct mode) or the cipher key deduced from using the key schedule (indirect mode). The key schedule in indirect mode applies the NOEKEON cipher itself on the cipher key with a null string as working key. The authors of the cipher recommend using the key schedule since it increases the resistance against related-key attacks. Due to the self-inverse structure of NOEKEON, implementing the decryption operation causes only little overhead in terms of code size and chip area. The best-known attack on NOEKEON applies on 5 out of 16 rounds and was published by Z'aba *et al.* [203] in 2008. Related-key attacks on both modes the direct one and the indirect one have been presented by Knudsen *et al.* [102] in 2001.

11.2.3 Present

Present is another lightweight block cipher suitable for implementation on constrained devices. It uses a substitution-permutation network (SPN) like AES and was introduced by Bogdanov *et al.* in 2007 [25]. Present operates on 64-bit data blocks and supports key lengths of 80 bits (Present-80) and 128 bits (Present-128). In our implementations we have selected Present-80 since it is most interesting for low-resource implementations. The round function of Present-80 for encrypting data consists of three basic operations: XOR with the round key, a substitution layer using a 4-bit S-box, and a bit-permutation layer. The round keys are derived from a key schedule that uses: bit rotation, application of the

4-bit S-box, and XOR with the actual round counter. For decrypting data, the round function uses a substitution layer with an inverse S-box and an inverse bit-permutation layer. Also the key-update function needs to be inverted by applying inverse bit rotation and application of the inverse S-box. Both encryption and decryption use 31 rounds. The best known attack on Present-80 applies on 26 out of the 31 rounds and has been published by Cho in 2010 [35].

11.2.4 SEA

SEA stands for Scalable Encryption Algorithm and was developed by Standaert *et al.* in 2006 [174]. The algorithm is a so-called Feistel block cipher and was designed for resource-constrained devices such as microcontrollers that have only a limited instruction set and little memory available. SEA uses a round function that is iteratively applied on its internal state together with a round key. For each round, a new round key is derived via a key schedule. Both round function and key schedule have a simple structure that consists of five operations: bitwise XOR, a 3-bit substitution box, word rotation, bit rotation, and modular addition. The round functions for encryption and decryption are quite similar. They only differ in the position and the direction of the word rotation operation inside the round function. As the name implies, SEA is highly scalable. This means that parameters like the word size b, the width n of the key and the plaintext, and the number of rounds n_r can be adjusted for the requirements of the target device. For an 8-bit microcontroller (word size $b = 8$) for example, the designers of the algorithm suggest $n = 96$ bits and $n_r = 93$ rounds. The large number of rounds is a consequence of the simple design of the cipher. SEA is a rather new algorithm and not as well researched as other block ciphers. Hence, further analysis of its security is still necessary.

11.2.5 XTEA

The Extended Tiny Encryption Algorithm (XTEA) was published in 1997 [133] and is the successor of the Tiny Encryption Algorithm (TEA). XTEA is a 64-bit block cipher with a Feistel structure that iterates a simple round function over a number of 64 rounds. The round function consists of bit-shift operations, XOR operations, and additions modulo 2^{32}. The key used by XTEA has a length of 128 bits and is divided into four sub keys with a length of 32 bits each. The key schedule is also quite simple. By applying a round-dependent selection function, one of the four sub keys is selected in each round and used as a round key. Encryption and decryption operation of XTEA have a similar structure allowing a rather compact implementation of the block cipher. The best known attack on XTEA is a so-called related-key rectangle attack that addresses 36 rounds out of the 64 rounds [111].

11.2.6 Trivium

Trivium is a hardware-oriented stream cipher developed by De Cannière *et al.* [30]. Although the stream cipher is optimized for hardware designs, it provides also low-resource usage in software implementations. Trivium follows a very simple design strategy and allows to generate up to 2^{64} bits of key stream from an 80-bit initial value (IV) and an 80-bit secret key. Trivium operates on a 288-bit internal state and iteratively applies an update function to generate the key stream. The update function extracts 15 bits from the internal state and produces 3 new state bits as well as one key-stream bit by using a combination of AND and XOR operations. After updating the 3 state bits the whole state is rotated by one position. Before the key-stream generation is started, initialization of the internal state is required. This is achieved by loading the 80-bit key and the 80-bit IV into the internal state and by applying the update function 1 152 times without generating any key-stream bits. There exist several attacks on reduced variants of Trivium as described in [46] and [184]. However, there is no successful attack against the full version of Trivium.

11.3 Implementation Results

This section presents the implementation results of the cryptographic algorithms on our microcontroller. All implementations have been done in our own assembler environment and optimized towards three targets: execution time, code size, and efficiency. In order to determine the efficiency of an implementation, a scaling factor $s = 10^8$ is divided by the product of execution time in clock cycles and code size in bytes (s is used to obtain easier-manageable values). As mentioned in the previous section, we have selected five block ciphers and one stream cipher for implementation. For the block ciphers both encryption and decryption are implemented. When the block ciphers need to compute round keys this is done on-the-fly during the encryption or the decryption routine. Moreover, we compare our results with implementations on other platforms like AVR microcontrollers from Atmel, PIC microcontrollers from Microchip, 68HC08 microcontrollers from Motorola, or 8051 microcontrollers from various manufacturers. The latter are based on a so-called Complex Instruction-Set Computer (CISC) architecture where the execution of a single instruction (a machine cycle) typically requires several clock cycles. This makes a comparison with 8051 microcontrollers difficult, since depending on the manufacturer, the number of clock cycles per instruction can vary. An overview of our implementation results is given in Table 11.4, which contains also results from implementations on the other microcontroller platforms.

11.3.1 AES

We have selected AES with a key length of 128 bits for implementation on our microcontroller. Encryption and decryption operations of AES are realized by implementing the round function once and then iteratively applying it on the

internal state. Round keys are computed on-the-fly through a key-update function. Since the decryption operation of AES starts with the last round key, an additional pre-processing step is required where the key-update function is applied ten times. Hence, decrypting data takes generally longer than encrypting data with this approach. S-box operation and inverse S-box operation are realized as look-up tables with 256 entries each. The AES implementation with the best efficiency requires only 3 304 clock cycles for encryption and only 5 037 clock cycles for decryption (both values already include the key schedule). Code size of this version is 1 940 bytes. By extensively using function calls instead of code duplication, for example by implementing the mix-columns operation only once and then applying it successively on all four columns, more than 200 bytes of code can be saved. However, this comes at the prize of a significantly longer execution time. Such a compact version needs 5 064 clock cycles for encryption and 8 226 clock cycles for decryption. Speeding up the implementation of AES goes into the opposite direction, code duplication is used to avoid functions calls. Our speed-optimized version of AES requires only 3 084 clock cycles for encrypting a data block and 4 505 clock cycles for decrypting a data block. This moderate speed up causes the code-size to increase to 2 158 bytes. Regardless of the optimization target, 39 registers are used by our implementations: 16 registers for the state, 16 registers for the round key, and 7 registers for temporary computations.

The AES algorithm is widely deployed and many implementations for various microcontroller platforms are available. In Table 11.4 we list some of them, and compare them with our results. We also added two encryption-only versions of our implementations, one optimized for speed and one optimized for code size, to provide better comparability with related work where the decryption operation is omitted. When comparing with implementations on AVR or PIC microcontrollers, our versions are not only faster but also more compact in code size, leading to a much better efficiency. At first glance, the situation looks different for our encryption-only versions. There, the AES implementations on the 8051 microcontrollers seem to provide better efficiency. However, it has to be noted that the performance numbers of the 8051 microcontrollers are related to machine cycles (a machine cycle requires typically several clock cycles).

11.3.2 NOEKEON

NOEKEON requires only bit-wise Boolean operations and cyclic shifts which can be implemented with compact code size. No large look-up tables are required as in case of AES. We are using NOEKEON in indirect mode that applies an additional key schedule to increase resistance against related-key attacks. The key schedule in indirect mode can be precomputed, since the operation is independent of the processed data and all rounds use the same key. Hence, a lot of computation time can be saved when storing the precomputed working key in the EEPROM instead of the original cipher key. Depending on the optimization target, the number of utilized registers differs. The speed-optimized version has the highest resource usage because it loads the working key into the register file,

Table 11.4: Implementation results of the cryptographic algorithms and comparison with related work.

Algorithm	Platform	Target	Code size	Encryption		Decryption	
			[bytes]	clock cycles	effi-ciency	clock cycles	effi-ciency
Block ciphers							
AES	**This work**	**size**	**1 704**	**5 064**	**11.6**	**8 226**	**7.1**
	This work	**eff.**	**1 940**	**3 304**	**15.6**	**5 037**	**10.2**
	This work	**speed**	**2 158**	**3 084**	**15.0**	**4 505**	**10.3**
	AVR [161]	-	3 410	3 766	7.8	4 558	6.4
	AVR [49]	-	2 606	6 637	5.8	7 429	5.2
	PIC [123]	-	2 478	5 273	7.7	7 041	5.7
AES (encr. only)	**This work**	**size**	**918**	**4 192**	**26.0**	**-**	-
	This work	**speed**	**1 110**	**3 004**	**30.0**	**-**	-
	8051 [43]	-	1 016	3 168[a]	31.1[a]	-	-
	8051 [43]	-	826	3 744[a]	32.3[a]	-	-
	8051 [43]	-	768	4 065[a]	32.0[a]	-	-
	68HC98 [43]	-	919	8 390	13.0	-	-
NOEKEON	**This work**	**size**	**414**	**7 563**	**31.9**	**7 546**	**32.0**
	This work	**eff.**	**532**	**5 839**	**32.2**	**5 824**	**32.3**
	This work	**speed**	**980**	**3 817**	**26.7**	**3 785**	**27.0**
	AVR [1]	-	774	10 416	12.4	10 191	12.7
Present	**This work**	**size**	**920**	**28 062**	**3.9**	**60 427**	**1.8**
	This work	**eff.**	**1 148**	**15 042**	**5.8**	**17 677**	**4.9**
	This work	**speed**	**2 146**	**8 958**	**5.2**	**11 592**	**4.0**
	AVR [156]	-	2 398	9 595	4.3	9 820	4.2
	AVR [156]	-	1 474	646 166	0.1	634 614	0.1
	AVR [49]	-	936	10 723	10.0	11 239	9.5
SEA	**This work**	**size**	**332**	**14 723**	**20.5**	**14 723**	**20.5**
	This work	**eff.**	**488**	**8 597**	**23.8**	**8 597**	**23.8**
	This work	**speed**	**786**	**8 053**	**15.8**	**8 053**	**15.8**
	AVR [161]	-	2 132	9 654	4.9	9 654	4.9
	AVR [174]	-	386	17 745	14.6	17 745	14.6
	AVR [47]	-	834	9 658	12.4	9 658	12.4
	8051 [47]	-	604	8 250[a]	20.1[a]	8 250[a]	20.1[a]
XTEA	**This work**	**size**	**504**	**17 514**	**11.3**	**19 936**	**10.0**
	This work	**eff.**	**820**	**7 786**	**15.7**	**8 928**	**13.7**
	This work	**speed**	**1 246**	**7 595**	**10.6**	**8 735**	**9.2**
	AVR [161]	-	1 160	6 718	12.8	6 718	12.8
	8051 [118]	-	542	6 954[a]	26.5[a]	7 053[a]	26.2[a]
	PIC [124]	-	962	7 408	14.0	7 408	14.0
Stream ciphers				Initialization		/128 bits	
Trivium	**This work**	**size**	**332**	**85 697**	**3.5**	**9 488**	**31.7**
	This work	**eff.**	**726**	**40 337**	**3.4**	**4 448**	**31.0**
	This work	**speed**	**1 226**	**39 833**	**2.0**	**4 112**	**19.8**
	AVR [1]	-	424	775 726	0.3	85 120	2.8

[a]The values for the 8051 microcontrollers refer to machine cycles.

resulting in 35 registers: 16 registers for the state, 16 register for the working key, and 3 registers for temporary computations. Keeping the whole working key in EEPROM and loading it only piece-by-piece during each round saves a lot of registers. The most-efficient version loads the working key in blocks of 4 bytes from the EEPROM, leading to 25 registers: 16 registers for the state, 6 registers for loading the key, 3 registers for temporary computations. In the code-size optimized version, the working key is loaded in blocks of 2 bytes from the EEPROM, saving another 2 registers. Encrypting one block of data with NOEKEON requires 3 817 clock cycles with the speed-optimized version, and 7 563 clock cycles with code-size optimized version. Decrypting data can be done with quite the same speed. Code size for combined implementation of encryption function and decryption function ranges from 414 bytes to 980 bytes.

A summary of the implementation results of NOEKEON is shown in Table 11.4. We also compare our results with an implementation on an Atmel AVR microcontroller. The comparison clearly points out that all our implementations have a shorter execution time and are much more efficient. Code size of our smallest and most-efficient versions is also lower than in case of the AVR implementation.

11.3.3 Present

Both encryption and decryption operation of Present have been implemented using a key length of 80 bits (Present-80). Round keys are computed on-the-fly. As in case of AES, decryption operation requires significantly longer than encryption operation since the last round key needs to be computed at the beginning (*i.e.* update round-key function is applied 31 times). For the most-compact implementation of Present, as much functionality as possible of the encryption and the decryption routine is shared to minimize code size. Two look-up tables with 16 entries each are used, one for the S-box operation and one for the inverse S-box operation. The bit rotation during the key schedule is performed bit wise in a loop. This results in a code size of 920 bytes, allowing encryption of data within 28 062 clock cycles, and decryption of data within 60 427 clock cycles. A more efficient implementation uses two additional look-up tables with 16 entries each. This allows to efficiently apply the S-box operation and the inverse S-box operation on the lower four bits and on the upper four bits. Bit rotation during key schedule is not only done bit wise in a loop, but in combination with swapping on byte basis. This implementation requires 15 042 clock cycles for encryption and 17 677 clock cycles for decryption. Code size increases to 1 148 bytes. The fastest version of Present uses two big look-up tables with 256 entries each that perform the S-box operation on a whole byte. Most of the function calls are replaced by code duplication. In that way, execution time reduces to 8 958 clock cycles for encryption and 11 592 clock cycles for decryption. Code size significantly increases to 2 146 bytes. All our versions of Present require a total number of 30 registers: 8 register for the internal state, 10 registers for the round key, and 12 registers for storing intermediate results.

An overview of the implementation results of Present is provided in Ta-

ble 11.4. A comparison with implementation results on Atmel AVR devices shows that our speed-optimized version allows encryption within less clock cycles and that our code-size optimized version is even more compact. However, the implementation of [49] provides better efficiency.

11.3.4 SEA

The simple structure of SEA allows a rather straight-forward implementation on the microcontroller. As suggested in [174] for 8-bit platforms, we have selected a key/plaintext length of 96 bits and performed 93 rounds for encryption and decryption of data ($SEA_{96,8}$). Encryption and decryption operation of SEA are quite similar and can efficiently be combined in a single function. Depending on a status bit stored in a register, encryption or decryption-specific routines can be selected during operation. In that way, a very compact implementation of SEA is obtained that requires only 332 bytes of code. Encryption or decryption of a 96-bit data block lasts 14 723 clock cycles each with this version. Due to the Feistel structure of SEA, the left half of the internal state is swapped with the right half after each round. The same applies to the round key after each round. These swap operations are time consuming and can be circumvented by using separate round functions for the left half and the right half of the internal state as well as separate key-update functions. With this optimization a significant speed up is obtained by only moderately increasing the code size. An encryption or decryption operation requires only 8 597 clock cycles. Code size increases to 488 bytes. A further speed up is obtained by using one dedicated function for encryption and one for decryption. This brings only a minor speed up of about 500 clock cycles while it increases the code size by nearly 300 bytes. Our implementations of SEA require 12 registers for storing the internal state, 12 registers for storing the round key, and another 5 registers for temporary computations. This gives an overall count of 29 registers. The two versions with combined encryption and decryption routine require an additional register for the status bit that indicates whether encryption or decryption is performed.

Table 11.4 gives an overview of the results and compares them with implementations on other microcontroller platforms. Our implementations have a good efficiency leading to a small code-size clock-cycle product. The efficiency is even better than on the 8051 microcontroller, whose execution time is indicated in machine cycles (a machine cycle typically requires several clock cycles).

11.3.5 XTEA

XTEA has a Feistel structure just like SEA. Thus, similar optimization strategies can be applied. Again, we implemented both encryption and decryption operation of the cipher. The most-compact version with respect to code size has been achieved by implementing only one round of the algorithm and by reusing as much code as possible for encryption and decryption. Moreover, bit-shift operations are performed bit wise in a loop. This allows implementing XTEA with only 504 bytes of code. Encrypting data with this version requires 17 514

clock cycles per block, decrypting data requires 19 936 clock cycles per block. The execution time of the cipher can be reduced in a first step by implementing two rounds of XTEA. Hence, swapping the two halves of the internal state after each round is done implicitly. Together with an optimization of the bit-shift operations, the execution time is reduced to 7 786 clock cycles for encryption and 8 928 clock cycles for decryption. Code size is nearly doubled and increases to 820 bytes. The fastest version of XTEA uses code duplication and provides only a minor speed up of about 200 clock cycles, while spending more than 400 bytes of additional code. The register usage of the XTEA implementations is between 23 registers for the most-compact version, and 27 registers for the fastest version. This low values result from the simple structure of the key schedule.

Table 11.4 compares our results of XTEA with implementations on other 8-bit microcontroller platforms. The implementations on the AVR microcontroller and on the PIC microcontroller are a bit faster than our speed-optimized version. This is a consequence of the limited instruction set of our microcontroller, which prevents from efficiently adding 32-bit words. Nevertheless, our microcontroller achieves compact code size with slightly better efficiency for encryption.

11.3.6 Trivium

Trivium is the last algorithm that has been implemented on the microcontroller. Although Trivium is a hardware-oriented design, it can be implemented in a very compact way in software. The code-size optimized version of Trivium uses a round function that generates one key-stream bit per iteration and leads to the smallest code size by using only 330 bytes. The small size comes at cost of execution time, resulting in 85 697 clock cycles for initialization and 9 488 clock cycles for generating 128 bits of key stream. The efficiency-optimized version generates 8 key-stream bits per iteration. This noticeably reduces execution time by doubling the code size. The speed-optimized version of Trivium generates 16 key-stream bits per iteration. Such an approach only slightly improves execution time by significantly increasing code size. All our versions of Trivium require 39 registers: 36 for the internal state and three for temporary computations and the key-stream bits.

The results of Trivium are listed in Table 11.4 together with an implementation on an AVR microcontroller. The implementation of Trivium on the AVR microcontroller is done in C, leading to poor performance values compared to our versions.

11.3.7 Summary of Implementation Results

The results above clarify that our microcontroller allows implementing the selected cryptographic algorithms in a very compact and efficient way. Our implementations of AES, NOEKEON, Present, SEA, and Trivium are faster than on the other compared 8-bit microcontroller platforms. Except in the case of AES, our implementations are also the most compact ones. There are several reasons for the good performance numbers of our implementations. First, our

microcontroller can use op to 64 8-bit registers, which is more than for example in the case of AVR or 68HC98 microcontrollers. Second, our microcontroller provides also instructions that operate on two input registers at once (*e.g.* MOVFF or XORFF) and most instructions execute within a single clock cycles (only control-flow operations require two clock cycles).

Comparing the performance numbers of the cryptographic algorithms shows that AES has the shortest execution time, but also requires most code size. When looking at code size, SEA and Trivium are the algorithms that can be implemented with a minimum number of bytes. However, the initialization phase of Trivium takes exceptionally long. All these results are used in the next section to give actual values for the hardware costs that arise from implementing the algorithms on the microcontroller.

11.4 Discussing the Costs of Integrating the Implemented Algorithms on Passive RFID Tags

Two important constraints need to be considered when implementing cryptographic algorithms on passive RFID tags: power consumption and chip size. Power consumption affects the read range of the tag while chip size affects the costs of the tag. First, the power consumption of our microcontroller is more or less independent of the number of instructions present in the synthesized ROM. Thus, increasing the number of instructions by implementing cryptographic algorithms will not increase the power consumption. Second, the chip size of our microcontroller is not fixed, rather it is mainly defined by the size of the register file and by the size of the synthesized ROM. Depending on the application, the register file can contain up to 64 8-bit registers. When the microcontroller uses these registers already for handling the control tasks, no additional hardware costs are introduced when reusing them for computing the cryptographic algorithms. Hence, the resulting chip size of the microcontroller is mainly influenced by the code size when implementing cryptographic algorithms. For this reason, we only consider the code size of the implemented algorithms as cost factor. Less attention is drawn on the execution speed of the algorithms, since RFID tags typically have enough time for the computations and only need to handle little data.

Actual values for chip-size increase have been determined by implementing the cryptographic algorithms on our microcontroller platform. These values are obtained by synthesizing the program code of the algorithm implementations described above. Synthesis has been done for a 0.35 µm CMOS process technology using a semi-custom design flow with Cadence RTL Compiler. Table 11.5 presents the synthesis results, by bringing code size of each implementation in relation with chip area. Code size is given in terms of bytes and chip area is given in terms of gate equivalents (GEs). The chip area of the implementations ranges from 745 GEs for the code-size optimized version of Trivium to 3 273 GEs

for the speed-optimized version of AES.

Looking at the synthesis results brings up an interesting observation that concerns the area efficiency of the implemented algorithms. The area efficiency in terms of bits per GE is not constant but strongly varies and mainly depends on two factors. First, the area efficiency is improved when the code size of an implementation increases. For example, the speed-optimized version of AES with 2 158 bytes of code has an area efficiency of 5.3 bits/GE, but the code-size optimized version of Trivium with 332 bytes of code has only 3.6 bits/GE. This varying area efficiency is caused by the synthesis tool, which can better optimize larger look-up tables. However, it is not intended that the algorithm implementations are used on their own, but together with the implementation of the communication protocol. This leads to a larger overall code size, which finally improves the area efficiency. Second, implementations with a lot of redundancy in the code reach even a much better area efficiency. An example for such an implementation is the speed-optimized version of Present which reaches 8.0 bits/GE. In this version, the 4-bit S-box is replicated 16 times to achieve faster execution of the algorithm (code duplication). Although this replication significantly increases code size, the chip area is only moderately increased since the synthesis tool removes redundancies in the resulting look-up table.

Table 11.5 gives not only an overview of the synthesis results, but also compares them with the area requirements of stand-alone hardware modules. In almost all cases, the hardware modules require more chip area than the implementations on our microcontroller (only code size is treated as cost factor since the register file is reused). Even the speed-optimized versions, which have the highest area requirements are smaller. The hardware implementation of Present is the only exception. It consumes about 300 GEs less than the most-compact version on the microcontroller. Looking at the results of AES shows that an implementation on the microcontroller supporting encryption and decryption can be realized within less than 3 000 GEs. This allows to save around 400 GEs compared to the smallest AES stand-alone hardware module. The same applies for the AES encryption-only version, where our implementations are about 500 GEs smaller than the most-compact hardware module.

Particularly for AES there exist several other approaches that try to minimize the costs of implementing the algorithm on a microcontroller. For example, microcontrollers with AES-specific design like the AESMPU [34] or microcontrollers with instruction-set extensions (ISE) [180]. Although both examples only need about half the code size of our AES implementations they are less flexible. The lack of flexibility comes from the AES-specific hardware parts that are used by both approaches. These parts need to be removed (redesign of the microcontroller on HDL level necessary) when implementing other cryptographic algorithms. Otherwise, the AES-specific parts will unnecessarily increase the chip size of the microcontroller. Moreover, the AESMPU is not designed for low-resource usage since it precomputes all round keys and stores them in its internal memory (176 8-bit registers are required). This enormous memory usage makes the AESMPU inapplicable for passive RFID tags. ISE are more attractive

Table 11.5: Synthesis results of the algorithm implementations on the microcontroller and comparison with dedicated hardware modules.

Algorithm	Platform	Target	Code size [bytes]	Area [GEs]	Area efficiency [bits/GE]
Block ciphers					
AES	**This work**	**size**	**1 704**	**2 911**	**4.7**
	This work	**efficiency**	**1 940**	**3 130**	**5.0**
	This work	**speed**	**2 158**	**3 273**	**5.3**
	Feldhofer [57]	-	-	3 400	-
AES (encr. only)	**This work**	**size**	**918**	**1 755**	**4.2**
	This work	**speed**	**1 110**	**1 871**	**4.7**
	Hämäläinen [69]	-	-	3 100	-
	Moradi [127]	-	-	2 400	-
NOEKEON	**This work**	**size**	**414**	**976**	**3.4**
	This work	**efficiency**	**532**	**1 127**	**3.8**
	This work	**speed**	**980**	**1 698**	**4.6**
	Bertoni [21]	-	-	$\approx$ 3 800[b]	-
Present	**This work**	**size**	**920**	**1 399**	**5.3**
	This work	**efficiency**	**1 148**	**1 763**	**5.2**
	This work	**speed**	**2 146**	**2 139**	**8.0**
	Poschmann [156]	-	-	1 075	-
SEA	**This work**	**size**	**332**	**786**	**3.4**
	This work	**efficiency**	**488**	**1 083**	**3.6**
	This work	**speed**	**786**	**1 619**	**3.9**
	Mace [112]	-	-	3 758	-
XTEA	**This work**	**size**	**504**	**1 230**	**3.3**
	This work	**efficiency**	**820**	**1 718**	**3.8**
	This work	**speed**	**1 246**	**2 507**	**4.0**
	Feldhofer [56]	-	-	2 636	-
Stream ciphers					
Trivium	**This work**	**size**	**332**	**745**	**3.6**
	This work	**efficiency**	**726**	**1 476**	**3.9**
	This work	**speed**	**1 226**	**2 228**	**4.4**
	Feldhofer [56]	-	-	2 390	-

[b] We have estimated the area requirement of the NOEKEON stand-alone hardware implementation according to the information given in [21].

than the AESMPU in terms of resource usage.

The ISE consume 791 GEs for the AES-specific hardware parts. Code size for implementing encryption and decryption via ISE is 840 bytes for the size-optimized version and 1 708 bytes for the speed-optimized version. When assuming an area efficiency of 3.7 bits/GE for 840 bytes and 4.7 bits/GE for 1 708 bytes, the ISE will end up with roughly 2 607 and 3 698 GEs, respectively. Hence, the speed-optimized version of the ISE approach is more than 400 GEs larger and the size-optimized version about 300 GEs smaller. This potential decrease of

chip area comes at cost of less flexibility. Moreover, the AES-specific hardware parts will slightly increase the overall power consumption of the microcontroller. Nevertheless, ISE are much faster. They allow to encrypt or decrypt a block of data within less than 1 500 clock cycles. Thus, when the execution time is an important factor for an application, using ISE is beneficial. Table 11.6 compares the performance numbers of our AES implementations with ISE. Due to the flexibility of our synthesizable microcontroller, it should not be too much effort to integrate also ISE if required in order to achieve an additional speed up.

Table 11.6: Comparison of our AES implementations with ISE.

Platform	**Encryption** clock cycles	**Decryption** clock cycles	**Code size** [bytes]	**Additional hardware** [GEs]	**Total area** [GEs]
This work (size)	**5 064**	**8 226**	**1 704**	**-**	**2 911**
This work (efficiency)	**3 304**	**5 037**	**1 940**	**-**	**3 130**
This work (speed)	**3 084**	**4 505**	**2 158**	**-**	**3 273**
ISE (size) Tillich [180]	1 442	1 443[a]	840	791	2 607
ISE (speed) Tillich [180]	1 259	1 259[a]	1 708	791	3 698

[a]Last round key is precomputed and directly supplied to the decryption function.

The results of our algorithm implementations let us come to two important conclusions. First, our microcontroller platform that is mainly intended for simple control tasks on RFID tags, allows also to efficiently implement cryptographic algorithms like AES, Present, or XTEA. Second, the additional hardware costs that are introduced by implementing cryptographic algorithms on our synthesizable microcontroller are in almost all cases lower (except in case of Present) than by using stand-alone hardware modules. The additional hardware costs are only affected by the code size of the algorithm. The data memory in the register file is already used by the microcontroller for handling control tasks and will not result in additional hardware costs. Note that this statement is only correct in an environment where the microcontroller is used anyway and the question how much does security cost arises. This makes our synthesizable microcontroller a resource saving and flexible concept to bring cryptographic security to passive RFID tags.

11.5 Summary

In this chapter, we have shown a very efficient concept of reusing a dedicated 8-bit microcontroller for the implementation of symmetric-key algorithms. The microcontroller, which is highly optimized for controlling tasks like protocol execution, is synthesizable and optimized concerning low chip area and low power consumption. It is also flexible concerning the program-memory size and the number of used registers. We evaluated the block ciphers AES, NOEKEON, Present, SEA, and XTEA as well as the stream cipher Trivium with respect to program size and required number of clock cycles. Our findings clearly show that

the implemented microcontroller is more efficient than other dedicated microcontrollers and outperforms even optimized hardware modules when considering the reuse of the microcontroller for protocol execution tasks.

12

Combined Implementation of Protocol Handling and Cryptographic Algorithm on a Low-Resource 8-Bit Microcontroller

The results in Chapter 10 and Chapter 11 let us come to two important conclusions. First, using a flexible tag platform based on a low-resource 8-bit microcontroller is advantageous for handling the protocol of RFID tags with advanced functionality. Second, the low-resource microcontroller used by the flexible tag platform is also suitable for efficiently implementing symmetric-key algorithms such as AES, NOEKEON, or SEA. Consequently, integrating both on the microcontroller seems to be a very promising approach for efficiently integrating security into low-cost RFID tags.

In this chapter we analyze the benefits of having a combined implementation of protocol handling and cryptographic algorithm on a low-resource microcontroller. We demonstrate this by using the flexible tag platform introduced in Chapter 10 which we adapt for our needs. Three different security-layer variants are implemented for evaluating the hardware costs introduced by them. The security-layer variants base on the cryptographic algorithms AES and NOEKEON, respectively. In contrast to related work, not only the costs of the cryptographic-algorithm implementation alone are considered, but also the costs that arise from protocol handling of the security layer. Our results underline that protocol handling constitutes a significant cost factor and must not be neglected. Depending on the security-layer variant and the utilized cryptographic algorithm, up to 66 % of the total overhead costs originate from protocol handling. Most of the information presented in this chapter have been published at the SecureComm conference 2011 [149] which is a joint work with

Martin Feldhofer.

The remainder of this chapter is structured as follows. In Section 12.1 we present a system overview of our low-cost tag. Section 12.2 gives details about the deployed security-layer variants and Section 12.3 describes the concept for realizing them on the tag. Implementation results are provided in Section 12.4. A summary in Section 12.5 finalizes the chapter.

12.1 System Overview

This section gives first a short overview of the architecture of our tag's digital part. The architecture bases mainly on the flexible tag platform presented in Chapter 10, which has been adapted for our needs. We have replaced the microcontroller with the enhanced version described in Chapter 11 that supports up to 64 8-bit registers and that allows implementation of larger programs. The cryptographic unit has been removed and a small true-random number generator (TRNG) has been added instead. EEPROM, framing logic, and bus arbiter have mainly remained unchanged. A schematic overview of the tag's digital part is given in Figure 12.1.

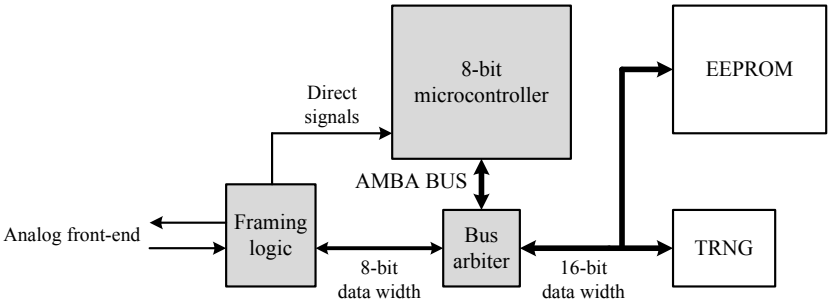


Figure 12.1: Architectural overview of the tag's digital part.

The high-level protocol functionality of the tag comprises file-management and security operations. Details about the commands can be found in Section 10.2.2. For the security operations we rely on the block ciphers AES and NOEKEON, respectively. Since no asymmetric cryptography is used, INTERNAL_AUTHENTICATE using ECDSA (which is listed Figure 10.2) is not available. High-level protocol functionality and cryptographic algorithm are entirely implemented in the program memory of the microcontroller. Hence, there is no dedicated coprocessor that handles encryption or decryption of data as typically found in the design of security-enabled tags. Random data that is required for security operations is generated within the TRNG and transferred to the memory of the microcontroller over the AMBA bus. The following section gives more detailed information about the utilized security-layer.

12.2 Description of the Security Layer

In order to quantify the costs of adding security functionality to our tag, two security services have been selected for implementation. The two security services are: tag authentication and reader authentication. Tag authentication ensures originality of the tag and prevents simple cloning of it (proof-of-origin). Reader authentication ensures originality of the reader and can be used to restrict access to certain resources on the tag. Hence, only legitimate readers that have successfully authenticated towards the tag are allowed, for example, to change configuration parameters or to read sensitive information from tag memory.

Both services are based on a challenge-response protocol using symmetric-key cryptography as defined in ISO 9798-2 [88]. The deployed cryptographic algorithm is a block cipher with a block size of n bits (n is even). Using symmetric-key cryptography requires that reader and tag share a secret key K. The key can be stored on the tag, for example, during a personalization phase that is performed within a protected environment (*i.e.* it can be assumed that there is no adversary).

12.2.1 Tag Authentication

The basic principle of tag authentication is illustrated in Figure 12.2. The reader initiates the authentication process by sending a randomly selected challenge r_R with a length of $\frac{n}{2}$ bits through a tag-authenticate command (INTERNAL_AUTHENTICATE) to the tag. After receiving r_R from the reader, the tag generates itself a random number r_T of the same length, and encrypts the concatenation of the two random numbers $r_R \mid r_T$ under the secret key K. The encrypted value is then sent to the reader, which can decrypt it with its secret key. If both reader and tag use the same secret key, the decrypted value will contain the random number r_R that has initially been selected by the reader, and the tag is treated as authentic. As shown in Figure 12.2, one additional reader command is necessary that has to be processed by the tag.

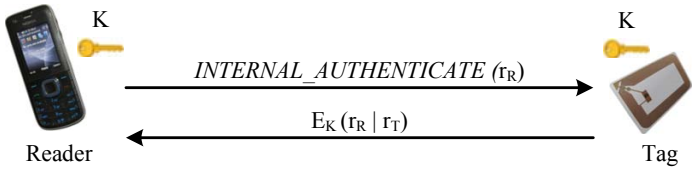


Figure 12.2: Basic principle of tag authentication.

12.2.2 Reader Authentication

The second security service is reader authentication. Figure 12.3 depicts an overview of the communication flow of this security service. The reader starts with sending a request command (GET_CHALLENGE) to the tag, which in turn

generates a random number r_T with a length of $\frac{n}{2}$ bits that is transmitted to the reader. It is important to note that the tag has to store r_T internally to be able to verify later whether the reader is authentic or not. After receiving r_T from the tag, the reader generates its own random number r_R (also with a length of $\frac{n}{2}$ bits), and encrypts the concatenation of the two random values $r_T \mid r_R$ (position of random numbers is interchanged compared to tag authentication) using its secret key K. As next step, the encrypted value is transmitted through a reader-authenticate command (EXTERNAL_AUTHENTICATE) to the tag, which decrypts the value using its secret key. Again, when both reader and tag use the same secret key K, the decrypted value will contain the random number r_T initially selected by the tag, and the reader is treated as authentic. As mentioned before, making this check requires the tag to internally store the value r_T, consuming $\frac{n}{2}$ bits of memory. Alternatively, the reader can also decrypt $r_T \mid r_R$ instead of encrypting it. This has the advantage that the tag only needs to support encryption and not encryption and decryption, which makes for some block ciphers a significant difference in terms of resource usage. The tag finalizes the authentication step by sending a message to the reader with the status of the authentication process (OK or FAIL). As illustrated in Figure 12.3, implementing reader authentication requires two additional reader commands that have to be handled by the tag (GET_CHALLENGE and EXTERNAL_AUTHENTICATE).

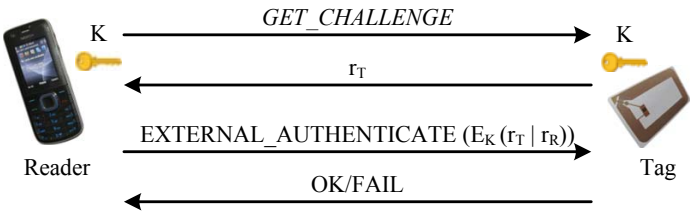


Figure 12.3: Basic principle of reader authentication.

12.2.3 Security-Layer Variants

For a detailed analysis of the costs that are caused by adding a security layer to our tag, three security-layer variants are considered. The first security-layer variant (named *Variant 1* in the following) only supports tag authentication. Thus, the tag needs to implement the encryption function of the block cipher and to handle one additional command. This is the least-expensive scenario. The second security-layer variant (*Variant 2*) realizes both services tag authentication and reader authentication. For reader authentication, the alternative method previously described is used, where the reader decrypts the value $r_T \mid r_R$. Thus, implementing only the encryption function of the block cipher on tag side also suffices for this variant. Three additional reader commands have to be handled by the tag and memory for storing r_T inside the tag has to be provided. The third security-layer variant (*Variant 3*) is the most ex-

pensive one concerning resource usage. Tag and reader authentication are supported. As in case of *Variant 2*, three additional reader commands need to be handled and memory inside the tag has to be reserved for storing r_T. However, the important difference to *Variant 2* is that the reader-authentication approach is used that requires the tag to support also the decryption function of the block cipher. In order to prevent potential attacks on protocol level such as reader impersonation, every tag should use a different secret key K. Further, the tags accepts an EXTERNAL_AUTHENTICATE command only if it directly follows a GET_CHALLENGE command. Hence, using an INTERNAL_AUTHENTICATE command after the GET_CHALLENGE command aborts the reader-authentication process. Table 12.1 gives an overview of the features and requirements of the three security-layer variants.

Table 12.1: Overview of the features and requirements of the three security-layer variants.

Security-layer variant	Tag authentication	Reader authentication	Additional reader commands	Memory for r_T	Encryption required	Decryption required
Variant 1	Yes	No	1	No	Yes	No
Variant 2	Yes	Yes	3	Yes	Yes	No
Variant 3	Yes	Yes	3	Yes	Yes	Yes

12.2.4 Selected Block Ciphers

The previously described security-layer variants base on symmetric-key cryptography. In particular, they rely on a block cipher that is used for encrypting and decrypting data, respectively. Two different block ciphers have been selected for realizing the security-layer variants: AES and NOEKEON. Selecting two different block ciphers allows analyzing their influence on the overall implementation costs of each security-layer variant. AES has been chosen because it is standardized and provides high security. NOEKEON has been selected since it provides a good trade off between security and resource usage. In contrast to AES, encryption and decryption function of NOEKEON can be implemented with very little overhead, keeping the costs of *Variant 3* with this block cipher quite low. A short description of the two block ciphers can be found in Section 11.2.

12.3 Concept for Implementing the Security-Layer Variants

The way we implement the security-layer variants on our tag differs from the traditional approach typically found in related work, where protocol handling and cryptographic algorithm are implemented separately. There, the protocol handling is implemented in a control state machine fixed in hardware and the

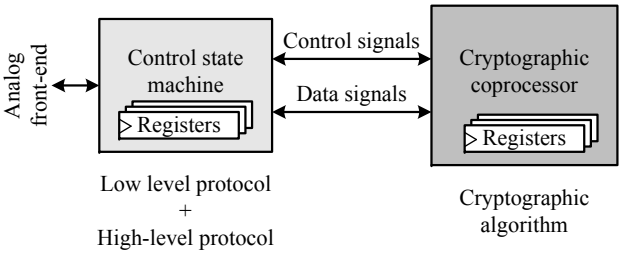


Figure 12.4: Traditional approach where protocol handling and cryptographic algorithm are implemented separately.

cryptographic algorithm is implemented within a coprocessor that is highly optimized for low-resource usage. A schematic view of this approach is given in Figure 12.4. As we have shown in Chapter 10, using a programmable controller for handling complex control tasks on RFID tags is advantageous. Such a design still fulfills the fierce requirements of passive low-cost RFID tags, but makes the design more flexible, easier to maintain, and faster to adapt. As a result, shorter design times can be achieved (time to market) which reduces the overall development costs.

Our tag uses a programmable controller for handling the complex parts of the protocol (high-level protocol). Complex parts of the protocol include for example: reconstructing chained reader commands, handling file-access commands, and managing configuration parameters of the tag. Moreover, when adding a security layer, control complexity further increases. Generation of random values has to be triggered and the values have to be transferred to concerning locations in memory. Encryption and decryption of data has to be initiated and results have to be checked (*e.g.* whether authentication has been successful or not). Combining the security layer with existing tag functionality like handling file-access commands and managing configuration parameters also increases control complexity. Hence, we only use a fixed state machine in hardware (called framing logic) for time-critical commands that require low control complexity (low-level protocol) and whose functionality is typically fixed. Complex protocol parts are processed by an 8-bit microcontroller optimized for low-resource usage. However, when deploying a microcontroller for handling parts of the protocol, we can reuse it for computing cryptographic algorithms as well. A schematic view of this combined approach is presented in Figure 12.5. The program code of the microcontroller contains both the implementation of the high-level protocol and the cryptographic algorithm. Another benefit of this combined approach is the easier and more efficient reuse of resources like memory (registers of the microcontroller).

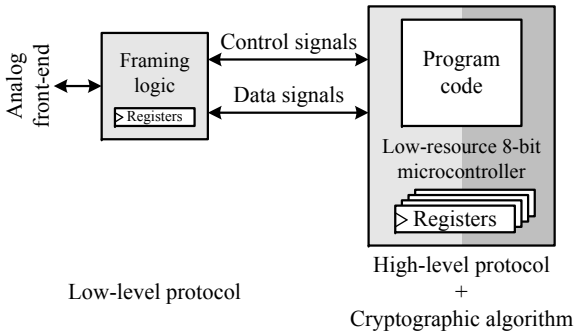


Figure 12.5: Combined approach where high-level protocol and cryptographic algorithm are handled by a low-resource microcontroller.

12.4 Implementation Results

This section presents implementation results of the security-layer variants previously described. We have implemented all three variants using the block ciphers AES and NOEKEON, respectively. For each block cipher, various versions with different optimization targets are used (size, efficiency, and speed). Implementation results are given for a 130 nm CMOS process technology [52] after place and route using Cadence RTL compiler. We have selected a CMOS process technology that is state-of-the art for RFID-tag design to obtain more-realistic results.

In the following a short overview of the deployed microcontroller and the resource-usage of the two block ciphers is given. Afterwards, implementation results of the different security-layer variants are presented.

12.4.1 Low-Resource 8-Bit Microcontroller

Central element of our security-enabled tag is the 8-bit microcontroller optimized for low-resource usage that we have described in Section 11.1. Main components of the microcontroller are: a control unit, a program counter, an arithmetic-logic unit (ALU), a register file, and a program ROM. Size of the register file is flexible and consists of at least 3 special-purpose registers (ACC, PCH, and STATUS). Depending on the targeted application, up to 61 additional registers can be included during design phase, resulting in a maximum of 64 registers. Each of the additional registers can be either configured as general-purpose register for temporarily storing data and making computations, or as input/output register for accessing and controlling external components (*e.g.* the AMBA bus). When more data memory is required, an additional RAM can be connected to the AMBA bus. The instruction set consists of 36 instructions, involving logical operations, arithmetic operations, and control-flow operations (see Table 11.2). The program ROM is realized as look-up table and contains the instructions that the microcontroller should execute. Size of the program ROM is also flexible.

The program ROM is divided into pages, where each page has a size of 512 bytes. Up to 256 pages can be addressed, resulting in a maximum size of 128 kB. Synthesizing the microcontroller core (without register file and program ROM) for a 130 nm CMOS process technology results in a chip area of 1 067 GEs.

12.4.2 Implementation Results of AES and NOEKEON

The two block ciphers AES and NOEKEON have been used for realizing the security-layer variants previously described. For each cipher, three different optimization targets have been used: speed, efficiency, and size. The target speed aims for shortest execution time of the cipher by using techniques like code duplication and loop unrolling, efficiency provides a good trade off between execution time and code size, and size is optimized for minimal code size where as many operations as possible are handled through function calls that can be reused. Encryption function and decryption function of both ciphers are implemented. Moreover, for security-layer variants *Variant 1* and *Variant 2*, also encryption-only versions of the two algorithms are realized (with targets speed and size). Data that needs to be encrypted or decrypted is located in the register file of the microcontroller. The cipher key is stored in the EEPROM and has to be loaded each time during processing of data.

We have taken the implementations of AES and NOEKEON described in Section 11.3 and adapted them for our needs. Two encryption-only version for NOEKEON have been added, one optimized for shortest execution time and one for minimal code size. A summary of the implementation results of AES and NOEKEON is given in Table 12.2. Compared to the results presented in Table 12.2, slightly lower code size and a bit longer execution times are stated for the AES implementations. The reason for this is that the adapted AES implementations in this chapter are using different functions for loading data and key.

Table 12.2: Summary of the implementation results of the block ciphers AES and NOEKEON that are used for the security-layer variants.

Algorithm	**Optimization target**	**Encryption** [clock cycles]	**Decryption** [clock cycles]	**Code size** [bytes]	**Utilized registers** -
AES	size	5 104	8 286	1 602	39
	efficiency	3 369	5 101	1 816	39
	speed	3 149	4 570	2 034	39
AES (encr. only)	size	4 270	n/a	858	39
	speed	3 070	n/a	1 050	39
NOEKEON	size	7 563	7 546	414	23
	efficiency	5 839	5 824	532	25
	speed	3 817	3 785	980	35
NOEKEON (encr. only)	size	7 553	n/a	382	23
	speed	3 805	n/a	652	35

12.4.3 Implementation Results of the Security-Layer Variants

Adding security to our tag influences mainly register-file size and ROM size of the microcontroller. For simplification, costs introduced by the TRNG and through storing additional data like the cipher key in the EEPROM are neglected. These costs are independent of the selected security-layer variant and the chosen block cipher.

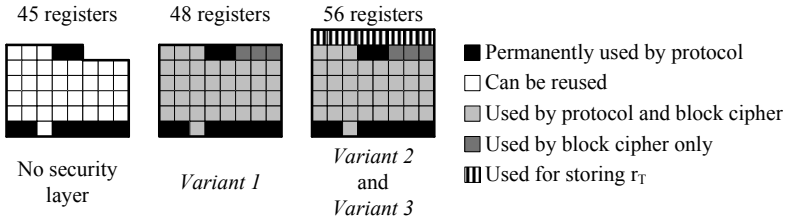


Figure 12.6: Utilization of the register file for different security-layer variants when using the code-size optimized version of AES.

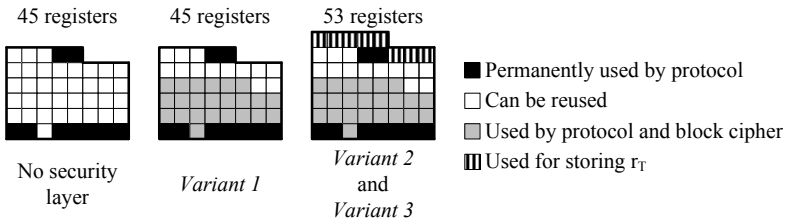


Figure 12.7: Utilization of the register file for different security-layer variants when using the code-size optimized version of NOEKEON.

Register-File Utilization

Our tag with advanced file-management functionality utilizes 45 8-bit registers in the register file and 2 214 bytes of code in the ROM for high-level protocol handling. Compared to the original flexible tag platform presented in Chapter 10, 19 additional registers are used. The increased register usage results from the program-counter register (PCH) that is necessary for accessing the ROM page wise and from the 18 registers that are required for temporarily storing the reader commands (*e.g.* to reassemble chained reader commands). The original flexible tag platform requires no PCH register and reuses memory from the cryptographic unit for temporarily storing the reader commands. Synthesizing the microcontroller with this configuration for a 130 nm target technology results in a chip size of roughly 9 kGEs (after place and route). These values serve as basis for subsequent comparisons.

Without any security layer, 45 registers are utilized by protocol handling. However, only 9 of the 45 registers are permanently used for handling the protocol (*e.g.* to store parameters and the status of the tag). The remaining 36 registers are used for temporarily storing data and are no longer used when a reader command has successfully been received. Consequently, these registers can be reused when computing cryptographic algorithms. Since the computation of AES on our microcontroller requires 39 registers, only 3 additional registers are necessary when combining the computation of protocol and cryptographic algorithm. When using NOEKEON, no additional registers are necessary. Even the "largest" NOEKEON version consumes only 35 registers and fits within the 36 registers that can be reused from protocol handling.

When selecting a security layer based on *Variant 2* or *Variant 3* that involves reader authentication, additional registers are required for storing the random number r_T generated by the tag. Reusing registers from protocol handling for storing r_T during processing of the GET_CHALLENGE command is not possible. When receiving the reader-authenticate command (EXTERNAL_AUTHENTICATE) afterwards, those registers would get overwritten before they can be used for checking the authenticity of the reader. The random number r_T has a length of $\frac{n}{2}$ bits. Since both AES and NOEKEON have a block length $n = 128$ bits, 8 registers are required for storing r_T. As a result, the total number of utilized registers increases to 56 when reader authentication is supported and AES is used. When applying NOEKEON, an overall number of 53 registers is necessary. A detailed view of the register-file utilization for different security-layer variants is given in Figure 12.6 for AES (code-size optimized) and in Figure 12.7 for NOEKEON (code-size optimized).

Total Overhead Costs

For determining the overall costs of the different security-layer variants, not only the size of the register file but also the size of the ROM has to be considered. ROM size is influenced by the security-layer variants through two parameters: the implementation of the block cipher and handling of the additional reader commands. Information about the code size of the different block-cipher implementations have already been given in Section 12.4.2 and will not be discussed here in more detail. The required code size for handling the additional reader commands depends on the security-layer variant. *Variant 1* causes a code-size increase of 250 bytes. This is the lowest value since only one additional command (INTERNAL_AUTHENTICATE) needs to be handled. Both *Variant 2* and *Variant 3* introduce three additional commands (INTERNAL_AUTHENTICATE, GET_CHALLENGE, and EXTERNAL_AUTHENTICATE) and increase the code size by 460 bytes and 452 bytes, respectively. Note that handling *Variant 2* requires slightly more code size than handling *Variant 3*. This results from the fact that additional control-flow operations are required in *Variant 2* (*e.g.* to decide where to continue with protocol handling after encrypting data). Table 12.3 summarizes the overall costs of the different security-layer variants when using either AES or NOEKEON. The lowest

overhead costs are obtained when deploying *Variant 1* with the code-size optimized version of NOEKEON, resulting in 632 bytes of additional code. The most-expensive approach is *Variant 3* with the speed-optimized version of AES, resulting in 11 additional registers and an code-size increase of 2 486 bytes.

Table 12.3: Overview of the overhead costs introduced by the different security-layer variants in terms of additional registers and increased code size.

Security layer		**Protocol costs**		**Block-cipher costs**		**Total costs**	
Variant	**Block cipher**	**Registers**	**Code size**	**Registers**	**Code size**	**Registers**	**Code size**
		-	[bytes]	-	[bytes]	-	[bytes]
	AES						
Variant 1	size	0	250	3	858	**3**	**1 108**
	speed	0	250	3	1 050	**3**	**1 300**
Variant 2	size	8	460	3	858	**11**	**1 318**
	speed	8	460	3	1 050	**11**	**1 510**
Variant 3	size	8	452	3	1 602	**11**	**2 054**
	efficiency	8	452	3	1 816	**11**	**2 268**
	speed	8	452	3	2 034	**11**	**2 486**
	NOEKEON						
Variant 1	size	0	250	0	382	**0**	**632**
	speed	0	250	0	652	**0**	**902**
Variant 2	size	8	460	0	382	**8**	**842**
	speed	8	460	0	652	**8**	**1 112**
Variant 3	size	8	452	0	414	**8**	**866**
	efficiency	8	452	0	532	**8**	**984**
	speed	8	452	0	980	**8**	**1 432**

Synthesizing our tag with the different security-layer variants for a 130 nm CMOS process technology gives actual numbers about the area requirements in hardware. The register file of the microcontroller is built up with latches to minimize chip area. The ROM of the microcontroller is implemented as look-up table which gets mapped by the synthesis tool to an unstructured mass of standard cells. As mentioned above, the 8-bit microcontroller including register file and program ROM consumes about 9 kGEs for high-level protocol handling without security layer. Every 8-bit register that is added for the security layer increases the area by approximately 55 GEs. About 2 GEs per byte are required for additional program code in the ROM. Detailed synthesis results after place and route obtained with Cadence RTL compiler are provided in Table 12.4. The least-expensive security-layer variant, which is *Variant 1* with the code-size optimized version of NOEKEON, results in an area overhead of 1 074 GEs. The most-expensive security-layer variant, which is *Variant 3* with the speed-

optimized version of AES, leads to an overhead of 4 465 GEs. Hence, total size of the digital part of the security-enabled tag with advanced file-management functions lies between 10.1 kGEs and 14.5 kGEs (excluding EEPROM and TRNG).

Table 12.4: Overview of the overhead costs in terms of additional chip area (GEs) after place and route introduced by the different security-layer variants.

Security layer		Protocol costs			Block-cipher costs			Total costs
Variant	**Block cipher**	**Registers**	**ROM size**	**Total**	**Registers**	**ROM size**	**Total**	
		[GEs]	[GEs]	[GEs]	[GEs]	[GEs]	[GEs]	[GEs]
	AES							
Variant 1	size	0	500	**500**	165	1 352	**1 517**	**2 017**
	speed	0	500	**500**	165	1 450	**1 614**	**2 115**
Variant 2	size	453	804	**1 257**	165	1 450	**1 615**	**2 872**
	speed	453	804	**1 257**	165	1 513	**1 678**	**2 935**
Variant 3	size	453	712	**1 165**	165	2 607	**2 772**	**3 937**
	efficiency	453	712	**1 165**	165	2 816	**2 981**	**4 146**
	speed	453	712	**1 165**	165	3 135	**3 300**	**4 465**
	NOEKEON							
Variant 1	size	0	500	**500**	0	574	**574**	**1 074**
	speed	0	500	**500**	0	887	**887**	**1 387**
Variant 2	size	479	804	**1 283**	0	660	**660**	**1 943**
	speed	479	804	**1 283**	0	1 041	**1 041**	**2 323**
Variant 3	size	479	712	**1 191**	0	751	**751**	**1 942**
	efficiency	479	712	**1 191**	0	883	**883**	**2 074**
	speed	479	712	**1 191**	0	1 545	**1 545**	**2 736**

When considering only the area requirement of the block-cipher implementation, AES encryption and decryption function can be realized with 2 772 GEs to 3 300 GEs. Implementing the encryption-only version costs about 1 600 GEs. The overhead costs of NOEKEON are much smaller. Encryption and decryption function of NOEKEON can be implemented with 751 GEs to 1 545 GEs. The encryption-only version consumes between 574 GEs to 1 041 GEs. These low area values are a consequence of heavily reusing registers that are normally utilized for handling the protocol. Comparing these results with the reference implementations of the algorithms in Section 11.3 where no additional protocol handling has been considered let us come to two important conclusions. First, combining protocol handling and computation of the cryptographic algorithm has allowed to improve the area efficiency of the ROM, since the synthesizer can better optimize larger look-up tables. Second, even if not all registers that are required for the computation of the cryptographic algorithm can be reused from protocol handling (as in case of AES), most of the resulting overhead costs are still below the values of the reference implementation. Further, it has to be noted that the area values given in this chapter are already after place and

route, whereas the values of the reference implementations are after synthesis. Typically, area values after place and route are slightly larger than those after synthesis. This emphasizes the efficiency of our approach where protocol handling and computation of the cryptographic algorithm are implemented jointly.

Costs introduced by handling the additional reader commands and potentially storing the random number r_T range from 500 GEs to 1 283 GEs. Although often neglected in related work, handling the protocol part of the security layer constitutes a significant portion of the overall costs. Depending on the deployed cryptographic algorithm, costs for handling the protocol part can be even the dominating factor. This is clearly pointed out by our implementation results of the security-layer variants that use NOEKEON as block cipher (see Table 12.4). When using *Variant 2* with the code-size optimized version of NOEKEON, nearly 66 % of the total area overhead is introduced by the implementation of the protocol.

Timing Results and Power Consumption

Computing a cryptographic algorithm in software on an 8-bit microcontroller requires typically more clock cycles than computing it in hardware on a dedicated coprocessor. Thus, we have evaluated the answer times of the security-layer commands to verify whether the resulting values are still practicable. The answer time is the time between the end of a reader command and the beginning of the tag response. Our RFID tag bases on the ISO 14443 standard [91, 93] that specifies a basic data rate of 106 kbps. Handling the protocol at this data rate requires the 8-bit microcontroller to be clocked at 106 kHz (one cycle per bit). The resulting answer times of the INTERNAL_AUTHENTICATE command and the EXTERNAL_AUTHENTICATE command for the different security-layer variants are listed in Table 12.5. The values for the INTERNAL_AUTHENTICATE command range from 32.53 ms to 74.93 ms, and the values for the EXTERNAL_AUTHENTICATE command range from 30.63 ms to 79.83 ms. Most of the answer time is utilized for computing the block cipher, only little time is consumed by protocol handling. The answer time of the GET_CHALLENGE command is constantly 3.72 ms and is independent of the security-layer variant and the deployed block cipher.

Answer times of 50 ms and more are not critical for our tag. The ISO 14443 standard provides a mechanism called *waiting-time extension* (WTX) that allows to temporarily increase the answer time of the tag for computation-intensive commands. When a higher data rate is used (*e.g.* 212 kbps or 424 kbps as specified in the standard) or when the clock frequency of the microcontroller is increased, shorter answer times can be achieved if required by a specific application. In both cases power consumption of the microcontroller will increase due to the higher clock frequency. In order to get power-consumption values of our microcontroller, we have used Cadence Encounter Power System. Simulating our microcontroller with the most-expensive security-layer variant (*Variant 3* with speed-optimized version of AES) gives an average power value of 2 µW at a clock frequency of 106 kHz and a voltage of 1.2 V (this equals a power con-

Table 12.5: Overview of the answer times of the INTERNAL_AUTHENTICATE command and the EXTERNAL_AUTHENTICATE command for different security-layer variants.

Security-layer		Tag authentication (INTERNAL_AUTH.)			Reader authentication (EXTERNAL_AUTH.)		
Variant	**Block cipher**	**Protocol**	**Block cipher**	**Total**	**Protocol**	**Block cipher**	**Total**
		[ms]	[ms]	[ms]	[ms]	[ms]	[ms]
	AES						
Variant 1	size	3.57	40.28	**43.85**	n/a	n/a	n/a
	speed	3.57	28.96	**32.53**	n/a	n/a	n/a
Variant 2	size	3.60	40.28	**43.88**	1.67	40.28	**41.95**
	speed	3.60	28.96	**32.56**	1.67	28.96	**30.63**
Variant 3	size	3.58	48.15	**51.73**	1.66	78.17	**79.83**
	efficiency	3.58	31.78	**35.36**	1.66	48.12	**49.78**
	speed	3.58	29.71	**33.29**	1.66	43.11	**44.77**
	NOEKEON						
Variant 1	size	3.57	71.25	**74.82**	n/a	n/a	n/a
	speed	3.57	35.90	**39.47**	n/a	n/a	n/a
Variant 2	size	3.60	71.25	**74.85**	1.67	71.25	**72.92**
	speed	3.60	35.90	**39.50**	1.67	35.90	**37.57**
Variant 3	size	3.58	71.35	**74.93**	1.66	71.19	**72.85**
	efficiency	3.58	55.08	**58.66**	1.66	54.94	**56.60**
	speed	3.58	36.01	**39.59**	1.66	35.71	**37.37**

sumption of 1.67 µA). This power value is much lower than the values provided in Chapter 10 and Chapter 11 (around 20 µW) for the microcontroller, since we are now using a more advanced CMOS process technology (130 nm instead of 350 nm). Doubling for example the clock frequency, will roughly also double the power consumption of the microcontroller.

Another advantage that arises from the combined implementation of protocol handling and cryptographic algorithm on the microcontroller is that no additional power is consumed for handling the security layer. When using a dedicated coprocessor, additional power would be required during computation of the cryptographic algorithm. Having low power consumption is an important design goal for passive RFID tags.

12.5 Summary

In this chapter we have presented a security-enabled tag with advanced file-management functionality that is optimized for low resource usage. We have further evaluated the hardware overhead that arises from integrating different security-layer variants. The security-layer variants are based on the cryptographic algorithms AES and NOEKEON, respectively. We have used a com-

bined implementation of high-level protocol handling and cryptographic algorithm on a low-resource 8-bit microcontroller. This combined approach provides high flexibility and allows reusing registers of the microcontroller that are only temporarily used during protocol handling. In that way AES encryption function can be implemented with an overhead of about 1 600 GEs and NOEKEON encryption function with an overhead of about 600 GEs when using a 130 nm CMOS process technology. The microcontroller consumes only 2 µW of power at a clock frequency of 106 kHz. Costs of the security-layer variants range from 1 100 GEs to 4 500 GEs and consider also the protocol handling of the security layer. Protocol handling can make up a significant part of the costs introduced by the security layer (up to 66 %) and must not be neglected. Total size of the digital part of the security-enabled tag with advanced file-management functionality lies between 10.1 kGEs and 14.5 kGEs depending on the selected security layer (excluding EEPROM and TRNG).

13 Conclusion

In this thesis we have discussed security aspects that are very important for designing future low-cost RFID tags. We have performed implementation attacks on commercially available low-cost RFID tags including side-channel analysis and fault analysis. Countermeasures that aim for protecting RFID tags against side-channel analysis have been evaluated. Gaining knowledge about the susceptibility of low-cost tags against implementation attacks is important for properly integrating security mechanisms on them. We have further presented a flexible tag architecture that bases on a low-resource 8-bit microcontroller. The flexible architecture allows to efficiently handle complex control tasks on low-cost tags. We have also shown that symmetric-key algorithms can be implemented in a very compact way on the low-resource microcontroller. Our results point out that a combined implementation of protocol handling and computation of cryptographic algorithms is advantageous for integrating security into future low-cost RFID tags.

After a short motivation and a general introduction to the topic in Chapter 1, we have presented our research work in two parts. In the first part of this thesis we have focused on implementation attacks and on the evaluation of side-channel analysis countermeasures in context of low-cost RFID tags. We have given a brief overview of RFID technology in Chapter 2, followed by background information about implementation attacks in Chapter 3. Countermeasures against implementation attacks are introduced in Chapter 4.

In Chapter 5 we have evaluated the susceptibility of passive UHF RFID tags against side-channel analysis (SCA) attacks. We have analyzed commercially available RFID tags from various tag vendors. Successful differential electromagnetic analysis (DEMA) attacks on all evaluated tags have been presented at distances up to 1 m. The results have pointed out that passive UHF RFID tags

are highly susceptible to remote attacks.

Fault attacks on low-cost HF and UHF tags have been conducted in Chapter 6. For injecting a fault, temporarily antenna tearing, electromagnetic interferences, and optical inductions have been used. Target of the fault injection has been the writing of data to the internal memory of the tags. Our attacks have allowed to interrupt the writing of data at different points in time. The faulty values in the memory are not random, but can be largely influenced by an adversary. Hence, when integrating security to low-cost RFID tags, proper measures have to be taken to prevent such attacks.

After the implementation attacks on low-cost RFID tags, we have continued with the analysis of SCA countermeasures. Since current low-cost tags do not have integrated such countermeasures, prototype devices have been used instead. In Chapter 7 we have evaluated the effectiveness of randomization as an SCA countermeasure for RFID tags. A randomized AES implementation in software on HF and UHF tag prototypes has been used for evaluation. Several preprocessing techniques such as filtering, windowing, and differential frequency analysis (DFA) have been applied. Especially DFA has turned out to be a very effective technique to attack randomization-based countermeasures. However, the effort for attacking commercially available RFID tags is assumed to be higher when the countermeasure is realized in dedicated hardware.

Another SCA countermeasure that is aimed for protecting passive UHF RFID tags from remote attacks has been evaluated in Chapter 8. The so-called detached power-supply countermeasure has been applied to a prototype device that computes the AES. A basic and an enhanced version of the detached power supply has been analyzed. Both versions have shown a susceptibility to SCA attacks due to the non-ideal properties of the deployed analog switches. The SCA leakage can be reduced when increasing the integration interval used by the countermeasure. Using the detached power supply to protect passive UHF RFID tags from remote attacks is feasible. Additional countermeasures have to be integrated when protecting the tags from a more sophisticated attack like measuring the direct emissions close to the tag chip.

In the second part of this thesis we have concentrated on hardware-implementation aspects of low-cost RFID tags. Chapter 9 provides basic information about the design cycle and the design space of hardware circuits in general. We have emphasized the special requirements that have to be taken into account when designing digital hardware circuits for passive low-cost tags.

A flexible tag platform that is intended for implementing passive low-cost RFID tags has been presented in Chapter 10. The flexible tag platform bases on a low-resource 8-bit microcontroller that allows to handle complex control tasks on the tag. The efficiency of this approach has been demonstrated by designing an NFC-compatible tag with advanced file-access functionality and security features. The NFC-compatible tag has been manufactured as a chip in silicon. The results clearly point out that implementation of passive low-cost RFID tags is feasible using our flexible tag platform.

In Chapter 11 several symmetric-key algorithms have been implemented on

the low-resource microcontroller used by our flexible tag platform. The algorithm implementations on our microcontroller are more efficient than on other microcontrollers. When considering the reuse of the microcontroller for protocol handling, our algorithm implementations even outperform optimized hardware modules.

A combined implementation of high-level protocol handling and cryptographic algorithm on the low-resource microcontroller has been shown in Chapter 12. The combined approach provides high flexibility and allows reusing registers of the microcontroller that are only temporarily used during protocol handling. No dedicated hardware module for computing the cryptographic algorithm is necessary. Moreover, different security-layer variants have been evaluated for their resource usage. The results clarify that protocol handling can make up a significant part of the overhead costs introduced by the security layers and must not be neglected.

There are two main conclusions that we can draw from this thesis. First, contactless devices such as low-cost RFID tags are susceptible to implementation attacks similar to contact-based devices. In most cases, attacks on contactless devices are more difficult to conduct because of the strong reader field that disturbs measurements, limited access to I/O pins that makes triggering difficult, and low data rates that limit the number of attacks that can be mounted within a given time. However, the efficiency of attacks can be significantly improved when applying special preprocessing techniques and using programmable prototype tags. Hence, when adding security to low-cost RFID tags, appropriate countermeasure need to be integrated to prevent implementation attacks.

Our second conclusion relates to the hardware design of future low-cost tags. As the results in this thesis have shown, a flexible tag architecture that bases on a simple microcontroller is suitable for the implementation of low-cost RFID tags. Both power consumption and resource usage fulfill the fierce requirements of passive low-cost tags. The microcontroller can efficiently handle complex control tasks, which allows the integration of advanced tag functionality. Having RFID tags with advanced functionality will be an important requirement for the creation of new applications within the future Internet of Things. Moreover, when using a microcontroller for the protocol execution, reusing it for computing cryptographic algorithms is highly advantageous. The resource usage of such an approach is even below that of dedicated hardware implementations of the algorithms.

There are several open research points that need to be addressed in future work. For example, integrating efficient SCA and fault-analysis countermeasures to the algorithm implementations on the microcontroller. This will give details about the overhead costs (*e.g.* power consumption, execution time, register usage, and code size) that are introduced by integrating different countermeasure approaches. Another important topic is the evaluation of instruction-set extensions to lower the execution time of the algorithm implementations. By using instruction-set extensions, the resource usage of the implementations could potentially also be lowered (*e.g.* code size). An interesting research area for low-cost

RFID tags is the design of efficient true random-number generators. Are they able to generate enough random data within the desired time and with sufficient entropy? Also the realization of protected non-volatile memory structures needs to be addressed in future work. How can non-volatile memory be protected from unauthorized access and modification with minimal overhead costs?

Bibliography

[1] AVR-Crypto-Lib. Available online at http://www.das-labor.org/wiki/AVR-Crypto-Lib/en.

[2] RFID-Zapper. Chaos Communication Congress 2005, 2005.

[3] E. Abad, F. Palacio, M. Nuin, Zárate, A. Juarros, J. M. Gómez, and S. Marco. RFID smart tag for traceability and cold chain monitoring of foods: Demonstration in an intercontinental fresh fish logistic chain. *Journal of Food Engineering*, 93(4):394–399, Aug. 2009.

[4] D. G. Abraham, G. M. Dolan, G. P. Double, and J. V. Stevens. Transaction Security System. *IBM Systems Journal*, 30(2):206–229, June 1991.

[5] A. Abrial, J. Bouvier, M. Renaudin, P. Senn, and P. Vivet. A new contactless smart card IC using an on-chip antenna and an asynchronous microcontroller. *Solid-State Circuits, IEEE Journal of*, 36(7):1101 –1107, July 2001.

[6] O. Aciiçmez, W. Schindler, and c. K. Koç. Improving Brumley and Boneh timing attack on unprotected SSL implementations. In *Proceedings of the 12th ACM conference on Computer and communications security*, CCS '05, pages 139–146, New York, NY, USA, 2005. ACM.

[7] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM Side-channel(s). 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 29–45. Springer, 2003.

[8] D. Agrawal, J. R. Rao, and P. Rohatgi. Multi-channel Attacks. In C. D. Walter, Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*, pages 2–16. Springer, 2003.

[9] M. Aigner. Seven reasons for application of standardized crypto functionality on low cost tags. EU RFID Forum, 2007.

[10] F. Amiel, K. Villegas, B. Feix, and L. Marcel:. Passive and Active Combined Attacks: Combining Fault Attacks and Side Channel Analysis. In L. Breveglieri, S. Gueron, I. Koren, D. Naccache, and J.-P. Seifert, editors, *Fourth International Workshop on Fault Diagnosis and Tolerance in Cryptography, 2007, FDTC 2007: Vienna, Austria, 10 September 2007.*, pages 92–102. IEEE Computer Society, September 2007.

[11] 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.

[12] 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.

[13] ARM Ltd. AMBA Advanced Microcontroller Bus Architecture Specification. Available online at http://www.arm.com, 1997.

[14] Atmel Corporation. Website atmel.com - Secure RFID: CryptoRF. http://www.atmel.com/products/SecureRF.

[15] Atmel Corporation. 8-bit AVR Microcontroller with 128K Bytes In-System Programmable Flash. Available online at http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf, August 2007.

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

[17] M. Backes, M. Dürmuth, S. Gerling, M. Pinkal, and C. Sporleder. Acoustic Side-Channel Attacks on Printers. In *USENIX Security Symposium*, pages 307–322, 2010.

[18] D. Bailey and A. Juels. Shoehorning Security into the EPC Standard. In R. D. Prisco and M. Yung, editors, *International Conference on Security in Communication Networks (SCN 2006), Maiori, Italy, September 6-8, 2006, Proceedings*, volume 4116 of *Lecture Notes in Computer Science*, pages 303–320. Springer, September 2006.

[19] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan. The Sorcerer's Apprentice Guide to Fault Attacks. Cryptology ePrint Archive (http://eprint.iacr.org/), Report 2004/100, 2004.

[20] 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.

[21] G. Bertoni, J. Daemen, M. Peeters, V. Rijmen, and G. V. Assche. NOEKEON, The Return, January 2010. Available online at https://cryptolux.org/mediawiki.esc/images/7/7a/Noekeon-ESC.pdf.

[22] E. Biham and A. Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In B. S. K. Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525. Springer, 1997.

[23] M. Blitshteyn. Mastering RFID Label Converting: Where Understanding Static Control Can Help Prevent RFID Transponder Failures. Technical report, Ion Industrial, 2005.

[24] A. Bogdanov, D. Khovratovich, and C. Rechberger. Biclique Cryptanalysis of Full AES. Rump Session, Crypto 2011, 2011. Available online at http://research.microsoft.com/en-us/projects/cryptanalysis/aesbc.pdf.

[25] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurinand, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems – CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, September 2007. ISBN 978-3-540-74734-5.

[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] J. Brouchier, T. Kean, C. Marsh, and D. Naccache. Temperature Attacks. *Security Privacy, IEEE*, 7(2):79 –82, 2009.

[29] D. Brumley and D. Boneh. Remote timing attacks are practical. *Computer Networks*, 48(5):701–716, 2005.

[30] C. D. Canniére and B. Preneel. TRIVIUM Specifications. eSTREAM, ECRYPT Stream Cipher Project (http://www.ecrypt.eu.org/stream), Report 2005/030, April 2005.

[31] 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.

[32] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi. A Cautionary Note Regarding Evaluation of AES Candidates on Smart-Cards. In *Second Advanced Encryption Standard (AES) Candidate Conference*, Rome, Italy, 1999.

[33] Z. Chen and Y. Zhou. Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage. 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 242–254. Springer, 2006.

[34] C.-C. Chia and S.-S. Wang. Efficient Design of an Embedded Microcontroller for Advanced Encryption Standard. In *Proceedings of the 2005 Workshop on Consumer Electronics and Signal Processing (WCEsp 2005)*, 2005. Available online at http://www.mee.chu.edu.tw/labweb/WCEsp2005/96.pdf.

[35] J. Y. Cho. Linear Cryptanalysis of Reduced-Round PRESENT. In J. Pieprzyk, editor, *Topics in Cryptology - CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010, Proceedings*, volume 5985 of *Lecture Notes in Computer Science*, pages 302–317. Springer, March 2010.

[36] C. Clavier, J.-S. Coron, and N. Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, volume 1965 of *Lecture Notes in Computer Science*, pages 252–263. Springer, 2000.

[37] K. J. Compton, B. Timm, and J. VanLaven. A Simple Power Analysis Attack on the Serpent Key Schedule. Cryptology ePrint Archive (http://eprint.iacr.org/), Report 2009/473, 2009.

[38] J.-S. Coron. Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES'99, First International Workshop, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *Lecture Notes in Computer Science*, pages 292–302. Springer, 1999.

[39] P. Corsonello, S. Perri, and M. Margala. A New Charge-Pump Based Countermeasure Against Differential Power Analysis. In *Proceedings of*

the 6th International Conference on ASIC (ASICON 2005), volume 1, pages 66–69. IEEE, 2005.

[40] N. T. Courtois, S. O'Neil, and J.-J. Quisquater. Practical Algebraic Attacks on the Hitag2 Stream Cipher. In P. Samarati, M. Yung, F. Martinelli, and C. A. Ardagna, editors, *Information Security Conference – ISC'09*, volume 5735 of *Lecture Notes in Computer Science*, pages 167–176, Pisa, Italy, September 2009. Springer.

[41] J.-P. Curty, M. Declercq, C. Dehollain, and N. Joehl. *Design and Optimization of Passive UHF RFID Systems*. Springer, 2007. ISBN 978-0-387-35274-9.

[42] J. Daemen, M. Peeters, G. V. Assche, and V. Rijmen. Nessie proposal: NOEKEON, 2000. Available online at http://gro.noekeon.org/Noekeon-spec.pdf.

[43] J. Daemen and V. Rijmen. AES proposal: Rijndael. First AES Conference, August 1998.

[44] J.-F. Dhem, F. Kœune, P.-A. Leroux, P. Mestré, J.-J. Quisquater, and J.-L. Willems. A Practical Implementation of the Timing Attack. In J.-J. Quisquater and B. Schneier, editors, *Smart Card Research and Applications, Third International Conference, CARDIS '98, Louvain-la-Neuve, Belgium, September 14-16, 1998, Proceedings*, number 1820 in Lecture Notes in Computer Science, pages 167–182. Springer, 1998. Available online at http://www.dice.ucl.ac.be/crypto/techreports.html.

[45] J. Di Battista, P. Perdu, J.-C. Courrege, B. Rouzeyre, and L. Torres. Validation of differential light emission analysis on FPGA. In *Signals, Circuits and Systems (SCS), 2009 3rd International Conference on*, pages 1 –5, November 2009.

[46] I. Dinur and A. Shamir. Cube Attacks on Tweakable Black Box Polynomials. In A. Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 278–299. Springer, 2009.

[47] EFTON s.r.o. Implementing SEA on x51 and AVR. Available online at http://www.efton.sk/crypt/sea.htm.

[48] 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 D. Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008, Proceedings*, number 5157 in Lecture Notes in Computer Science, pages 203–220. Springer, 2008. ISBN 978-3-540-85173-8.

[49] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel. A Survey of Lightweight-Cryptography Implementations. *IEEE Design & Test of Computers - Design and Test of ICs for Secure Embedded Computing*, 24(6):522–533, November-December 2007. ISSN 0740-7475.

[50] EPCglobal. EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz Version 1.0.9, January 2005. Available online at http://www.epcglobalinc.org/.

[51] A. Facen and A. Boni. Power Supply Generation in CMOS Passive UHF RFID Tags. *Research in Microelectronics and Electronics 2006, Ph. D.*, pages 33–36, June 2006.

[52] Faraday Technology Corporation. Faraday FSA0A_C 0.13 μm ASIC Standard Cell Library, 2004. Details available online at http://www.faraday-tech.com.

[53] M. Feldhofer. A Concept for Controlling Security-Enhanced RFID Smart Tags using Embedded Microcontrollers. Master's thesis, Institute for Applied Information Processing and Communications (IAIK), Graz University of Technology, Inffeldgasse 16a, 8010 Graz, Austria, October 2003.

[54] M. Feldhofer, M. J. Aigner, M. Hutter, T. Plos, E. Wenger, and T. Baier. Semi-Passive RFID Development Platform for Implementing and Attacking Security Tags. In *Workshop on RFID / USN Security and Cryptography - RISC 2010, November 9-10, London, UK, 2010.*, 2010.

[55] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong Authentication for RFID Systems using the AES Algorithm. 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 357–370. Springer, August 2004.

[56] M. Feldhofer and J. Wolkerstorfer. *RFID Security: Techniques, Protocols and System-On-Chip Design*, chapter Hardware Implementation of Symmetric Algorithms for RFID Security, pages 373–415. Springer, 2008.

[57] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen. AES Implementation on a Grain of Sand. *IEE Proceedings on Information Security*, 152(1):13–20, October 2005.

[58] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting. Improved Cryptanalysis of Rijndael. In B. Schneier, editor, *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings*, volume 1978 of *Lecture Notes in Computer Science*, pages 213–230. Springer, April 2000.

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

[60] D. Gajski and R. H. Kuhn. New VLSI Tools - Guest Eidtors' Introduction. *IEEE Computer*, 16(12):11–14, 1983.

[61] K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic Analysis: Concrete Results. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.

[62] S. Garfinkel and B. Rosenberg. *RFID - Applications, Security, and Privacy*. Addison-Wesley, 2005.

[63] C. H. Gebotys, S. Ho, and C. C. Tiu. EM Analysis of Rijndael and ECC on a Wireless Java-Based PDA. In J. R. Rao and B. Sunar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*, pages 250–264. Springer, 2005.

[64] C. H. Gebotys, C. C. Tiu, and X. Chen. A Countermeasure for EM Attack of a Wireless PDA. In *International Conference on Information Technology: Coding and Computing (ITCC 2005), April 4-6, 2005, Las Vegas, Nevada, USA, Proceedings*, volume 1, pages 544–549. IEEE Computer Society, April 2005. ISBN 0-7695-2315-3.

[65] H. Gilbert and T. Peyrin. Super-Sbox Cryptanalysis: Improved Attacks for AES-like permutations. Cryptology ePrint Archive, Report 2009/531, 2009. http://eprint.iacr.org/.

[66] R. God. Lean Manufacturing of RFID Products - Put the Chip on the Box! In *Electronics Systemintegration Technology Conference, 2006. 1st*, volume 2 of *IEEE Conference Proceedings*, pages 1118–1121. IEEE Computer Society, September 2006.

[67] H. Grünbacher and A. Jaud. JAPROC - An 8 bit Micro Controller Design and Its Test Environment. In *Selected papers from the Second International Workshop on Field-Programmable Logic and Applications, Field-Programmable Gate Arrays: Architectures and Tools for Rapid Prototyping*, pages 146–151, London, UK, 1993. Springer-Verlag.

[68] J. Ha, C. Kim, S. Moon, I. Park, and H. Yoo. Differential Power Analysis on Block Cipher ARIA. In L. T. Yang, O. F. Rana, B. D. Martino, and J. Dongarra, editors, *High Performance Computing and Communcations, First International Conference, HPCC 2005, Sorrento, Italy, September 21-23, 2005, Proceedings*, volume 3726 of *Lecture Notes in Computer Science*, pages 541–548. Springer, 2005.

[69] 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.

[70] H. Handschuh and H. M. Heys. A Timing Attack on RC5. In S. E. Tavares and H. Meijer, editors, *Selected Areas in Cryptography '98, SAC'98, Kingston, Ontario, Canada, August 17-18, 1998, Proceedings*, volume 1556 of *Lecture Notes in Computer Science*, pages 306–318. Springer, 1999.

[71] 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.

[72] C. Herbst, E. Oswald, and S. Mangard. An AES Smart Card Implementation Resistant to Power Analysis Attacks. In J. Zhou, M. Yung, and F. Bao, editors, *Applied Cryptography and Network Security, Second International Conference, ACNS 2006*, volume 3989 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 2006.

[73] N. Homma, S. Nagashima, Y. Imai, T. Aoki, and A. Satoh. High-Resolution Side-Channel Attack Using Phase-Based Waveform Matching. 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 187–200. Springer, October 2006.

[74] Y. Hong, C. F. Chan, J. Guo, Y. S. Ng, W. Shi, M. Ho, L. K. Leung, K. N. Leung, C. S. Choy, and K. P. Pun. Design and Challenges of Passive UHF RFID Tag in 90nm CMOS Technology. In *IEEE International Conference on Electron Devices and Solid-State Circuits, 2008. EDSSC 2008.*, pages 1 –4. IEEE Computer Society, dec. 2008.

[75] E. Hubbers, W. Mostowski, and E. Poll. Tearing Java Cards. In *Proceedings of the 7th Edition of e-smart conference and demos, September 20-22, 2006 - Sophia Antipolis, French Riviera, France*, 2006.

[76] M. Hutter, M. Feldhofer, and T. Plos. An ECDSA Processor for RFID Authentication. 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 189–202. Springer, 2010.

[77] M. Hutter, M. Feldhofer, and J. Wolkerstorfer. A Cryptographic Processor for Low-Resource Devices: Canning ECDSA and AES like Sardines. In C. A. Ardagna and J. Zhou, editors, *Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems, Fifth*

International Workshop, WISTP 2011, Heraklion, Crete, Greece, June 1-3, 2011, Proceedings., volume 6633 of *Lecture Notes in Computer Science*, pages 144–159. Springer, 2011.

[78] M. Hutter, S. Mangard, and M. Feldhofer. Power and EM Attacks on Passive 13.56 MHz RFID Devices. In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems – CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 320–333. Springer, September 2007.

[79] M. Hutter, T. Plos, and M. Feldhofer. On the Security of RFID Devices Against Implementation Attacks. *International Journal of Security and Networks 2010*, 5(2/3):106–118, 2010.

[80] M. Hutter, J.-M. Schmidt, and T. Plos. RFID and its Vulnerability to Faults. In E. Oswald and P. Rohatgi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2008, 10th International Workshop, Washington DC, USA, August 10-13, 2008, Proceedings*, volume 5154 of *Lecture Notes in Computer Science*, pages 363–379. Springer, August 2008.

[81] M. Hutter, J.-M. Schmidt, and T. Plos. Contact-Based Fault Injections and Power Analysis on RFID Tags. In *European Conference on Circuit Theory and Design 2009, ECCTD*, 2009.

[82] IEEE. IEEE Standard 1076-2000: VHDL Language Reference Manual. Available online at http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=893288, 2000. ISBN 0-7381-1948-2.

[83] IEEE. IEEE Standard 1364-2001: Verilog hardware description language. Available online at http://ieeexplore.ieee.org/servlet/opac?punumber=7578, March 2001. ISBN 0-7381-2826-0.

[84] Infineon Technologies AG. Secuity and Chip Card ICs SLE 66CX1360PE, 2002.

[85] Infineon Technologies AG. Security and Chip Card ICs SLE 88CFX4000P. Available online at http://www.ic-on-line.cn/iol/datasheet/sle88cfx4000p_1310434.pdf, 2003.

[86] International Organisation for Standardization (ISO). ISO/IEC 7816: Identification cards - Integrated circuit(s) cards with contacts, 1989.

[87] 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.

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

[89] 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.

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

[91] International Organization for Standardization (ISO). ISO/IEC 14443-3: Identification Cards - Contactless Integrated Circuit(s) Cards - Proximity Cards - Part3: Initialization and Anticollision. Available online at http://www.iso.org, 2001.

[92] International Organization for Standardization (ISO). ISO/IEC 18000-6: Information Technology AIDC Techniques — RFID for Item Management – Part 6: Parameters for air interface communications at 860-960 MHz, 2004.

[93] International Organization for Standardization (ISO). ISO/IEC 14443-4: Identification Cards - Contactless Integrated Circuit(s) Cards - Proximity Cards - Part4: Transmission Protocol. Available online at http://www.iso.org, 2008.

[94] J. Jaffe. More Differential Power Analysis: Selected DPA Attacks, June 2006. Presented at ECRYPT Summerschool on Cryptographic Hardware, Side Channel and Fault Analysis.

[95] M. Joye, A. K. Lenstra, and J.-J. Quisquater. Chinese Remaindering Based Cryptosystems in the Presence of Faults. *Journal of Cryptology*, 12(4):241–245, December 1999. ISSN 0933-2790.

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

[97] M. G. Karpovsky, K. J. Kulikowski, and A. Taubin. Robust Protection against Fault-Injection Attacks on Smart Cards Implementing the Advanced Encryption Standard. In *2004 International Conference on Dependable Systems and Networks (DSN 2004), 28 June - 1 July 2004, Florence, Italy, Proceedings*, DSN, pages 93–101. IEEE Computer Society, 2004.

[98] R. Karri, K. Wu, P. Mishra, and Y. Kim. Concurrent Error Detection of Fault-Based Side-Channel Cryptanalysis of 128-Bit Symmetric Block Ciphers. In *Proceedings of the 38th Design Automation Conference, DAC*

2001, Las Vegas, NV, USA, June 18-22, 2001, pages 579–585. ACM, June 2001.

[99] U. Karthaus and M. Fischer. Fully Integrated Passive UHF RFID Transponder IC With 16.7-µW Minimum RF Input Power. *IEEE Journal of Solid-State Circuits*, 38:1602–1608, 2003.

[100] C. H. Kim and J.-J. Quisquater. Fault Attacks for CRT Based RSA: New Attacks, New Results, and New Countermeasures. In D. Sauveron, C. Markantonakis, A. Bilas, and J.-J. Quisquater, editors, *Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems, First IFIP TC6 / WG 8.8 / WG 11.2 International Workshop, WISTP 2007, Heraklion, Crete, Greece, May 9-11, 2007, Proceedings.*, volume 4462 of *Lecture Notes in Computer Science*, pages 215–228. Springer, 2007.

[101] D. A. Kirkpatrick and A. L. Sangiovanni-Vincentelli. Techniques For Crosstalk Avoidance In The Physical Design Of High-performance Digital Systems. In *IEEE/ACM International Conference on Computer-Aided Design, 1994E*, pages 616–619, Nov 1994.

[102] L. R. Knudsen and H. Raddum. On Noekeon NES/DOC/UIB/WP3/009/1, 2001. Available online at https://www.cosic.esat.kuleuven.be/nessie/reports/phase1/uibwp3-009.pdf.

[103] P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In N. Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, number 1109 in Lecture Notes in Computer Science, pages 104–113. Springer, 1996.

[104] P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In M. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[105] O. Kömmerling and M. G. Kuhn. Design Principles for Tamper-Resistant Smartcard Processors. In *Proceedings of the 1st USENIX Workshop on Smartcard Technology (Smartcard '99), Chicago, Illinois, USA, May 1011, 1999*, pages 9–20, McCormick Place South, May 1999. USENIX Association. ISBN 1-880446-34-0.

[106] M. G. Kuhn. *Compromising emanations: eavesdropping risks of computer displays*. PhD thesis, University of Cambridge, 2003. Available online at http://www.cl.cam.ac.uk/TechReports/.

[107] K. J. Kulikowski, M. G. Karpovsky, and A. Taubin. Robust Codes for Fault Attack Resistant Cryptographic Hardware. In *Second Workshop on*

Fault Diagnosis and Tolerance in Cryptography - FDTC 2005, Edinburgh, Scotland, UK, September 2, 2005, Proceedings, September 2005. Revised and republished in the FDTC 2006 proceedings with the title "Fault Attack Resistant Cryptographic Hardware with Uniform Error Detection".

[108] K. J. Kulikowski, M. G. Karpovsky, and A. Taubin. Fault Attack Resistant Cryptographic Hardware with Uniform Error Detection. In L. Breveglieri, I. Koren, D. Naccache, and J.-P. Seifert, editors, *Fault Diagnosis and Tolerance in Cryptography, Third International Workshop, FDTC 2006, Yokohama, Japan, October 10, 2006, Proceedings*, volume 4236 of *Lecture Notes in Computer Science*, pages 185–195. Springer, October 2006. Revised and republished version of the FDTC 2005 paper "Robust Codes for Fault Attack Resistant Cryptographic Hardware".

[109] K. Lemke, K. Schramm, and C. Paar. DPA on n-Bit Sized Boolean and Arithmetic Operations and Its Application to IDEA, RC6, and the HMAC-Construction. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems – CHES 2004, 6th International Workshop, Cambridge, MA, USA, August 11-13, 2004, Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 205–219. Springer, 2004.

[110] J.-C. Lo, S. Thanawastien, and T. R. N. Rao. Concurrent error detection in arithmetic and logical operationsusing Berger codes. In *Proceedings of 9th Symposium on Computer Arithmetic*, 1989.

[111] J. Lu. Related-key rectangle attack on 36 rounds of the XTEA block cipher. In *International Journal of Information Security*, volume 8, pages 1–11. Springer, February 2009.

[112] F. Mace, F.-X. Standaert, and J.-J. Quisquater. ASIC Implementations of the Block Cipher SEA for Constrained Applications. In J. Munilla, A. Peinado, and V. Rijmen, editors, *Workshop on RFID Security 2007 (RFIDSec07), July 11-13, Malaga, Spain*, pages 103–114, 2007.

[113] A. S. Man, E. S. Zhang, V. K. Lau, C. Tsui, and H. C. Luong. Low Power VLSI Design for a RFID Passive Tag baseband System Enhanced with an AES Cryptography Engine. In *1st Annual RFID Eurasia, Istanbul, Turkey, September 5-6, 2007, Proceedings*, pages 1–6. IEEE, September 2007.

[114] S. Mangard. A Simple Power-Analysis (SPA) Attack on Implementations of the AES Key Expansion. In P. J. Lee and C. H. Lim, editors, *Information Security and Cryptology - ICISC 2002, 5th International Conference Seoul, Korea, November 28-29, 2002, Revised Papers*, volume 2587 of *Lecture Notes in Computer Science*, pages 343–358. Springer, 2003.

[115] S. Mangard. Exploiting Radiated Emissions - EM Attacks on Cryptographic ICs. In T. Ostermann and C. Lackner, editors, *Proceedings of*

Austrochip 2003, October 3, 2003, Linz, Austria, pages 13–16, October 2003. ISBN 3-200-00021-X.

[116] S. Mangard, M. Aigner, and S. Dominikus. A Highly Regular and Scalable AES Hardware Architecture. *IEEE Transactions on Computers*, 52(4):483–491, April 2003.

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

[118] Marko Pavlin. Encription Using Low Cost Microcontrollers. Available online at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.61.5755&rep=rep1&type=pdf.

[119] V. Mattoli, B. Mazzolai, A. Mondini, S. Zampolli, and P. Dario. Flexible Tag Datalogger for Food Logistics. *Proceedings of the Eurosensors XXIII Conference*, pages 1215–1218, 7 2009.

[120] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *USENIX Workshop on Smartcard Technology (Smartcard '99)*, pages 151–162, May 1999.

[121] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Power Analysis Attacks of Modular Exponentiation in Smartcards. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES'99, First International Workshop, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *Lecture Notes in Computer Science*, pages 144–157. Springer, 1999.

[122] Microchip. PIC16F - Flash MCU with nanoWatt XLP Technology. Available online at http://www.microchip.com, 2010.

[123] Microchip Technology Inc. AN821: Advanced Encryption Standard Using the PIC16XXX. Available online at http://ww1.microchip.com/downloads/en/AppNotes/00821a.pdf, June 2002.

[124] Microchip Technology Inc. AN953: Data Encryption Routines for PIC18 Microcontrollers. Available online at http://ww1.microchip.com/downloads/en/AppNotes/00953a.pdf, January 2005.

[125] P. Midya. Efficiency analysis of switched capacitor doubler. In *Circuits and Systems, 1996., IEEE 39th Midwest symposium on*, volume 3, pages 1019–1022, Aug 1996.

[126] H. Mika, H. Mikko, and Y.-o. Arto. Practical Implementations of Passive and Semi-passive NFC Enabled Sensors. In *Proceedings of the 2009 First International Workshop on Near Field Communication*, NFC '09, pages 69–74, Washington, DC, USA, 2009. IEEE Computer Society.

[127] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang. Pushing the Limits: A Very Compact and a Threshold Implementation of AES. In K. G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011, 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 69–88. Springer, 2011.

[128] E. D. Mulder, P. Buysschaert, S. B. rs, B. Preneel, G. Vandenbosch, and I. Verbauwhede. Electromagnetic Analysis Attack on an FPGA Implementation of an Elliptic Curve Cryptosystem. In *The International Conference on Computer as a Tool 2005, EUROCON 2005*, pages 1879–1882, November 2005.

[129] National Institute of Standards and Technology (NIST). FIPS PUB 140-1: Security Requirements for Cryptographic Modules, 1994. Available online at http://www.itl.nist.gov/fipspubs/.

[130] National Institute of Standards and Technology (NIST). FIPS-186-2: Digital Signature Standard (DSS), January 2000. Available online at http://www.itl.nist.gov/fipspubs/.

[131] National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard, November 2001. Available online at http://www.itl.nist.gov/fipspubs/.

[132] National Institute of Standards and Technology (NIST). FIPS-186-3: Digital Signature Standard (DSS), 2009. Available online at http://www.itl.nist.gov/fipspubs/.

[133] R. M. Needham and D. J. Wheeler. Tea extensions. Technical report, Computer Laboratory, University of Cambridge, October 1997.

[134] NFC Forum. NFC Forum Type 4 Tag Operation - Technical Specification, March 2007.

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

[136] N. S. A. (NSA). TEMPEST: a signal problem - the story of the discovery of various compromising radiations from communications and comsec equipment. Available online at http://www.nsa.gov/public_info/_files/cryptologic_spectrum/tempest.pdf, 1972.

[137] NXP Austria GmbH. Website mifare.net - contactless smart cards. http://www.mifare.net.

[138] NXP Semiconductors. P5Cx012 - Secure dual interface and contact PKI smart card controller. Available online at http://www.nxp.com/documents/data_sheet/P5CX012_02X_40_73_80_144_FAM_SDS.pdf, 2008.

[139] NXP Semiconductors. LPC1000(L) - 32-bit MCU. Available online at http://www.nxp.com, 2011.

[140] Y. Oren and M. Feldhofer. WIPR – Public Key Identification on Two Grains of Sand. In S. Dominikus, editor, *Workshop on RFID Security 2008 (RFIDSec08), July 9-11, Budapest, Hungary*, pages 15–27, July 2008.

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

[142] S. B. Örs, F. K. Gürkaynak, E. Oswald, and B. Preneel. Power-Analysis Attack on an ASIC AES Implementation. In *International Conference on Information Technology: Coding and Computing (ITCC 2004), April 5-7, 2004, Las Vegas, Nevada, USA, Proceedings*, volume 2, pages 546–552. IEEE Computer Society, April 2004. ISBN 0-7695-2108-8.

[143] E. Oswald. *Advances In Elliptic Curve Cryptography*, volume 317 of *London Mathematical Society Lecture Note Series*, chapter IV, Side-Channel Analysis, pages 69–86. Cambridge University Press, 2005.

[144] J. Park, H. Lee, and M. Ahn. Side-Channel Attacks against ARIA on Active RFID Device. In *International Conference on Convergence Information Technology, 2007*, pages 2163–2168. IEEE Computer Society, November 2007.

[145] C. Piguet, J.-M. Masgonty, C. Arm, S. Durand, T. Schneider, F. Rampogna, C. Scarnera, C. Iseli, J.-P. Bardyn, R. Pache, and E. Dijkstra. Low-power design of 8-b embedded CoolRisc microcontroller cores. *Solid-State Circuits, IEEE Journal of*, 32(7):1067 –1078, July 1997.

[146] T. Plos. Implementation of a Security-Enhanced Semi-Passive UHF RFID Tag. Master's thesis, Institute for Applied Information Processing and Communications (IAIK), Graz University of Technology, Inffeldgasse 16a, 8010 Graz, Austria, May 2007.

[147] T. Plos. Susceptibility of UHF RFID Tags to Electromagnetic Analysis. In T. Malkin, editor, *Topics in Cryptology - CT-RSA 2008, The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008, Proceedings*, volume 4964 of *Lecture Notes in Computer Science*, pages 288–300. Springer, April 2008.

[148] T. Plos. Evaluation of the Detached Power Supply as Side-Channel Analysis Countermeasure for Passive UHF RFID Tags. In M. Fischlin, editor, *Topics in Cryptology - CT-RSA 2009, The Cryptographers' Track at the RSA Conference 2009, San Francisco, CA, USA, April 20-24, 2009, Proceedings*, volume 5473 of *Lecture Notes in Computer Science*, pages 444–458. Springer, April 2009.

[149] T. Plos and M. Feldhofer. Analyzing the Hardware Costs of Different Security-Layer Variants for a Low-Cost RFID Tag. In *Security and Privacy in Communication Networks - 7th International ICST Conference, SecureComm 2011, London, UK, September 7-9, 2011, Proceedings*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer, 2011.

[150] T. Plos and M. Feldhofer. Hardware Implementation of a Flexible Tag Platform for Passive RFID Devices. In *Proceedings of the 14th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2011), Oulu, Finland, August, 2011, Proceedings*, pages xxx–xxx. IEEE Computer Society, August 2011.

[151] T. Plos, H. Groß, and M. Feldhofer. Implementation of Symmetric Algorithms on a Synthesizable 8-Bit Microcontroller Targeting Passive RFID Tags. In A. Biryukov, G. Gong, and D. Stinson, editors, *17th Annual Workshop on Selected Areas in Cryptography - SAC 2010, Waterloo, Canada, August 12-13, 2010, Proceedings*, volume 6544 of *Lecture Notes in Computer Science*, pages 114–129. Springer, 2010.

[152] T. Plos, M. Hutter, and M. Feldhofer. On Comparing Side-Channel Preprocessing Techniques for Attacking RFID Devices. In H. Y. Youm and M. Yung, editors, *Information Security Applications, 10th International Workshop, WISA 2009, Busan, Korea, August 25-27, 2009, Revised Selected Papers*, volume 5932 of *Lecture Notes in Computer Science*, pages 163–177. Springer, December 2009.

[153] T. Plos, M. Hutter, and C. Herbst. Enhancing Side-Channel Analysis with Low-Cost Shielding Techniques. In M. S. Christoph Lackner, Timm Ostermann and R. Spilka, editors, *Proceedings of Austrochip 2008, October 8, 2008, Linz, Austria*, pages 90–95, October 2008. ISBN 978-3-200-01330-8.

[154] T. Popp, M. Kirschbaum, T. Zefferer, and S. Mangard. Evaluation of the Masked Logic Style MDPL on a Prototype Chip. In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems – CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 81–94. Springer, September 2007. ISBN 978-3-540-74734-5.

[155] T. Popp and S. Mangard. Masked Dual-Rail Pre-Charge Logic: DPA-Resistance without Routing Constraints. In J. R. Rao and B. Sunar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*, pages 172–186. Springer, 2005.

[156] 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.

[157] I. K. Proudler. Idempotent AN codes. In *IEEE Colloquium on Signal Processing Applications of Finite Field Mathematics*, pages 8/1–8/5, London, UK, June 1989. IEEE.

[158] J.-J. Quisquater and D. Samyde. A new Tool for Non-Intrusive Analysis of Smart Cards Based on Electro-Magnetic Emissions, the SEMA and DEMA Methods,. Presented at the rump session of EUROCRYPT 2000, 2000.

[159] J.-J. Quisquater and D. Samyde. Eddy Current for Magnetic Analysis with Active Sensor. In *Proceedings of the 3rd International Conference on Research in SmartCards (E-Smart'02), Nice, France, September, 2002*, pages 185–194. UCL, September 2002.

[160] T. Rao. Biresidue Error-Correcting Codes for Computer Arithmetic. *Computers, IEEE Transactions on*, C-19:398–402, May 1970.

[161] 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.

[162] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978. ISSN 0001-0782.

[163] 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.

[164] 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.

[165] J.-M. Schmidt, T. Plos, M. Kirschbaum, M. Hutter, M. Medwed, and C. Herbst. Side-Channel Leakage Across Borders. In D. Gollmann and J.-L. Lanet, editors, *Smart Card Research and Advanced Applications 9th IFIP WG 8.8/11.2 International Conference, CARDIS 2010, April 13-16, 2010, Passau, Germany, Proceedings*, Lecture Notes in Computer Science, pages 36–48. Springer, April 2010.

[166] SecureRF. SecureRF - Secure RFID Solutions. http://http://www.securerf.com.

[167] A. Shamir. Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, volume 1965 of *Lecture Notes in Computer Science*, pages 71–77. Springer, 2000.

[168] A. Shamir. Method and Apparatus for Protecting RFID Tags from Power Analysis. Patent Number WO 2008/019246 A2, February 2008. Available online at http://www.freepatentsonline.com/.

[169] A. Shamir and E. Tromer. Acoustic cryptanalysis - On nosy people and noisy machines. http://www.wisdom.weizmann.ac.il/~tromer/acoustic/. preliminary proof-of-concept presentation.

[170] Silicon Laboratories. C8051F90x-91x Ultra Low-Power MCUs. Available online at http://www.silabs.com, 2010.

[171] S. P. Skorobogatov. *Semi-invasive attacks - A new approach to hardware security analysis*. PhD thesis, University of Cambridge - Computer Laboratory, 2005. Available online at http://www.cl.cam.ac.uk/TechReports/.

[172] 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.

[173] ST Microelectronics. ST22N144 - Smartcard 32-bit RISC MCU with 144 KBytes EEPROM. Available online at http://www.sea.com.ua/img/info/stm/st22n144.pdf, 2006.

[174] F.-X. Standaert, G. Piret, N. Gershenfeld, and J.-J. Quisquater. SEA: a Scalable Encryption Algorithm for Small Embedded Applications. *Lecture Notes in Computer Science*, 3928:222–236, 2006.

[175] H. Stockman. Communication by Means of Reflected Power. *Proceedings of the IRE*, 36(10):1196 – 1204, oct. 1948.

[176] D. Suzuki, M. Saeki, and T. Ichikawa. Random Switching Logic: A Countermeasure against DPA based on Transition Probability. Cryptology ePrint Archive (http://eprint.iacr.org/), Report 2004/346, December 2004.

[177] J. M. Tahir, S. S. Dlay, R. N. G. Naguib, and O. R. Hinton. Fault tolerant arithmetic unit using duplication and residue codes. *Integr. VLSI J.*, 18(2-3):187–200, 1995.

[178] K. Tan, S. Tan, and S. Ong. Functional failure analysis on analog device by optical beam induced current technique. In M. Radhakrishnan, P. Ho, and C. W. Kin, editors, *Proceedings of the 6th International Symposium on Physical and Failure Analysis of Integrated Circuits (IPFA97), Raffles City Convention Centre, Singapore, July 21-25, 1997*, pages 296–301, Raffles City Convention Centre, Singapore, July 1997. IEEExplore, IEEE.

[179] Texas Instruments. MSP430C11x1 - Mixed Signal Microcontroller. Available online at http://focus.ti.com, 2008.

[180] S. Tillich. *Instruction Set Extensions for Support of Cryptography on Embedded Systems*. PhD thesis, Institute for Applied Information Processing and Communications (IAIK), Graz University of Technology, Inffeldgasse 16a, 8010 Graz, Austria, November 2008.

[181] S. Tillich, M. Feldhofer, M. Kirschbaum, T. Plos, J.-M. Schmidt, and A. Szekely. Hardware Implementations of the Round-Two SHA-3 Candidates: Comparison on a Common Ground. In *Proceedings of Austrochip 2010, October 6, 2010, Villach, Austria*, pages 43–48, October 2010. ISBN 978-3-200-01945-4.

[182] S. Tillich, M. Feldhofer, M. Kirschbaum, T. Plos, J.-M. Schmidt, and A. Szekely. Uniform Evaluation of Hardware Implementations of the Round-Two SHA-3 Candidates. August 2010.

[183] P. Tuyls and L. Batina. RFID-Tags for Anti-counterfeiting. In D. Pointcheval, editor, *Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings*, volume 3860 of *Lecture Notes in Computer Science*, pages 115–131. Springer, 2006.

[184] M. Vielhaber. Breaking ONE.FIVIUM by AIDA an Algebraic IV Differential Attack. Cryptology ePrint Archive (http://eprint.iacr.org/), Report 2007/413, 2007. http://eprint.iacr.org/.

[185] X. Wang, Y. L. Yin, and H. Yu. Finding Collisions in the Full SHA-1. In V. Shoup, editor, *Advances in Cryptology - CRYPTO 2005, 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer, 2005.

[186] X. Wang and H. Yu. How to Break MD5 and Other Hash Functions. In R. Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2005.

[187] M. Witteman. Advances in Smartcard Security. *Information Security Bulletin*, (7):11–22, July 2002.

[188] W. H. Wong. Timing attacks on RSA: revealing your secrets through the fourth dimension. *ACM Crossroads*, 11(3):5, 2005.

[189] K. Wonkong, K. Seungchul, B. Younghwan, J. Sungik, P. Youngsoo, and C. Hanjin. A Platform-Based SoC Design of a 32-bit Smart Card. *ETRI journal*, 25 6(25/6):510 – 516, 2003.

[190] P. Wright and P. Greengrass. *Spycatcher / by Peter Wright ; with Paul Greengrass.* Mandarin, Port Melbourne :, 1989.

[191] K. Wu, R. Karri, G. Kuznetsov, and M. Gössel. Low Cost Concurrent Error Detection for the Advanced Encryption Standard. In *Proceedings 2004 International Test Conference (ITC 2004), October 26-28, 2004, Charlotte, NC, USA*, pages 1242–1248. IEEE, 2004.

[192] M. Wu, X. Zeng, J. Han, Y. Wu, and Y. Fan. A high-performance platform-based SoC for information security. In *Proceedings of the 2006 Asia and South Pacific Design Automation Conference*, ASP-DAC '06, pages 122–123, Piscataway, NJ, USA, 2006. IEEE Press.

[193] H. Yan, H. Jianyun, L. Qiang, and M. Hao. Design of low-power baseband-processor for RFID tag. In *International Symposium on Applications and the Internet Workshops, (SAINT 2006), Phoenix, Arizona, USA ,23-27 January, 2006. Proceedings*, pages 4–7. IEEE Computer Society, January 2006.

[194] B. Yang, K. Wu, and R. Karri. Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard. In *Proceedings of the International Test Conference on International Test Conference*, CCS '05, pages 139–146, New York, NY, USA, 2005. ACM.

[195] B. Yang, K. Wu, and R. Karri. Secure Scan: A Design-for-Test Architecture for Crypto Chips. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 25(10):2287–2293, 2006.

[196] S. Yang, W. Wolf, N. Vijaykrishnan, D. N. Serpanos, and Y. Xie. Power Attack Resistant Cryptosystem Design: A Dynamic Voltage and Frequency Switching Approach. In *2005 Design, Automation and Test in Europe Conference and Exposition (DATE 2005), 7-11 March 2005, Munich, Germany*, pages 64–69. IEEE Computer Society, 2005. ISBN 0-7695-2288-2.

[197] Z. Yang and S. Mourad. Deep Submicron On-chip Crosstalk. In *Instrumentation and Measurement Technology Conference, 1999. IMTC/99. Proceedings of the 16th IEEE*, volume 3, pages 1788–1793 vol.3, 1999.

[198] D. J. Yeager, A. P. Sample, and J. R. Smith. *RFID Handbook: Applications, Technology, Security, and Privacy*, chapter WISP: A Passively Powered UHF RFID Tag with Sensing and Computation, pages 261–278. CRC Press, 2008.

[199] S.-M. Yen and M. Joye. Checking Before Output May Not Be Enough Against Fault-Based Cryptanalysis. In *IEEE Transactions on Computers*, volume 49 of *IEEE Transactions on Computers*, pages 967–970. IEEE Computer Society, 2000.

[200] H. Yoo, C. Kim, J. Ha, S. Moon, and I. Park. Side Channel Cryptanalysis on SEED. In C. H. Lim and M. Yung, editors, *Information Security Applications, 5th International Workshop, WISA 2004, Jeju Island, Korea, August 23-25, 2004, Revised Selected Papers*, volume 3325 of *Lecture Notes in Computer Science*, pages 411–424. Springer, 2004.

[201] Y. Yu, Y. Yang, Y. Fan, and H. Min. Security Scheme for RFID Tag. Auto-ID Labs Fudan Univesity, White Paper, 2006.

[202] Y. Yu, Y. Yang, N. Yan, and H. Min. A Novel Design of Secure RFID Tag Baseband. In *RFID Convocation, Brussels, Belgium, March 14, 2007*, 2007.

[203] M. R. Z'aba, H. Raddum, M. Henricksen, and E. Dawson. Bit-Pattern Based Integral Attack. In K. Nyberg, editor, *Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Proceedings*, volume 5086 of *Lecture Notes in Computer Science*, pages 363–381. Springer, February 2008.

[204] Z. Zhu. RFID Analog Front End Design Tutorial (version 0.0), 2004. Available online at http://autoidlabs.eleceng.adelaide.edu.au/researchpapers.htm.

Author Index

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)