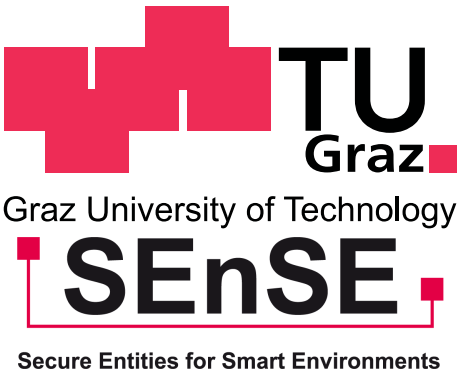
Security and Privacy in an RFID-based Electronic Payment System

Andreas Zankl
andreas.zankl@student.tugraz.at

Master's Thesis with the Collaboration of



Secure Entities for Smart Environments Devision

Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology
Inffeldgasse 16a
8010 Graz, Austria

Supervisor: Ao.Univ.-Prof. Dr. Karl-Christian Posch
Assessor: DI Thomas Korak, B.Sc.

Physical Analysis and Cryptographic Engineering Division

Temasek Laboratories @ NTU
Nanyang Technological University
Research Techno Plaza
50 Nanyang Drive
Singapore 637553

Supervisor: Asst. Prof. Dr. Axel York Poschmann

Graz, March 2014

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.

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 entnommenen Stellen als solche kenntlich gemacht habe.

....
(Date) (Signature)

Abstract

This thesis examines an electronic payment system deployed nationwide in the state of Singapore regarding system security and customer privacy. The focus is put on two RFID-based smart cards that are a fundamental part of the payment environment and widely used in public transport and everyday life: a single trip ticket and a long-term card. To further an open discussion of security and privacy in information technology, this system has been chosen to be analyzed.
In our work we discuss the secure usage of the single trip ticket and based on that demonstrate a denial of service attack which impairs the system's availability. In addition we show that the same ticket can be used to travel without paying the fare. Next, we assess an attack on the long-term card, which is already used in practice and with which comprehensive personal data can be retrieved without the consent of the card holder. Our contribution is to explain how and discuss to what extent the privacy of the customers can be affected by an adversary. Therefore, we present an experiment where user data was gathered over a period of six months and show techniques to infer sensitive personal information from the data. Furthermore we analyze the standard specifying the electronic purse implemented on the long-term card and evaluate the smart card's side-channel resistance with a Correlation Power Analysis attack on the electromagnetic emanation of the chip.
Finally, we discuss immediate and long-term solutions to the issues discovered during our work.

Keywords: RFID, contactless, ISO/IEC 14443, NFC, smart card, ticket, electronic payment, e-purse, electronic purse, cashless, public transport, fare, security, denial of service, fare evasion, privacy, movement tracking, behavior profiling, side-channel analysis, correlation power analysis, differential electromagnetic analysis, electromagnetic emanation

Kurzfassung

Diese Arbeit beschäftigt sich mit der Analyse eines im Staat Singapur weit verbreiteten, elektronischen Bezahlsystems im Hinblick auf Systemsicherheit und Privatsphäre der Kunden. Der Fokus liegt hierbei auf zwei RFID-basierten Smartcards, die ein wesentlicher Bestandteil des Bezahlsystems sind und die im öffentlichen Verkehr und alltäglichen Leben eine breite Anwendung finden: ein Einzelfahrtticket und eine Langzeitkarte. Das System wurde deshalb zur Analyse ausgewählt, um einen offenen Umgang mit dem Thema Sicherheit und Privatsphäre in der IT weiter zu fördern.
In unserer Arbeit diskutieren wir die sichere Verwendung des Einzelfahrttickets und stellen darauf basierend einen Denial of Service Angriff vor, der die Verfügbarkeit des Bezahlsystems beeinträchtigt. Darüber hinaus zeigen wir, dass das gleiche Ticket für Fahrten genutzt werden kann, ohne dafür zu bezahlen. Als Nächstes beurteilen wir einen Angriff auf die Langzeitkarte, der bereits in der Praxis angewendet wird und durch den man umfassende persönliche Daten des Kartenbesitzers ohne dessen Zustimmung abgreifen kann. Unser Beitrag ist, zu erklären, wie ein Angreifer die Privatsphäre der Kunden beeinträchtigen kann, und zu diskutieren, in welchem Ausmaß dies möglich ist. Aus diesem Grund präsentieren wir ein 6-monatiges Experiment, im Zuge dessen Benutzerdaten gesammelt wurden, und zeigen Techniken, mit denen man sensible persönliche Informationen von den Daten ableiten kann. Darüber hinaus analysieren wir den Standard, der die elektronische Geldbörse der Langzeitkarte spezifiziert, und evaluieren die Seitenkanalresistenz der Smartcard mittels eines Correlation Power Analysis Angriffs auf die elektromagnetische Abstrahlung des Chips.
Abschließend diskutieren wir Kurz- sowie Langzeitlösungen für die im Laufe der Arbeit entdeckten Probleme.

Stichwörter: RFID, kontaktlos, ISO/IEC 14443, NFC, Smartcard, Ticket, elektronisches Bezahlsystem, e-purse, elektronische Geldbörse, bargeldlos, öffentlicher Transport, Fahrpreis, Sicherheit, Denial of Service, Schwarzfahren, Privatsphäre, Bewegungsprofil, Verhaltensanalyse, Seitenkanalanalyse, Correlation Power Analysis, differenzielle elektromagnetische Analyse, elektromagnetische Abstrahlung

Preface

This thesis is the result of a research visit at the Physical Analysis and Cryptographic Engineering (PACE) group, which is part of Temasek Laboratories @ NTU and located at Nanyang Technological University (NTU), Singapore.

Axel Poschmann, head of the PACE group and my main supervisor at NTU, promoted the idea of evaluating the security and privacy of EZ-Link, a nation-wide cashless payment system in Singapore that is extensively used in public transport. The focus of the research and consequently the results presented here mainly revolve around the contactless smart cards, since they are a substantial building block forming the link between the user and the system.

As the published results of a security evaluation can pose serious and immediate threats to a system, informing the manufacturer and other affected parties beforehand about discovered flaws should be part of any professional security researcher's ethical codex. Consequently, the results presented in this thesis have been communicated to the Land Transport Authority of Singapore (LTA) prior to publication.

Furthermore I want to point out that no confidential, no non-public information about the system was provided by EZ-Link Pte Ltd and that the work presented here is solely based on public information properly referenced throughout the thesis and on reverse engineering.

During these various reverse engineering tasks I had the opportunity to examine, understand and evaluate both hardware and software components, as it is strongly embraced in the Telematics master's program, which I am writing this thesis for. My local assessor, Thomas Korak, was closely involved in the entire process. He is a member of the Secure Entities for Smart Environments (SEnSE) group, which is part of the Institute for Applied Information Processing and Communications (IAIK) located at Graz University of Technology.

Acknowledgments

Many people have helped me with this thesis in various ways. First of all, I would like to thank Axel Poschmann for inviting me to Singapore to work within a group of great researchers. He got me interested in the topic, kept me motivated along the way as well as believed in me throughout my stay and beyond. During the project I have had valuable input from Hoon Wei Lim, who did great work on the subject before. Much credit also goes to Yu Wen Siah for the decapsulation of the smart card ICs and to Sebastian Kutzner, who helped me with everything that was going on in the laboratory. Together with Marc Stöttinger and Chien-Ning Chen, he was also of great help in getting settled in the new environment. A big thanks for many interesting discussions to Dirmanto Jap, Markku-Juhani O. Saarinen and Phuong Ha Nguyen and to all other researchers I have met at SPMS. It was an honor and great fun working with all of you at NTU.

Furthermore, I want to thank Thomas Korak for his enduring support dating back to the very first step I set into the IMPA Lab, for his invaluable input to this thesis and his patience while working together over a great distance. Thanks to Karl-Christian Posch for supporting my research stay abroad and for focusing me on the things that matter most.

Finally, I want to thank my family and friends that kept me going with their unconditional support and encouragement. A big thanks also to all that I have met along the way for their various contributions to this work.

Contents

List of Figures

List of Tables

Conventions

Numbers In this document the radix point is marked by a dot '.' whereas the thousands separator is marked by a comma ','. Binary representation of a number is denoted with a subscript b (e.g. 1010_b), a hexadecimal one with a subscript h (e.g. $\texttt{4f}_h$) and a decimal one with a subscript d (e.g. 10_d). If no subscript letter is given and unless stated otherwise, the number is in decimal representation.

Naming The term *EZ-Link* refers to the cashless payment system itself, whereas *EZ-Link Pte Ltd* is the name of the company behind it. *ez-link Card* refers to the core contactless smart card that comes in a variety of different models.

Date & Time Dates are not always given using the same format, but always in an unambiguous way, for instance 'Thu, 01 Jan 1970 00:00:00 GMT'. For the time of day the 24-hour notation is used.

Abbreviations

ASCII	American Standard Code for Information Interchange
ATR	Answer To Reset
AFI	Application Family Identifier
AID	Application Identifier
API	Application Programming Interface
APDU	Application Protocol Data Unit
ATM	Automated Teller Machine
CAN	Card Application Number
CID	Card Identifier
COS	Card Operating System
CSN	Card Serial Number
CBC	Cipher-Block-Chaining
CLA	Class Byte
(C)MOS	(Complementary) Metal Oxide Semiconductor
CT	Computed Tomography
CEPAS	Contactless e-Purse Application Specification
CPA	Correlation Power Analysis
CRC	Cyclic Redundancy Check
DES	Data Encryption Standard
DF	Dedicated File
DEMA	Differential Electromagnetic Analysis
DPA	Differential Power Analysis

EEPROM	Electrically Erasable Programmable Read-Only Memory
EM	electromagnetic...
EF	Elementary File
EMV	Europay, MasterCard and Visa
FID	File Identifier
GSM	Global System for Mobile Communications
HD	Hamming Distance
HW	Hamming Weight; Hardware
HF	High Frequency
HITS	Household Interview Travel Survey
ID	Identifier
IV	Initial Vector
INS	Instruction Byte
IC	Integrated Circuit
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
LTA	Land Transport Authority of Singapore
LRT	Light Rail Transit
MRT	Mass Rapid Transit
MAS	Monetary Authority of Singapore
NTU	Nanyang Technological University
NFC	Near Field Communication
NETS	Network for Electronic Transfers (Singapore) Pte Ltd
OTP	One-time Programmable
PICC	Proximity Integrated Circuit Card
PUPI	Pseudo-Unique PICC Identifier
RFID	Radio-Frequency Identification
RFU	Reserved For Future Use
SAM	Secure Access Module
SFI	Short File Identifier
SCA	Side-Channel Analysis
SPA	Simple Power Analysis
S$, SGD	Singapore Dollar
SW	Software
SVF	Stored Value Facility
SIM	Subscriber Identity Module
S-box	Substitution Box
TRP	Terminal Reference Parameter
3DES, TDEA	Triple Data Encryption Algorithm
UHF	Ultra High Frequency
UID	Unique Identifier

Used Symbols

V_{DD}	Positive Supply Voltage Identifier
V_{SS}	Negative Supply Voltage Identifier

Chapter 1

Introduction

In this chapter the reasons for making EZ-Link the topic of this thesis as well as a short outline of the content are given.

1.1 Motivation

According to the Land Transport Authority of Singapore an average of 7.097 million passenger trips were registered daily on Singapore's public transportation system in 2012, the overall population of Singapore being 5.312 million at that time. The journeys include the Mass Rapid Transit system (MRT), a mixture of metro and overhead railway, the Light Rail Transit system (LRT), a smaller railway system feeding the MRT, the bus system and taxis. [1]

One widely used way to pay for public transport trips are so-called Stored Value Facilities (SVF), defined by the Monetary Authority of Singapore (MAS) as a way of storing value (e.g. on a smart card) that can in turn be used to buy goods and services from merchants [2]. In 2012 a total of 3,015 million transactions worth 2,351 million Singapore Dollar (S$) were made in Singapore using SVFs. [3]

As of May 2012, there are three widely accepted SVFs approved by the MAS: the *NETS CashCard*, introduced by Network for Electronic Transfers (Singapore) Pte Ltd (NETS) in November 1996, the *NETS FlashPay*, introduced by NETS in October 2009, and the *ez-link Card*, introduced by EZ-Link Pte Ltd in April 2002. Two of those systems, the NETS FlashPay and the ez-link Card, are used in public transport and in its latest version follow the Contactless e-Purse Application Specification (CEPAS), a Singaporean standard. [4]

In a press release this year EZ-Link Pte Ltd stated that until February 2014 more than 17 million CEPAS compliant ez-link Cards have been issued [5].

Given the great number of commuters, the broad acceptance and large scale of the payment system itself and consequently the large sums of money involved, not only on the system but also on the user side (one ez-link Card can store up to S$ 500 [6]), the EZ-Link system caught our attention.

But as an outside observer, why analyze security and privacy aspects of a well-established system that has successfully been in use for such a long time without any major issues being reported? To answer that question, I would like to refer to an analogy Ross Anderson gave in 1993 about the safety of flying.

> *When an aircraft crashes, it is front page news. Teams of investigators rush to the scene, and the subsequent enquiries are conducted by experts from organisations with a wide range of interests - the carrier, the insurer, the manufacturer, the airline pilots' union, and the local aviation authority. Their findings are examined by journalists and politicians, discussed in pilots' messes, and passed on by flying instructors.* [7]

When big security systems fail, such an institutionalized and open learning mechanism is still not common practice, with bad publicity and fear of economic loss certainly being among possible reasons. However, drawing the curtain over severe incidents prevents the community from learning valuable lessons. And while flying got safer over time, grave security flaws can still be found in a lot of today's systems and products. This way, the average system will hardly get any better in practice.

To counter that development, security researchers have done their part in contributing to a more open discussion about security failures. By analyzing commercial products and reporting security issues, they provide valuable feedback to multiple parties. First, the public gets a third-party expert opinion about systems they put their trust and personal data in. Second, following the principle of responsible disclosure, which in short means giving affected parties enough time to react to discovered issues, manufacturers are given the opportunity to fix bugs before anyone can maliciously exploit them. This way, they are able to improve their products further. Many big companies have realized this and are now giving rewards to people that find significant security issues and choose to responsibly disclose them. Third, it's a contribution to the scientific community and eventually helps in making future products more secure to begin with.

In that sense security researchers continue to be a vital part of this crucial learning mechanism, especially since they can afford the luxury of having an impartial point of view, being bound only to their scientific values. This thesis shall be a contribution to that process.

1.2 Outline

The focus of this thesis lies on the contactless smart cards of the EZ-Link system, of which two types are covered in detail: the ez-link Card and the so-called Standard Ticket.

After giving some basic facts about the EZ-Link system and the contactless smart cards themselves in Section 3.1, about their underlying technology called Radio-Frequency Identification (RFID) in 3.2 as well as about the basics of smart cards in 3.3 and the principles of side-channel analysis (SCA) in 3.4, including differential electromagnetic analysis (DEMA), the Standard Ticket and the ez-link Card will be separately assessed regarding security and privacy aspects in Sections 4 and 5 respectively. In both cases this starts with an analysis of the cards' underlying chips followed by their properties as well as usage in the system and leads to a discussion of security and privacy related concerns.

Due to its simple architecture and accessible documentation, the smart card hosting a Standard Ticket is thoroughly examined in 4.2 whereas its usage in the system is discussed in 4.3. After assessing privacy concerns in 4.4, the security evaluation in Section 4.5 describes a simple yet effective denial of service attack and also demonstrates how a Standard Ticket can be used to travel without paying the fare. After possible solutions are discussed, the chapter ends with a short conclusion in 4.6.

In order to analyze the more access-restricted smart card platform of the ez-link Card, the CEPAS standard is explained in 5.3 together with an analysis of the actual implementation found on an ez-link Card. A refinement of the attack presented by Kerschbaum et al. in [8], which is already in use in practice, is shortly explained in that section as well, since to the author's knowledge it has neither been published in scientific literature nor discussed in a security and privacy related context yet. In order to illustrate the impact of the refined attack, the results of its practical execution in the course of a long-term experiment are presented in Section 5.4 together with potential privacy impairments of customers and corresponding solutions. The security evaluation in 5.5 starts with a quick assessment of the CEPAS standard, which is then used to develop a side-channel analysis scenario. The measurements of the ez-link Card's electromagnetic emanation and the results of the correlation power analysis attacks performed on them are presented before concluding in 5.6.

A summary of the results presented in this thesis is given in Chapter 6 together with topics for future research.

Chapter 2

Research

This chapter gives an overview of research done in related fields and on similar topics. After that the important definitions of the terms *security* and *privacy* are stated together with the goals of this thesis and a short summary of its contribution to the field of study.

2.1 Related Work

At its core EZ-Link provides means of electronic, cashless payment. Together with its use of contactless smart cards and its broad acceptance in public transport it is among many other similar systems around the world. From a security perspective, much depends on the smart cards themselves and how they are used, as literature shows.

Smart Cards One well-documented case of a smart card security vulnerability directly affecting electronic payment systems is the MIFARE Classic, a contactless smart card by NXP Semiconductors.

After the inner workings of its cryptographic algorithm CRYPTO-1 were revealed in 2007/08 by Nohl et al. [9, 10], a group of Dutch researchers led the platform's security downfall (e.g. de Koning Gans et al. [11, 12], Garcia et al. [13, 14]) and nowadays multiple practical attacks exist with a well roundup given by Tan [15]. With the published attacks, MIFARE Classic based payment systems cannot only be more thoroughly analyzed as shown with the Oyster card in London by Tan [15], but also essentially broken as demonstrated by Welte [16] with the EasyCard in Taipei or by Kasper et al. [17] with a German payment system.

Incidents also revolve around other platforms. In [18] Bono et al. present attacks on the Digital Signature Transponder (DST) by Texas Instruments, challenging the security of payment systems like the Exxon-Mobil SpeedPass that make use of the DST. In [19] Verdult demonstrates cloning of a disposable ticket for the OV-Chipkaart System in Rotterdam and how this enables unlimited free rides. The disposable tickets are MIFARE Ultralight based, a contactless smart card at the simpler end of the MIFARE series. However, the vulnerabilities lie in the system's usage of the cards rather than in the cards themselves.

Platforms that offer more security features have also been analyzed in literature. Successful side-channel attacks on the MIFARE DESFire (MF3ICD40) used for instance in San Francisco as the Clippercard are published by Oswald et al. in [20], while a low-cost emulation device for the very same chip is given by Kasper et al. in [21].

Attacks on the DESFire make use of exploiting its electromagnetic (EM) radiation, illustrating the potential of EM analysis of contactless smart cards. This topic is also covered by Hutter et al. [22, 23], Kasper et al. [24] and Korak et al. [25].

Payment Systems One of the early adopters of electronic payment with contactless smart cards is the Octopus system in Hong Kong introduced in 1997 to the general public [26]. Its development and acceptance has been studied since, e.g. Bailey et al. [27]. Privacy aspects have been covered by Greenleaf et al. [28] and Chung [29], including an incident where personal data of users was transfered to third parties for marketing purposes. However, only a project report from the University of British Columbia by Lee at al. [30] gives a first insight specifically into security aspects of the Octopus system.

Regarding the EZ-Link system, implementation facts and basic architectural aspects are covered by Sim et al. [31] and Yang [32]. Identification of an information leak in the EZ-Link system and its privacy implications are presented together with an improved system architecture by Kerschbaum et al. [8]. Information on the security aspects of the EZ-Link system has to the author's knowledge not been published so far, a gap this thesis tries to close in a first approach.

2.2 Problem Definition

The goal of the work presented in this thesis is to get insight in security and privacy aspects of the EZ-Link system. But before defining goals and posing further questions, the terms *security* and *privacy* and what is meant by them in this thesis must be defined more precisely.

Privacy The right to privacy is a fundamental human right defined in article 12 of the Universal Declaration of Human Rights.

> Article 12: *No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.* [33]

But while the right to privacy is universal, a general definition of the term itself is more difficult to agree upon. Nevertheless, this thesis will follow a statement given in 1967 by Alan Westin in his work *Privacy and Freedom*.

> *Privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others.* [34]

The important point is that individuals are aware and in control of a selective exposure of their personal information, a concept that forms the very contrast to the *arbitrary interference* mentioned in article 12 above and is therefore worth pursuing. Facets of an information technology system that could adversely affect this concept shall be considered during the evaluation process and are simply referred to as privacy aspects.

Security The strong dependence on the context it is used in makes it extremely difficult to give a precise definition of the term security. The approach used in this thesis is borrowed from the field of security engineering, where systems can be designed in a three step process: threats are identified, policies are formulated to mitigate them and mechanisms eventually implement this mitigation. The meaning of security is then defined by the policies, which state what is and what is not allowed in the system. If a policy can be violated despite the mechanisms enforcing it, a threat is introduced to the system and the security is said to have been breached.

As an outside observer, detailed policies of a closed source system are difficult to reveal. However, the sole purpose of a system usually defines a set of simple policies that can be used to assess an observation regarding possible violations. If for instance a service that usually costs money can be used for free, this can be classified as a security breach without the knowledge of any internal policies. Similar, an impaired service availability falls into this category.

Therefore, security aspects as referred to in this thesis are facets of a system which can be used to assess the proper enforcement of system policies.

Goals This thesis shall be a first step in giving a clear picture of EZ-Link's strengths and weaknesses regarding security and privacy aspects. It is intended to be a baseline for future research work. The focus is put on the contactless smart cards involved. They are the starting point of our investigations and consequently lead to other research directions. At the beginning of our work, we asked ourselves the following broad questions and in the course of the project tried to answer them as well as enhance them with more precise ones pointing towards the most promising directions.

- What types of cards are used in the system and what are their properties?
- How are the cards used in the system?
- What can be said about security and privacy aspects?
- What are promising future research directions?

2.3 Our Contribution

In our work we identify the type and properties of the Standard Ticket's underlying chip and thoroughly assess its features and usage in the system. Based on that we present a simple yet effective denial of service attack that impairs the system's availability. In addition we show how an attacker can use the ticket for fare evasion and discuss possible solutions for both attacks.

Next, we give insight in the CEPAS standard that specifies the electronic purse stored on an ez-link Card and based on that demonstrate how the requirements of the attack published by Kerschbaum et al. [8] can be lowered, thus making it easy to execute in practice. This refined attack has already been implemented by the open source Android application FareBot. However, it is important to point out that no malicious activity of this application has been observed by the author and that the project is advertised to provide public transit card holders with detailed information only about their own card [35, 36]. Since to the author's knowledge the refined attack has neither been discussed in a security and privacy context nor published in reviewed scientific literature yet, we

illustrate how it works, discuss to what extent it impairs the privacy of EZ-Link customers and assess its practical impact. For that purpose, user data has been gathered from an ez-link Card in the course of an experiment over the period of six months. Using the data set we demonstrate analysis techniques and show how they can be used to infer sensitive personal information about the card holder.

Furthermore, we develop a practice-oriented side-channel analysis scenario based on the CEPAS standard that facilitates an easy measurement process and provides a theoretical way of recovering all cryptographic keys stored on an ez-link Card. The proof-of-concept first-order correlation power analysis performed on the electromagnetic emanation of the chip demonstrates the feasibility of the scenario.

Except for this proof-of-concept side-channel analysis attack, we limited ourselves to low-cost, off-the-shelf equipment to implement all presented attacks. By keeping the hardware and software requirements to a minimum and implementing attacks with an Android application running on a Google Nexus S smartphone, we demonstrate that our attacks are practical and easy to execute.

In the course of the thesis, we discuss possible short- and long-term solutions for all discovered security and privacy issues.

Chapter 3

Background

This chapter contains an introduction to EZ-Link, the contactless smart cards that are part of the system and their underlying technology called Radio-Frequency Identification. After that the basic properties of a smart card are discussed as well. Furthermore the field of side-channel analysis is introduced and the topics about Differential Power Analysis and about the Data Encryption Standard are covered in more detail.

3.1 EZ-Link

EZ-Link is a Singapore-wide electronic payment system introduced in 2002 by EZ-Link Pte Ltd, a subsidiary of the Land Transport Authority of Singapore (LTA). It is one of the widely accepted Stored Value Facilities as approved by the Monetary Authority of Singapore. In the case of EZ-Link this means that money can be stored electronically on a smart card and later be deducted at merchants in exchange for services and goods. A third party referred to as holder is responsible for holding the actual money in the meantime and paying the merchant after the purchase, since the card itself only stores a digital value representing an amount of money. [2, 4]

EZ-Link payments can be used for public transport (MRT, LRT, bus and taxi), Electronic Road Pricing as well as Electronic Parking System. They are also accepted at selected governmental and educational institutions as well as retail stores. A complete list can be found under [37]. EZ-Link Pte Ltd, the company behind the EZ-Link brand, is an important part of Singapore's public transport solution, for which several players team up. The Public Transport Council and the Land Transport Authority of Singapore are involved government authorities, SBS Transit Ltd and SMRT Corporation Ltd are transport operators and EZ-Link Pte Ltd as well as NETS Pte Ltd are card managers. Transit Link Pet Ltd, also a subsidiary of the LTA, is a service provider that operates throughout this constellation of different parties. They process transit transactions and provide card sales, refunds, replacements and top-ups of CEPAS-compliant cards. [38]

EZ-Link Cards can be re-charged in MRT and LRT stations, in post offices, in retail stores and at selected ATMs [40]. There are three additional ways to *top up*, which means to reload, the card. With *EZ-Online* a card can be topped up anywhere with a card reader connected to a computer using traditional online payment [41]. With *Top and Tap* the same online payment is done, but the amount can be transferred to the card at a later point in time and also without an own card reader [42]. *EZ-Reload* automatically reloads a card when the value stored is insufficient for payment and in turn charges a previously

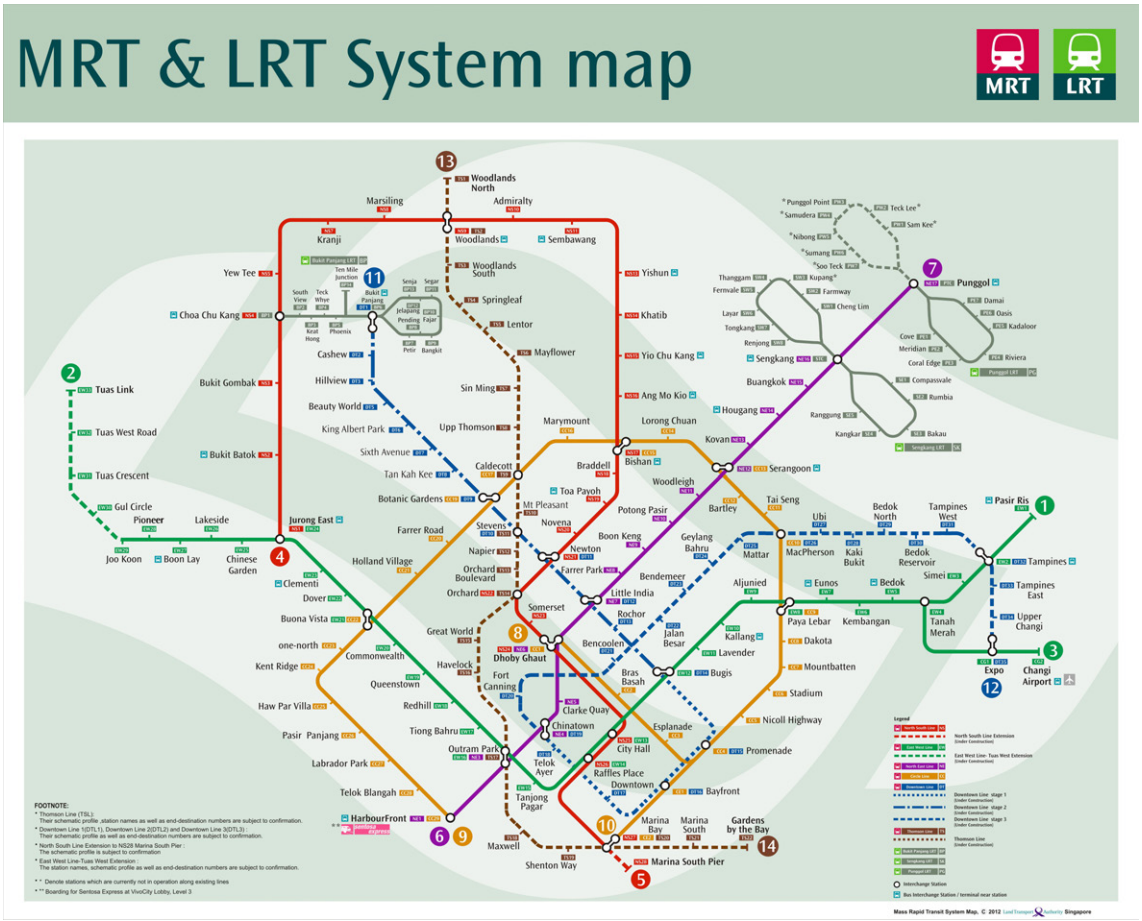


Figure 3.1: Singapore MRT and LRT Map (incl. development plans) [39].

deposited giro account or credit card [43]. For an overview of Singapore's MRT and LRT system, Figure 3.1 shows current (solid) and future (dashed) lines.

3.1.1 Cards

As diverse the top up possibilities are, as manifold is the card portfolio. The *EZ-Link Season Pass*, personalized in the form of an identity card, allows unlimited travel on MRT, LRT and buses in a selected period of time; a concept the non-personalized *Singapore Tourist Pass* follows as well, however with different payment models [44, 45]. In addition to being an ordinary ez-link Card, the *PAssion ez-link Card* is a membership card for the Peoples Association [46], *"a statutory board to promote racial harmony and social cohesion in Singapore"* [47]. Other than that, there are some co-branded cards with credit card companies and the possibility to use an NFC-enabled smartphone as an ez-link Card. In this case the electronic purse is typically stored on the SIM card in the device and information like account balance or transaction history can be directly accessed with the EZ-Link Mobile App. With this solution no actual card has to be carried around. [48]

Apart from these choices, the focus in this thesis is put on the very basic cards available to be used for public transport. The first is the so-called *Standard Ticket*, a limited-use paper ticket for single trips when traveling with MRT or LRT. The second is the *ez-link Card* in its non-personalized form and with its core application: the electronic purse.

What all of the available cards have in common, is their contactless interface to the outside world, enabling *tap-and-go*. It's sufficient to just hold a card in proximity to the reader without the necessity of ever touching it. This way it is possible to leave a card in the wallet while making payments. The technology this procedure is based on is generally summarized under the broad term Radio-Frequency Identification or RFID.

3.2 Radio-Frequency Identification

In simple terms RFID allows quick identification of objects over a wireless communication channel. The device identifying the object is usually referred to as reader or interrogator whereas the object itself is commonly called tag or transponder. What is common to all RFID systems is that reader and tag are communicating without the necessity of having physical contact. Over this wireless path, data can be exchanged, clock and energy provided. Usually reader and tag are part of a bigger system with an application that interfaces the RFID sub-system.

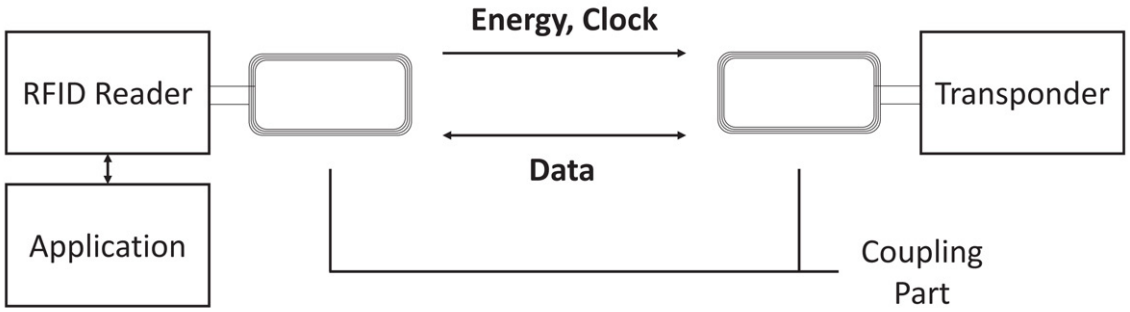


Figure 3.2: Basic RFID Setup [49].

Other than that there is a big diversity of systems and many properties to differentiate between them. Starting from a physical perspective, signal frequencies range from low frequency (LF) 30...300 kHz, over high frequency (HF) 3...30 MHz and ultra high frequency (UHF) 0.3...3 GHz up to microwave >3 GHz bands. Communication distances can be separated in close coupling (<1 cm), remote coupling (0.01...1 m) and long-range (>1 m). Linked to frequency and operating distance, electric, magnetic or electromagnetic fields are used with reflection resp. backscatter, load-modulation or (sub-)harmonics transmission techniques to couple reader and tag. Apart from the physical communication, tags differ in many properties. They are micro-controller or state machine based, with or without own power supply and communicate full-duplex, half-duplex or sequentially with different data rates. In addition they come in a dazzling array of different form factors and their diverse set of features regarding memory and computing power just reflect the ever-growing number of various applications they are used in. [49]

3.2.1 ISO/IEC 14443

The tags used in the EZ-Link system come in an ID-1 form factor, which is well-known from banking or identity cards and defined in ISO/IEC 7810 [50]. Hence, the tags are also referred to as cards. Both cards analyzed in this thesis operate in the HF band at 13.56 MHz and have an operating range of a few centimeters. From the physical charac-

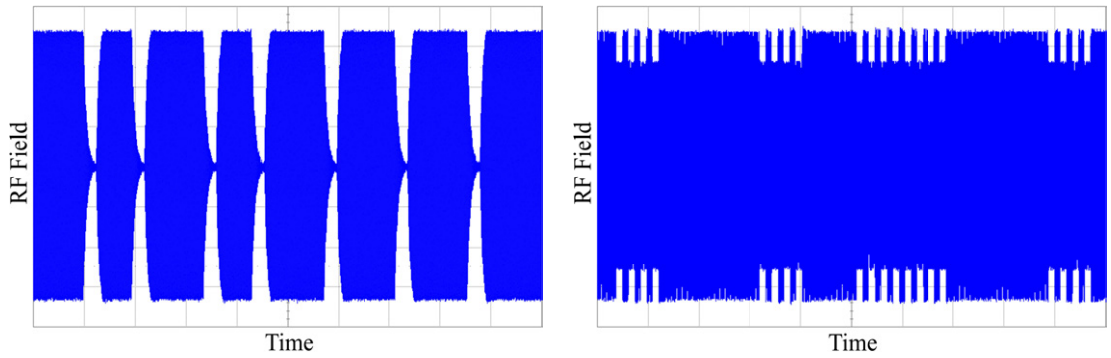
teristics up to the protocol level, they follow the ISO/IEC 14443 standard, which consists of four parts and defines two types of communication interfaces (Type A and B). Given the limited operating distance, the standard speaks of proximity cards.

- Part 1: *Physical Characteristics*, includes mechanical, electric, magnetic and temperature related stress limits [51]
- Part 2: *Radio frequency power and signal interface*, includes digital transmission properties [52]
- Part 3: *Initialization and anticollision*, includes basic commands and protocol properties [53]
- Part 4: *Transmission protocol*, includes support for high-level interfaces [54]

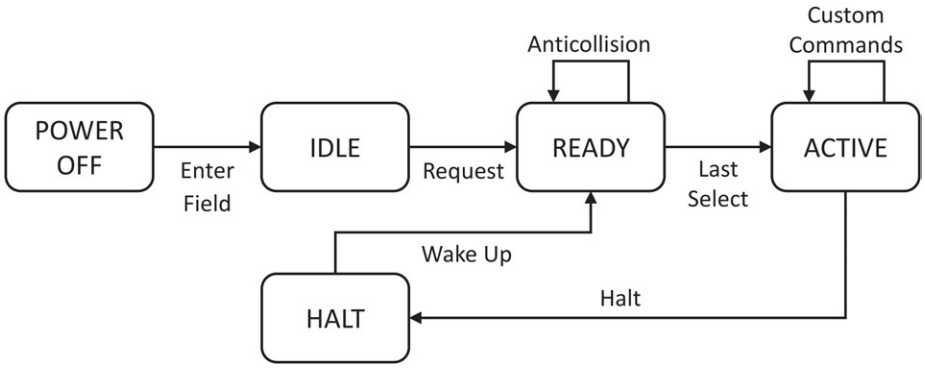
Type A The Standard Ticket follows these standards up to ISO/IEC 14443-3 and implements the Type A communication interface. In general the reader supplies the tag with energy and clock via a 13.56 MHz energizing radio-frequency field. To transfer data, the reader uses an Amplitude-shift keying (ASK) modulation scheme with a modulation index of 100 % and a Modified Miller encoding. The tag responds with a Manchester encoded on-off-keying load modulation on a subcarrier frequency of approximately 847 kHz. Initially, both devices transmit with a data rate of approximately 106 kbit/s, but higher data rates are possible as well. Figure 3.3 shows parts of a message exchange between reader and tag. The ASK 100 % scheme can be clearly seen in 3.3a, where a Request command is shown. The much weaker back modulation of the tag shows only little drops of the signal in 3.3b, where parts of the corresponding Answer-To-Request are illustrated. [52]

Each device that follows ISO/IEC 14443-3 Type A must have a so-called unique identifier or UID with 4, 7 or 10 bytes. This identifier is used to distinguish different tags and can be static or randomly generated on the fly. The standard also defines a set of device states and commands to traverse them, which are illustrated in Figure 3.3c. [53]

When the tag enters the field of the reader and draws enough energy, it powers itself up and within 5 ms enters the IDLE state, where it is able to receive commands. With a Request command it moves on to the READY state, which is used to perform the so-called Anticollision sequence. This is a procedure during which the reader can select one tag out of many that are currently present in its field. The basic principle behind it is that the reader sends a bit pattern and each tag that can match this pattern with its UID responds to the command. Since multiple responses can be detected, the reader extends the pattern until only one tag is left. Depending on the size of the UID this procedure is split up into at most three separate sequences, each covering a different part of the UID. A new sequence is initiated with a Select command. The further the Anticollision-Select sequences advance, the less tags can match the pattern and eventually only one is left. After the last Select, the chosen tag is in ACTIVE state and typically proprietary commands take over. If the tag is not needed anymore, it can be sent to HALT state with the correspondent command. In this state the tag ignores all further commands except for a Wake Up, after which the tag goes into READY state and can be selected again. Note that for better understanding this is a simplified version of what is specified in the standard. [53]



(a) Reader Command. (b) Tag Answer.



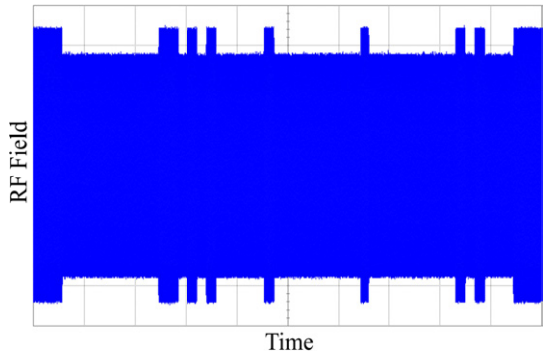
(c) Simplified States and Commands [53].

Figure 3.3: ISO/IEC 14443-2 Type A Communication.

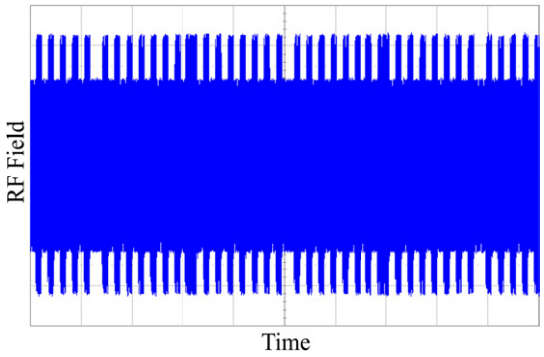
Type B The ez-link Card follows the standards up to ISO/IEC 14443-4 and implements the communication interface Type B. The difference to Type A is that the reader uses a modulation index of only 10 % and a Non-return-to-zero Level encoding (NRZ-L). The tag responds with load modulation on the same subcarrier frequency as Type A, but also uses an NRZ-L encoding together with a Binary phase-shift-keying scheme. Initially, the data rate is again around 106 kbit/s and higher data rates are possible. Figure 3.4 shows parts of a Type B communication between reader and tag. It can be clearly seen that the reader field in 3.4a, showing parts of a Request command, only drops a fraction of Type A. The different encoding of the tag can also be seen in 3.4b, where parts of the Answer-To-Request are illustrated. [52]

The communication interface of Type B shown in Figure 3.4c revolves around the same states as Type A, but other than that is quite different from it. After the tag is powered up, it listens for commands in IDLE state. The reader issues a Request command, which is already more powerful than Type A. It includes an Application Family Identifier (AFI), which can be used to limit communication only to a certain type of tags (e.g. transportation tickets). It also includes a parameter that defines if the command is an ordinary Request or also a Wake Up. Furthermore the Request implements a different approach to the Anticollision known from Type A. Here, tags can choose a time slot from a set the reader provides to send an answer in. If the AFI matches and a slot has been found, the tag responds with an answer including another identifier, the Pseudo-Unique PICC Identifier (PUPI), and proceeds to the READY state. Here, the reader can put the tag into HALT state or advance it to ACTIVE state with an Attrib command, if

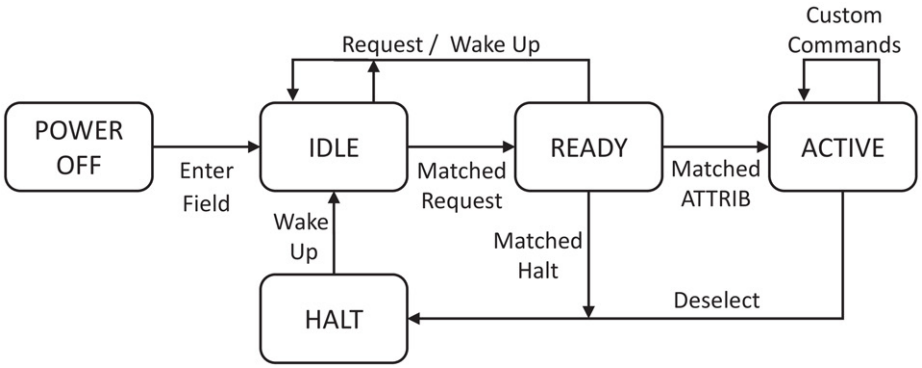
it uses the previously transmitted PUPI. In the Attrib command the reader assigns the tag yet another identifier, the Card Identifier (CID), in order to distinguish all currently active tags. From the ACTIVE state the reader can either send the tag to HALT state with a Deselect command or proceed with custom commands. Again, note that this is a simplified version and that further details can be found in the standard. [53]



(a) Reader Command.



(b) Tag Answer.



(c) Simplified States and Commands [53].

Figure 3.4: ISO/IEC 14443-2 Type B Communication.

Even though entity identification is a built-in objective in RFID and the ISO/IEC 14443 standard, the functionality of many cards goes well beyond it. For this reason, the Type A and B interfaces defined in the standard are used as a transportation layer for platform specific communication, similar to any wired interface. This happens from the ACTIVE states onwards and can be proprietary or follow yet another standard. Consequently, feature-rich applications that sit on top of the communication stack defined in ISO/IEC 14443 can take advantage of the contactless nature of RFID. In this scenario the built-in identification mechanism is often just a useful side-effect. Popular examples for such applications include personal identity cards as well as access control, electronic ticketing, payment and loyalty systems. Due to their contactless interface, their ID-1 form factor and rich set of functionalities, such cards are usually referred to as *contactless smart cards*.

From ISO/IEC 14443-3 onwards the Standard Ticket implements originally proprietary commands that have by now been partially included in the Near Field Communication (NFC) standards. The ez-link Card uses a different approach and additionally follows ISO/IEC 14443-4, which is a framework to set up another communication channel on top of layer 3. It uses this option to encapsulate connections to yet another standardized

interface according to ISO/IEC 7816, which defines contact-based electronic identification cards and is widely adopted when it comes to smart cards. One benefit of this approach are so-called dual-interface cards that offer both contact-based and contactless interfaces at the same time.

NFC A set of standards that incorporates ISO/IEC 14443 is well-known under the name of Near Field Communication. It is defined in ISO/IEC 18092 [55] and ISO/IEC 21481 [56] and has been gaining support on various platforms, one of which is the operating system Android. With the integration in modern smartphones and tablets, it is nowadays more convenient than ever to access and use NFC-compliant cards.

3.3 Smart Cards

Contactless interfaces as discussed in the previous section were only an addition to the already established and contact-based concept of smart cards. The early days of smart cards date back to as far as 1968, where German inventors Jürgen Dethloff and Helmut Grötrupp registered the idea of enhancing an identification card with an integration circuit as a patent. In the 1970s the rapid advances in microelectronics made it possible for the semiconductor industry to supply integrated circuits with nonvolatile memory and processing logic on a single silicon die at acceptable prices. Since many inventions regarding smart cards were made in Germany and France, a first big step ahead was made in 1984 when the French postal and telecommunication services authority successfully used electronic telephone cards in a field trial. This success story continued and is today more prominent than ever, also thanks to the introduction of smart cards in the Global System for Mobile Communications (GSM) network. This worldwide communication network with billions of subscribers is one of the most well-known applications of smart cards, which in this context are usually referred to as Subscriber Identity Module or SIM cards. [57]

But also for the banking industry smart cards have proven to be an ideal medium. With the growing need to provide a more secure alternative to magnetic stripes, with the uprising of modern cryptography, crossing from military over to civil applications, and with continuously advancing electronic data processing capabilities, smart cards had the potential to fulfill all requirements regarding security and processing complexity while being small and easy to handle in everyday life. This manifested itself in the EMV and later EMVCo specifications (original efforts by **E**uropay, **M**asterCard and **V**isa) ensuring international compatibility of nowadays hundreds of millions of smart cards used for banking and payment. With their familiar looking ID-1 form factor, such banking cards range also among the most popular applications of smart cards. [57]

3.3.1 Properties

Types Apart from their contact-based or contactless communication interface, smart cards are also distinguished by their chip complexity. *Memory cards* usually focus on storing data and therefore offer a limited processing logic that is used to access non-volatile memory, often an EEPROM. This includes simple write protection mechanisms as well as more sophisticated security logic performing cryptographic algorithms for authentication purposes. Memory cards are typically tailored to a particular application and are thus limited regarding their flexibility but also rather inexpensive. The EZ-Link Standard Ticket falls into this category. *Processor cards* feature a central processing unit (CPU)

that is typically accompanied by a ROM unit, where the operating system resides, a RAM unit, which is the CPU's working memory, and an EEPROM unit, where program code and data is read and written. Modern operating systems can be customized to suit particular needs and support the secure co-existence of multiple applications on one card. The greater flexibility and the general purpose design sometimes has to be complemented by specialized co-processors that increase the efficiency of particular tasks. A common choice are cryptographic co-processors that are additionally hardened to maintain a good level of security. The ez-link Card is such a processor card. [57]

COS Similar to other general purpose computing systems like desktop PCs or smartphones, processor cards typically run an operating system as well. But unlike Windows- or UNIX-based systems, size and complexity of operating systems on smart cards are very limited. Due to strong requirements arising from this limited environment, the so-called card operating systems (COS) highly optimize their usage of hardware components, which is why achieving a decent level of hardware abstraction and consequently a hardware-independent software design is a challenging task. Together with a large set of different hardware platforms and extremely diverse requirements stemming from target applications, this is one reason for the big diversity of different COS available on the market. [57]

Card operating systems can range from a few kilobytes and a few thousand lines of code written in assembly language to hundreds of kilobytes and a few hundred thousand lines of code written in C. They work with simple 8-bit processors as well as with powerful 32-bit processors and even support complex programs like web-servers or interpreters for downloadable code. Nowadays such operating systems typically have a layer-based design in contrast to early library-based or monolithic approaches. What all of them have in common are high standards regarding software quality, since correcting defects in a shipped operating system is usually very expensive, for instance causing a recall campaign. This is why a COS has to be reliable and robust under any circumstances. Besides hardware abstraction and program execution, the main tasks of a COS include data transmission to and from the smart card, external command execution and file management. [57]

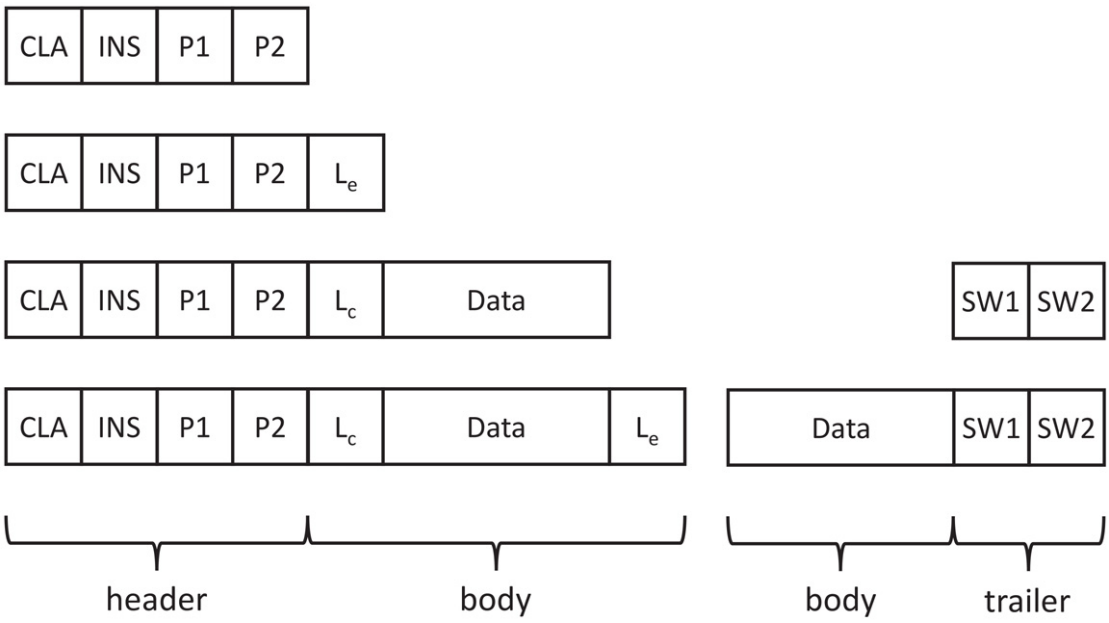
Communication When powered up, smart cards listen to commands from the outside world in a typical master-slave relationship, which means they never proactively send data without an external stimulus. When a command is received, the required task is performed and a response is sent, after which the smart card waits for the next command. If ISO/IEC 7816-3 is followed, the first response smart cards send is not triggered by an actual command but rather by the power-on reset sequence of the card. It is called Answer to Reset or ATR and contains information about the card and parameters for data transmission. It has a maximum length of 33 bytes, which is often not fully used if the card has to be ready as soon as possible for further commands. The elements of an ATR are defined in ISO/IEC 7816-3 and listed in Table 3.1. [57]

The initial character is mandatory for every ATR and specifies one out of two possible data transmission conventions, *direct* and *inverse*. According to direct convention (coded as TS = $\texttt{3b}_h$), a bit value of '1' is transmitted using the logic high voltage level, whereas a '0' uses logic low. Furthermore the least-significant bit is transmitted first. Inverse convention (coded as TS = $\texttt{3f}_h$) is the exact opposite. The format character specifies which of the interface bytes TA_1, TB_1, TC_1, TD_1 are given and the number of historical bytes K. The interface bytes specify remaining transmission parameters and can be cascaded

Element	Description
TS	Initial byte
T0	Format byte
TA_1, TB_1, TC_1, ...	Interface bytes
T_1, T_2, ..., T_k	Historical bytes
T_{ck}	Check byte

Table 3.1: Elements of an Answer to Reset [57].

if more space is required, yielding TA_i, TB_i, TC_i, TD_i with $i = 1, 2, \ldots$. Unlike the previous characters, the coding of historical bytes largely depends on the operating system, but often an ASCII representation is used and software as well as hardware component identifiers or versions are included. The check character is a simple checksum calculated by taking the XOR of the previous bytes starting from T0, but depending on the currently used protocol may or may not be present. [57, 58]



(a) Command APDU. (b) Response APDU.

Figure 3.5: Application Protocol Data Unit Types [57].

After the smart card has been powered up, it listens for commands. If the card implements ISO/IEC 7816-4, a standardized command format can be used that sits on top of the transmission layer and is independent of the current transmission protocol. This approach is especially beneficial, if multiple interfaces are used to talk to a smart card, since the commands on application level remain unchanged. The data packets that are exchanged between terminal and smart card are called Application Protocol Data Units or APDUs as shown in Figure 3.5. There are different formats of variable length for commands and responses. [59]

Command APDUs start with a class byte (CLA) used to distinguish different applications and their corresponding command sets. Typical examples are $\texttt{a0}_h$ for GSM applications or $\texttt{8X}_h/\texttt{9X}_h$ for proprietary use. The instruction byte (INS) codes the actual

command and is followed by two parameter bytes (P1, P2), which can provide additional information needed by the smart card to execute the task. While these bytes belong to the header and are always transmitted, the command body can be constructed in four different ways depicted in Figure 3.5a. The field L_e specifies the length of the response the terminal is expecting, where a value of 00_h denotes all data available. If the terminal needs to transmit more data to the smart card, it can extend the command APDU with a variable-length data block that is preceded by L_c, which specifies the size of the block. As illustrated in Figure 3.5b, Response APDUs optionally start with a general-purpose data block followed by two mandatory status bytes (SW1, SW2), also called return code. This status word contains the result of the command execution and in case of an error can indicate what exactly went wrong. However, the exact meaning of the return codes may vary from application to application, only 9000_h almost always indicates a successful command execution. [57, 59]

A set of common commands is defined in ISO/IEC 7816-4 and complemented by more application-specific ones in other specifications and standards. ISO/IEC 7816-8 for instance contains commands for configuration and execution of cryptographic functions, the specifications TS 51.011 as well as TS 51.014 contain commands for the telecommunication sector and SIM cards, and the EMV specification defines commands for financial applications. [57]

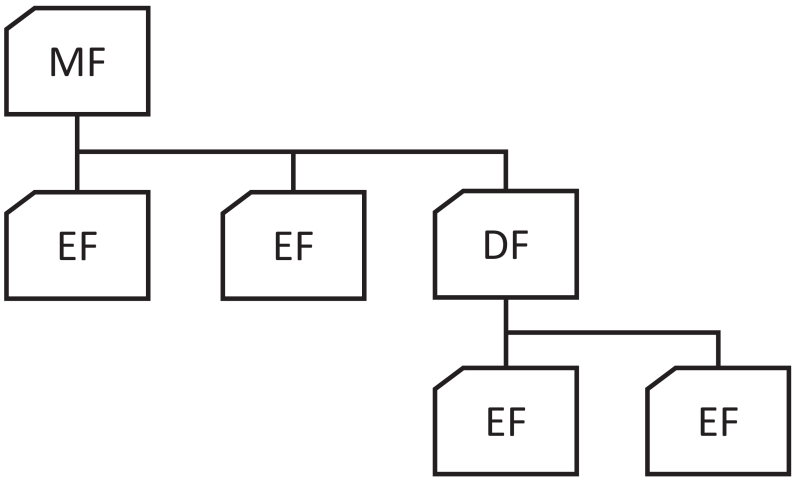


Figure 3.6: Smart Card File Hierarchy [57].

Files One of the main purposes of smart cards is their ability to store data through a well-defined interface. As specified in ISO/IEC 7816-4, most modern card operating systems implement this mechanism as a hierarchically structured file management system that supports hardware-independent addressing. Files are usually split up into a header, which contains access rights and file properties, and a body, which contains the actual data. This ensures safer and more flexible handling of stored data. However, file management is often very limited, especially after the smart card has been issued. [57]

There are basically two different file types. Elementary files (EF) actually contain user data, whereas dedicated files (DF) act as directories and list other EFs or DFs that logically belong together. Thus, starting from the root DF, also called master file (MF), a hierarchical structure is created as depicted in 3.6. The MF addresses the entire smart card storage memory and must be present on any card. Additionally, EFs can be marked

as operating-system-only, so-called internal elementary files, or accessible from the outside, called working elementary files. Every file is identified by a 2-byte file identifier (FID), which is used to select it in order to operate on it. For instance, the FID $\texttt{3f00}_h$ is reserved for the MF. Elementary files may additionally have a 5-bit short file identifier (SFI). Explicitly selecting a file means issuing a SELECT command with the FID before sending the actual command that operates on the selected file. In contrast to that, certain commands support implicit selection with the SFI being passed on as a command parameter, but this is limited to files in the current or root directory. If only a few files are needed, the implicit approach can provide a significant speed-up. [57]

3.3.2 Payment Systems

Performing payment transactions is one of the popular applications for smart cards. In contrast to traditional cash, smart cards have a few advantages that make them appealing in everyday life. They are small, light, easy to use and eliminate the need to constantly carry big amounts of cash, which is especially useful when traveling abroad. Also, they potentially reduce time spent at the checkout counter. Due to their electronic capabilities, they introduce new and useful features like contactless interfaces as well as payment authorization and better protection from theft. There are three basic concepts for using smart cards in a payment system. [57]

Credit cards are used to pay for goods and services with the amount being deducted from the account well after the purchase has happened. This is referred to as *pay later*. For this process the merchant typically pays a fee of a few percent of the purchase amount. Since smart cards fulfill stricter security requirements, they are a suitable replacement of the long outdated but still used magnetic stripes. When paying with a debit card, the payment is typically authorized by a background system right away and the amount is immediately deducted from the bank account, although there may be a threshold under which no online checks are performed. This scheme is referred to as *pay now*. The third concept is an electronic purse. While in the other two scenarios smart cards were basically used to authorize and trigger transactions in the background, e-purses actually store a virtual representation of real money. They are loaded before a transaction is made, which means that the electronic balance is increased in exchange for real money before buying any goods. At the merchant, the customer's balance is decreased and the merchant's one increased. After some time the merchant can exchange the virtual amount for real money again at the operator of the system. Thus, this concept is referred to as *pay before*. [57]

Similar to credit and debit cards, the operator must be carefully chosen and the background system properly designed. A big incentive to become an operator is the fact that by paying real money in exchange for a digital value, users actually give the operator an interest-free loan until the merchant reclaims the money again. In large-scale e-purse systems, the accumulated interest for the operator is significant whereas the loss for a single user is small but not always negligible and may thus be considered as a hidden fee. Furthermore, if the operator goes bankrupt and no further precautions were taken, users and merchants loose the value currently stored in the e-purses. Also, the scenario of a defective smart card has to be addressed properly, since otherwise the value on a broken card is lost as well. Because of these reasons, usually banks or similar institutions are chosen as operators and some effort has to be put into the design of the system. [57]

3.4 Side-Channel Analysis

The field of side-channel analysis revolves around the execution of algorithms. From a generic point of view, an algorithm takes an input and produces some output, which is illustrated in Figure 3.7. But unlike in theory, an algorithm that is implemented and executed in practice introduces additional challenges. Implementations can be done in software or hardware and often try to optimize properties such as execution speed, throughput, footprint, reliability, energy consumption, accuracy or maintainability. And despite all these different aspects that fundamentally shape the actual execution of an algorithm, it still produces the same output as defined in theory.

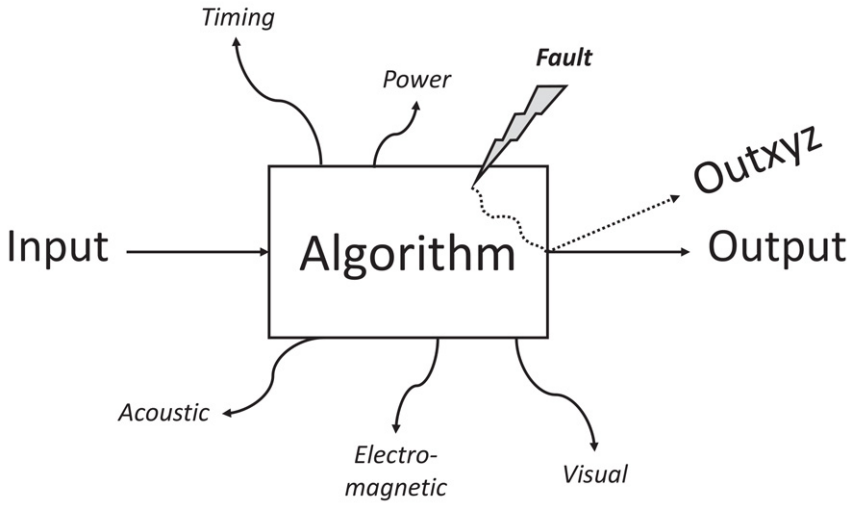


Figure 3.7: Execution of an Algorithm in Practice.

But in addition to delivering a result, an implementation always gives away more information to its environment. These secondary outputs are called side-channels and include the execution time of the algorithm, the power consumption and electromagnetic emanation of the device it is running on and even acoustical or visual phenomena that occur together with the calculations. The idea behind side-channel analysis is to look for relations between the algorithm, particularly the processed data, and these additional information sources. The relations are then used to infer details about what is being processed and how it is processed during the algorithm. If that is possible, it is said that the device *leaks* information. Consequently, the implementation itself is as important as the algorithm, since it strongly contributes to these leaks but can also mitigate them.

Side-channels are typically measured within normal operating limits, however there is also another approach illustrated in Figure 3.7 that uses the same idea of inferring details about the algorithm and its processed data. If an implementation is run at the edge of or even outside its operating specifications, faults are likely to happen. This abnormal behavior can manifest itself in incorrect calculation results, in feedback of erroneous internal states or even in accidental output of secret information. Again, these fault-induced information sources can be used to reveal further internal details, which is referred to as fault analysis. This thesis, however, will focus on non-fault-induced side-channels.

Putting all this into a security context, the algorithms that are executed are typically of cryptographic nature and inferred internal details usually relate to secret information

like cryptographic keys or PINs and passwords. In other words, side-channel and fault analysis can be used to attack implementations of cryptographic algorithms.

3.4.1 Power Consumption

Out of all the possible side-channels, this thesis focuses on the power consumption of digital circuits. The design flow of such circuits starts with a high-level functional specification followed by a more precise behavioral description mostly in a hardware description language. The design is then translated into simple functional blocks in a process known as *synthesis*. These blocks are logic cells or gates and may offer only limited functionality, but when interconnected, they can perform powerful and complex operations. Consequently every algorithm in practice can be broken down or synthesized into these logic cells, of which essentially two types exist. [60]

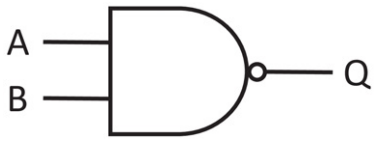
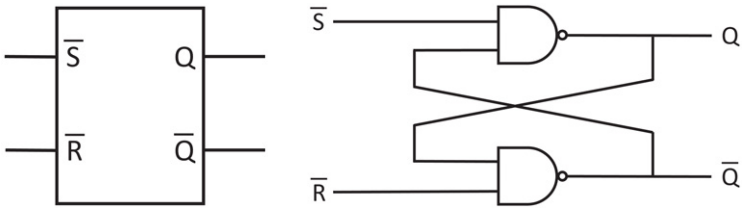


Figure 3.8: NAND Logic Cell [61].

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

Table 3.2: NAND Truth Table.

Combinational cells implement Boolean functions, which means that their outputs are logic combinations of their current inputs. A very well-known example is a NAND gate, performing the Not-AND Boolean function $Q = \neg(A \wedge B)$ as depicted in Figure 3.8. [60]



(a) Logic Cell. (b) NAND Structure.

Figure 3.9: Set-Reset Latch [61].

S	R	Q
0	0	hold
0	1	0
1	0	1
1	1	n.a.

Table 3.3: SR-Latch Operation.

Sequential cells on the other hand produce outputs that not only depend on current inputs but also on preceding ones or initial values. Consequently they are able to memorize values and are thus used for memory structures like latches, flip-flops or registers. A Set-Reset-Latch or SR-Latch built from NAND gates in Figure 3.9 is an example for such a sequential cell. The inputs S and R are written inverted as $\overline{S}$ respectively $\overline{R}$, because they are active-low signals. Table 3.3 illustrates the operation of the latch. If the Set and Reset line are both logic low ('0'), the previous value of Q is saved and no change at the output occurs. If the Set line is logic high ('1') and the Reset line logic low, the latch *sets* the output to logic high. Vice versa, the output is *reset* to logic low. The state of both inputs being logic high is not allowed, since both outputs would be logic high as well, which would contradict the output relation $Q = \neg\overline{Q}$. All in all, one bit of information can be saved with such a structure. [61]

CMOS Going from the logic cell level of the digital circuit further down to a more fine-grained stage, cells get replaced by transistors, which are tiny electronic switches and the most fundamental building blocks of integrated circuits. Out of the different transistor types, metal-oxide semiconductor (MOS) transistors are widely used to implement digital circuits and come in p-channel (PMOS) as well as n-channel (NMOS) variants. A very common process technology that translates logic cells to MOS transistors is the so-called complementary MOS or CMOS technology. The idea behind this logic style is to arrange groups of PMOS and NMOS transistor in a complementary fashion to form bigger units, the logic cells. PMOS groups are used as pull-up networks towards the logic high voltage level, referred to as V_{DD}, whereas NMOS groups work like pull-down networks towards logic low, called V_{SS}. Figure 3.10 shows the previously mentioned NAND cell implemented in CMOS technology. [61]

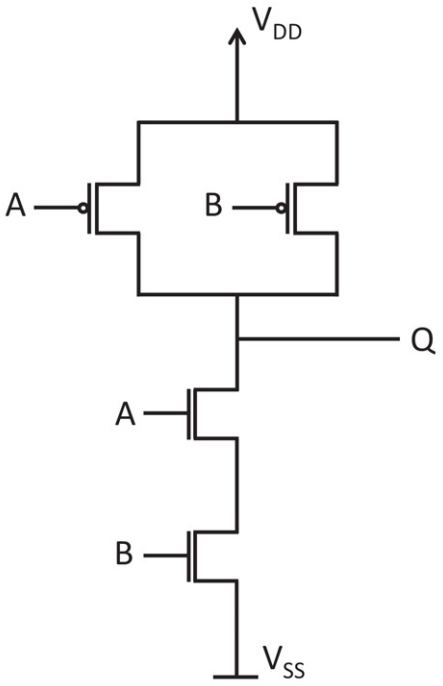


Figure 3.10: NAND Cell in CMOS.

The complementary groups are designed in a way that they never conduct at the same time when a state has settled. Thus, when the inputs do not change and remain *static*, then the outputs do no change either and almost no current is drawn by the cell. Consequently for output transitions of $0 \rightarrow 0$ and $1 \rightarrow 1$ CMOS cells have only very little *static* power consumption. When the inputs change and trigger an output switch, significantly more current is drawn for a short period of time. This is referred to as *dynamic* power consumption and occurs during output transitions of $0 \rightarrow 1$ and $1 \rightarrow 0$. It is caused by charging the load capacitance of the cell and by short circuits during the transition process. Phenomena that increase dynamic power consumption even further are glitches. They stem from the fact that signals have a certain propagation delay in digital circuits and that combinational cells often take inputs from different sources. Thus, it can happen that an input changing early initiates a cell output transition and another input arriving late will trigger yet another transition or interrupt the first one. The bigger the combinational circuit is, the bigger the impact on the dynamic power consumption. [60]

Because of the CMOS design, dynamic is more dominant than static power consumption. This observation indicates that the overall power consumption of a CMOS circuit significantly varies with the data that is being processed. This supports the idea of analyzing side-channels that relate to the power consumption, which can be done in two ways. First, there is a direct approach where a power measurement circuit can be used to tap the side-channel between the digital circuit and the power supply. Since power depends on current and voltage, an easy and popular way is measuring the current consumption while keeping the voltage supply constant. In its simplest form, the voltage drop across a resistor in the V_{DD} or V_{SS} line can be measured. The second possibility is an indirect access via the electromagnetic emanation that comes with the current flow in the device. It can be measured with special electromagnetic probes. This approach even introduces a spatial component to the measurements, since subsections of a digital circuit can be potentially assessed separately. [60]

HW/SW The observations about the CMOS process technology are very generic and cover a large set of different ways to execute an algorithm. As long as the hardware that eventually processes the data behaves as described above, its power consumption can be potentially used as a side-channel. Thus, regardless of whether the algorithm is run on pure hardware, like an Application-Specific Integrated Circuit (ASIC) or a Field-Programmable Gate Array (FPGA), or in software on a processing unit that only executes a set of predefined instructions, it is expected that information leaks to the outside world as long as no preventive measures are taken.

3.4.2 Analysis Methods

After a side-channel is found and a signal that incorporates its information can be measured, an analysis step is used to finally reveal device-internal details. The field of side-channel analysis was pioneered in 1999 by Kocher et al. [62], and has been an active field of research ever since. Two common methods are discussed in this section: Simple Power Analysis and Differential Power Analysis.

SPA A straightforward way of analyzing a recorded side-channel signal, often referred to as trace, is described by *Simple Power Analysis* or SPA. The idea behind it is to map (parts of) the power profile to what is actually being done on the device at that instant of time. A very strong prerequisite to this method is a certain level of knowledge about how the algorithm is implemented and how the device it runs on works in detail. If that knowledge is given, the advantage of SPA is that it works with very few traces, which keeps the measurement time short. However, the traces themselves must be of good quality (e.g. not noisy), which introduces requirements for the measurement setup. [60]

A prominent example is the implementation of the asymmetric cipher RSA introduced by **R**ivest, **S**hamir and **A**dleman in 1978 [63]. It is calculated using exponentiation with modular reduction and as every Public-Key Cryptosystem comes with a private and a public key. The following two equations illustrate how encryption, decryption and also message signing (= decryption of a plain message) works. c and m stand for ciphertext and (plain) message respectively, e denotes the public whereas d represents the private exponent. n is used for the modular reduction, an essential property of the cipher.

$$c \equiv m^e \pmod{n} \tag{3.1}$$

$$m \equiv c^d \pmod{n} \tag{3.2}$$

Since in practice very large numbers are involved in the process, efficient implementation of the rather costly exponentiation is usually a requirement. One common approach is the square-and-multiply algorithm, mentioned in the original paper in [63]. To start with, d is written in its binary representation $d = d_K d_{K-1} d_{K-2} ... d_0$, where d_i denotes the binary digits. Then, the following algorithm is executed to calculate $m \equiv c^d \pmod{n}$.

1. $m = 1$
2. **For** $i = K...0$ **Do**
   (a) $m \equiv m^2 \pmod{n}$
   (b) **If** $d_i = 1$ **Then** $m \equiv m \cdot c \pmod{n}$
3. Stop.

The important property of this algorithm is that the multiplication in step *2b* is only done when the current binary digit of d is '1', thus adding an additional operation to the squaring. This is something that can be observed in the power profile, since different operations typically consume different amounts of power. Figure 3.11 shows a simulated trace that relates to the power consumption of a device during exponentiation. The clearly distinct power profiles of squaring and multiplication shall illustrate how to derive the secret exponent d from one trace only. The assignment which power consumption level corresponds to which operation can be done by looking at the algorithm again. It clearly states that squaring ('S') is always the first operation, thus the lower power level is assigned to it. With the higher power level corresponding to multiplication ('M'), the trace yields the sequence *S M - S - S M - S*, which gives $d = 1010_b = 10_d$. Even though RSA is secure in theory, a simple implementation can be successfully attacked with little effort, which demonstrates the powerfulness of side-channel analysis.

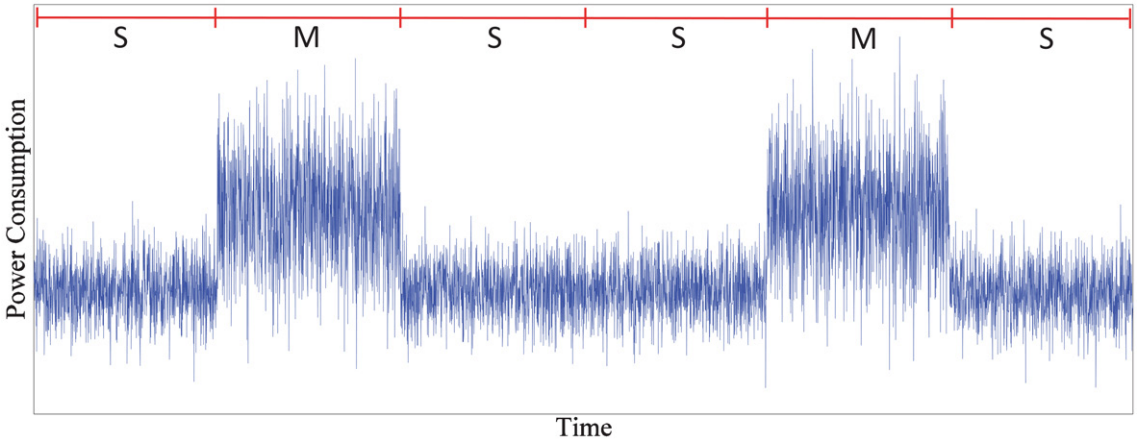


Figure 3.11: RSA Square and Multiply Power Trace.

DPA The biggest disadvantage of SPA is the required knowledge about implementation and device details followed by the fact that the traces must be of rather good quality. These obstacles are being addressed with the next analysis method called *Differential Power Analysis* or DPA, which is very well described in [60]. This technique consists of five steps that are summarized here and depicted in Figure 3.12.

1. **Algorithm Analysis** DPA can be used to reveal constant secret data. As a first step such a secret target must be specified. In order to do this, the algorithm is analyzed for appropriate intermediate values. The requirement is that these values i are a function of the constant secret part c and a varying known part v, which yields $i = f(v, c)$. In cryptographic algorithms typical candidates are key mixing functions in the first or last round. This is because at the beginning and end of the algorithm, varying plain- and ciphertext parts, which are typically known, are processed together with the constant secret key.

2. **Measurement** After an intermediate value has been found, the algorithm is executed N times in a row. For each execution the value of v is saved together with the measured power consumption t. The varying inputs are summarized in a vector $\mathbf{v} = (v_1, v_2, v_3, ..., v_N)$, whereas the measured traces of length T are stored in a $N \times T$

matrix **T**. It is very important that the traces are aligned, which means that the same operation always occurs at the same instant of time and consequently in the same column of the matrix **T**.

3. **Intermediate Values** Now that all values for the varying input are given, hypothetical intermediate values can be calculated. In order to do this, the vector **v** is combined with vector $\mathbf{c} = (c_1, c_2, c_3, ..., c_M)$, which contains all possible values for the constant input c. The new matrix **I** that is created contains N rows, one for each algorithm execution, which are each filled with M possible outcomes of the intermediate value function $i_{n,m} = f(v_n, c_m)$. The goal of DPA is now to find out which column and consequently which value of c has been used in the algorithm.

4. **Power Model** The next step is to translate the previously calculated intermediate values to a hypothetical power consumption. This is where additional knowledge of the device comes into play. The more it is known about the implementation of the algorithm, the better the simulated power consumption values relate to the real ones. Common choices are discussed below. The result of this step is again a $N \times M$ matrix called **H**, where each intermediate value is substituted with its corresponding hypothetical power consumption.

5. **Comparison** The inputs to the last step are the hypothetical power consumption matrix **H** of size $N \times M$ and the measurement trace matrix **T** of size $N \times T$. The goal now is to compare the power consumption at each instant of time (equals one column in **T**) to all power consumptions that the different values for c would produce (equals all columns in **H**). Every comparison is saved as one value in a new result matrix **R** of size $M \times T$. The entry $r_{i,j}$ tells how likely constant value choice number i was actually processed at time instant j. Typically the calculation is done in a way that the maximum or minimum of all entries in **R** yields the correct constant value choice and at which time instant it is processed. The statistical tools commonly used in this step are also discussed below.

When mapping an intermediate value to a simulated power consumption, the knowledge about the device plays an important role. The more it is known about the digital circuit, the better the simulated values relate to the real ones. And while a circuit designer has access to all kinds of details, an attacker usually deals with a black box scenario. This is why the two power models presented here are rather simple but also very generic and thus apply to a wide range of implementations. As already discussed, CMOS circuits by design show a bigger dynamic power consumption than a static one. This dynamic portion depends on the number of $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions that occur in the circuit. Under the assumption that both transitions cause the same power consumption and that the static portion can be neglected, counting the $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions in a given time frame yields a simple indicator known as the Hamming-Distance Model. In general the Hamming distance (HD) is a metric to describe a difference. In computer science the HD of two binary values is defined as the number of bits that need to be modified to convert the first into the second value. For example the HD of 7 (111_b) and 1 (001_b) is 2, since the two most significant bits must be changed to turn a 1 into a 7. Consequently the Hamming distance can be used to keep track of the number of transitions in a digital circuit, if two consecutive states are known. [60]

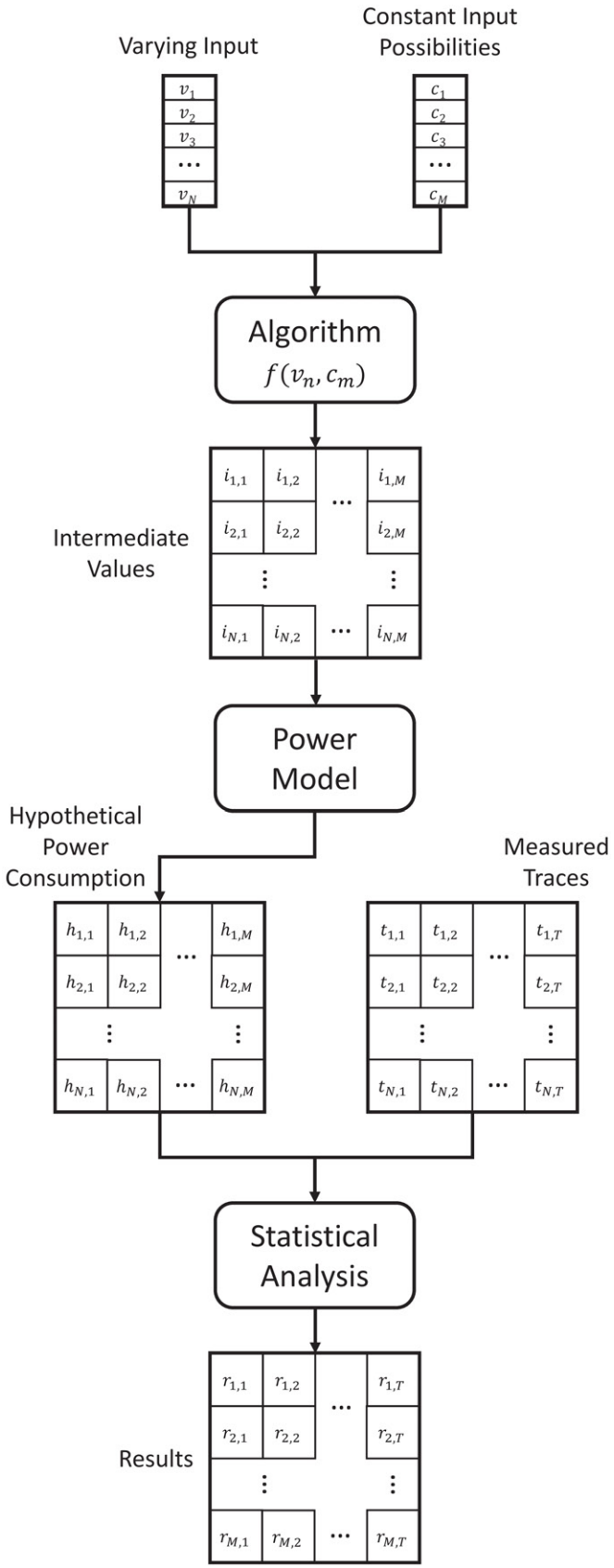


Figure 3.12: Steps of a Differential Power Analysis [60].

For combinational cells this is hardly ever the case in practice, since glitches interfere with the circuit. For sequential cells likes registers the value that is saved typically gets updated each clock cycle. Thus, if the register contents in two consecutive clock cycles are known, the Hamming distance can be calculated. Data buses that are often connected to such sequential cells are also suitable for the HD model, even more because their big capacitive load significantly contributes to the overall power consumption. Consequently values that are transferred one after the other over the bus are potential targets. If two consecutive states are not known, an even simpler model can be used to estimate the power consumption. It is called the Hamming-Weight Model. For a given number the Hamming weight (HW) is defined as the number of digits different from zero. In binary, this corresponds to the number of ones. Consequently the HW of 5 (101_b) is 2. For binary numbers the Hamming distance relates to the Hamming weight as shown in Equation 3.3. [60]

$$HD(v_1, v_2) = HW(v_1 \oplus v_2) \tag{3.3}$$

The XOR of two values represents the differences as ones, which are then counted as the Hamming weight and subsequently yield the Hamming distance. In contrast to the HD model, the HW model ignores values before or after a given one. In theory, this does not suit CMOS circuits very well, which have the characteristic of consuming more power when transitions occur. However, in practice the Hamming weight is not completely unrelated to the power consumption and even fits it very well in certain situations. For example, if a bus pre-charges all lines to zero each time a new value is transferred, then both models describe the behavior equally well, as shown in Equation 3.4. The same concept applies to registers that are initialized with or reset to zero and then filled with a new value. [60]

$$HD(0, v) = HW(0 \oplus v) = HW(v) \tag{3.4}$$

As shown above, both models have in common that certain details of the implementation must be known or easy to guess by an attacker. Since cryptographic algorithms nowadays follow Kerckhoff's principle [64], their inner working is publicly documented and scientific work on various implementations is often available. Thus, this prerequisite is typically fulfilled.

After the intermediate values are translated to simulated power consumption values with an appropriate power model, they must be compared with the real measurements. During that process, one instant of time or one column of matrix $\mathbf{T}$ is compared to the power consumption of all possible values for the unknown constant input c, which are stored column-wise in matrix $\mathbf{H}$. In other words, for each moment in time a vector of length N is compared to M vectors also of length N. This is done T times. A common metric to describe the similarity of two sequences is correlation. It can express the linear relationship between two variables as a correlation coefficient ρ, which lies between -1 and 1. The value 0 denotes no correlation, whereas ± 1 represents full positive respectively negative correlation. The formula to obtain ρ is given in Equation 3.5, where $Cov(X, Y)$ denotes the covariance of variables X and Y and $Var(\cdot)$ their corresponding variances. [60]

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sqrt{Var(X) \cdot Var(Y)}} \tag{3.5}$$

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x}) \cdot (y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \cdot \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{3.6}$$

When actually performing a DPA, ρ must be estimated as shown in Equation 3.6. The result r is called Sample Pearson Correlation coefficient. From a practical point of view, this metric is resilient (e.g. robust against noise) and compared to other distinguisher has a very good balance of computational effort and outcome, even more when the power model matches the target very well. Thus, the correlation coefficient is widely used in differential side-channel analysis, which is why the entire analysis process is often referred to as Correlation Power Analysis or CPA.

DPA attacks are often distinguished by the number of intermediate values they attack and thus how many points in time they consider. If one intermediate value is processed, the attack is said to be 'first-order', two values would be 'second-order' and so on, which is then often summarized as 'high-order'. The attacks presented in this thesis are first-order CPA attacks.

DEMA As already discussed shortly, two side-channels that relate to the power consumption of a device are typically measured in practice: current consumption and electromagnetic emanation. The methods described in this section can be applied to both of them, which in case of a DPA together with EM traces becomes a so-called Differential Electromagnetic Analysis or DEMA. Measuring the electromagnetic emanation of a device has two basic advantages. First, the device can be measured in place and no circuit around it has be modified, which has an important implication for practical attacks, because the less invasive the attack is, the less suspicion is raised. Second, electromagnetic radiation can be used to only focus on what is important. Typically integrated circuits have very few and often only one supply pin, over which the current for the entire circuit is drawn. Consequently, when measuring the current, not only the targeted part of the circuit is recorded (e.g. the cryptographic core) but also other parts like communication interfaces or memories. In contrast to that, electromagnetic fields are strongest close to where they are induced. Consequently, the idea is to focus on a specific part of the circuit by measuring the near-by EM field, which is known as localized EM radiation. It has been shown by Heyszl et al. [65] that for localized EM measurements small distances between circuit and probe as well as high measurement resolutions are recommended. However, to ensure that the EM probe has minimal distance to the circuit, the package of the chip is usually opened, which renders the attack to be more conspicuous in practice again. Nevertheless, EM emanation is an important alternative to current drain in order to utilize the side-channel opened up by a circuit's power consumption.

3.4.3 Data Encryption Standard

One example of a cryptographic algorithm that can be targeted during a side-channel analysis is the so-called Data Encryption Standard or DES. In 1973 the National Bureau of Standards issued a call for proposals looking for a cryptographic algorithm that would be turned into a national standard. IBM and the research team around Horst Feistel submitted an algorithm called Lucifer, which was passed on to the National Security Agency (NSA) for review. The modifications made to the algorithm by the NSA caused much controversy in the upcoming years but essentially remained and formed the final algorithm officially proclaimed as DES. [67]

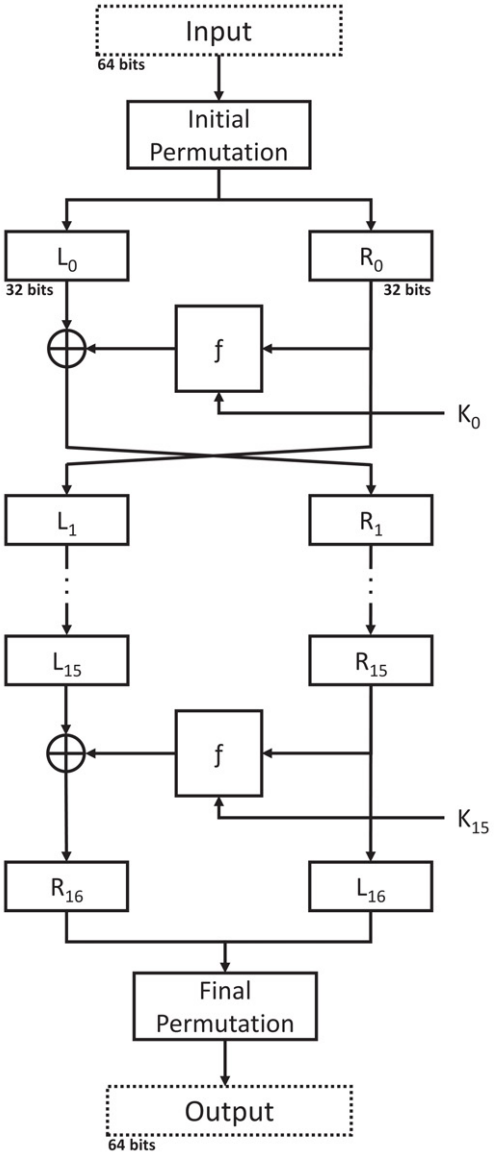


Figure 3.13: DES Feistel Network [66].

DES is a symmetric block cipher. It operates on input blocks of 64 bits and produces outputs of same size. The 64-bit key contains eight integrity bits, thus reducing the effective key size to 56 bits. DES has a so-called balanced Feistel structure that is shown in Figure 3.13. This means that the internal state is split in half (here: R_i, L_i), one of which is fed into a round function f. The output is XOR'ed with the other half and then the halves are swapped. For the DES this is done 16 times ($i = 0...15$), R_{16} and L_{16} already being the output stage. Before and after these rounds there are two permutations, the Initial and Final Permutation, which are inverse to each other. However, they do not add to the cryptographic strength of the cipher. All 16 rounds have their own 48-bit round key K_i derived from the original 56-bit key. This is done with a set of permuted choices. The advantage of the Feistel structure used in the DES is that encryption and decryption only differ in a reverse ordering of the round keys. Other than that, the algorithm stays the same, which is particularly beneficial for hardware implementations. The round function is shown in Figure 3.14. It takes an half block and increases its size with an expansion function, a permutation with bit re-usage, to that of a round key. Both are XOR'ed and fed into eight substitution boxes (S-box), which take 6-bit inputs and transform them into 4-bit outputs according to a non-linear function. These S-boxes form the core security of the DES. Without them the cipher would be easily breakable because of its otherwise linear operations. After that another permutation spreads the output of each S-box. [66, 67]

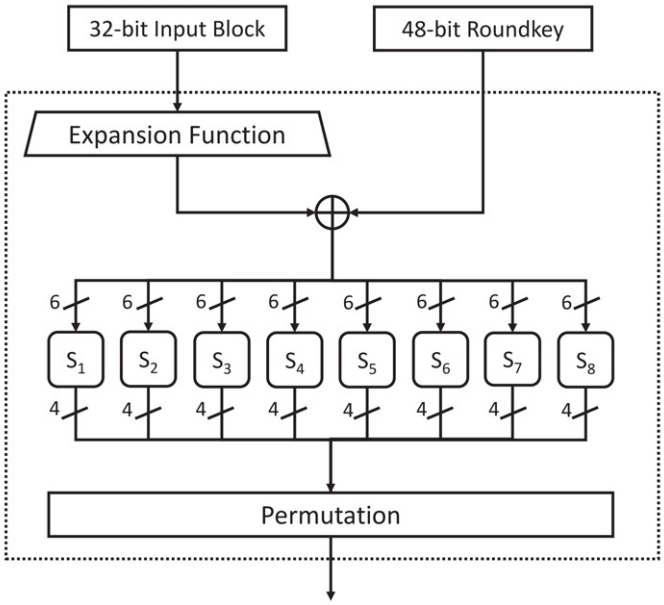


Figure 3.14: DES Round Function [66].

DPA Targets As already discussed, the selection of an appropriate intermediate value and power model are important steps during a DPA. The intermediate value must depend on a known varying and a secret constant value. A first approach is to look at the first or last round, since the plain- and ciphertext are often known. Furthermore the intermediate value must be chosen in a way that a power model can be used that fits the implementation well. For hardware implementations of many cryptographic algorithms it is common practice to store the internal state in registers and update it each round or after certain steps. If two consecutive values of such a register are known, the HD power model can be used.

Under the assumption that a hardware DES uses such a state register, the first round can be used in an attack. L_0 and R_0 can be easily derived from the plaintext and are thus known and varying inputs. The first round key K_0 would be the secret and constant part. It is assumed that the part of the register, where R_0 is stored, gets updated after the first round with the new value R_1. This relation is illustrated in Equation 3.7.

$$R_{i+1} = L_i \oplus f(R_i, K_i) \tag{3.7}$$

With the varying and constant inputs set, the hypothetical intermediate values are then all possibilities for R_1. The power model that translates them into a simulated power consumption is $\mathrm{HD}(R_0, R_1)$. Since processing all possible values for K_0 at once is a big computational effort, usually only parts of it are considered during an attack. This divide-and-conquer approach can be done by reducing the number of bits involved, for instance by calculating only a subset of the S-boxes. Consequently, to recover an entire round key, multiple attacks have to be mounted. Other possible DPA targets are the left halves, e.g. L_0 and L_1, or the in- and output of the S-boxes as illustrated in Equations 3.8 and 3.9.

$$S_{in,j} = (E(R_i) \oplus K_i)_j \tag{3.8}$$

$$S_{out,j} = SBox(S_{in,j}) \tag{3.9}$$

Here, $E(\cdot)$ denotes the expansion function, i defines the current round and j the current 6-bit chunk of the overall S-box input and also the current 4-bit chunk of the overall S-box output. In the first round R_0 is again the known and varying input and K_0 again the constant secret. The corresponding power model is $\mathrm{HD}(S_{in,j}, S_{out,j})$. Note that only 4 bits of the input are selected. A variation of this last scenario would be to use a Hamming weight power model and apply it only to the output of the S-boxes, $\mathrm{HW}(S_{out,j})$.

Security It shall be mentioned here that DES itself cannot be considered cryptographically secure nowadays and is thus not recommended to be used as is. This has definitely been shown in 1998 by the Electronic Frontier Foundation (EFF), who built the EFF DES Cracker. With this device it is possible to recover any DES key within a couple of days using a brute-force attack [68]. Furthermore there are numerous alternative ciphers nowadays that meet a broad spectrum of different requirements, one of them being the official successor to DES, the Advanced Encryption Standard or AES, which has been well studied and widely adopted.

However, in compatibility scenarios DES can still be used to build a cryptographically secure cipher, if cascaded multiple times in a row. This structure was published as a recommendation by the National Institute of Standards and Technology (NIST) under the name Triple Data Encryption Algorithm or TDEA, which is usually referred to as Triple DES, TDES or 3DES. The idea is to use DES three times in a row for en- and decryption. The exact modus operandi is given in Equations 3.10 and 3.11. P and C denote plain- respectively ciphertext, E and D stand for DES in en- and decryption mode and K_i represents the key for the current DES stage. [66]

$$C = E_{K_3}(D_{K_2}(E_{K_1}(P))) \tag{3.10}$$

$$P = D_{K_1}(E_{K_2}(D_{K_3}(C))) \tag{3.11}$$

There are two keying options mentioned in the NIST paper. The first defines three unique keys ($K_1 \neq K_2 \neq K_3$) having 168 effective key bits, the second one defines two unique keys ($K_1 \neq K_2, K_3 = K_1$) having 112 effective key bits. These options are sometimes referred to as 3TDES and 2TDES. Since DES is still the main building block of TDEA, the DPA targets do not change. However the effort to recover an entire TDEA key increases depending on the keying option. [66]

Chapter 4

Standard Ticket

This chapter contains a short introduction to the Standard Ticket, a section about the underlying chip and its properties and one about its usage in the EZ-Link system. After that security and privacy concerns are discussed followed by a short conclusion.

4.1 Introduction

The EZ-Link Standard Ticket is a limited-use ticket for single MRT or LRT trips. The version evaluated in this thesis is already the next generation of the ticket, which has been phased in from November 2012 to March 2013 and completely replaced the old one shown in Figure 4.1a [69]. For this reason, only the new version of the Standard Ticket is analyzed here, which can be bought at General Ticket Machines in train stations by selecting a route on the display. The system converts the route to a price, which has to be paid. The ticket can then be used to go on the selected route, which is either a one-way or round trip. After that it has to be reloaded in order to be used again.

Visual Appearance There are several limitations printed on the back side, which apply to the usage of a next generation Standard Ticket shown in Figures 4.1b and 4.1c. A trip that has been paid for is valid only on the day of purchase. Furthermore the ticket can be reloaded up to six times and is valid for 30 days. The ticket cost of S$ 0.1 is added to the first fare, subtracted from the third and additionally discounted on the last trip, which renders the ticket free of charge and gives an incentive to use it to its full extent.

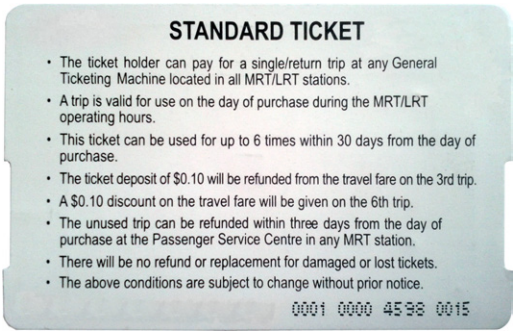
Apart from those terms and conditions, which dominate the visual appearance of both sides, the serial number of the ticket is printed on the back side in the lower right corner. The logos of the Singapore public transport and Transit Link are printed on the front side in the upper left and right corner respectively.

4.2 Platform

This section is split up into two parts, the first of which contains the identification of the Standard Ticket's underlying chip whereas the second discusses its properties and features.

(a) Phased-out Standard Ticket.

(b) Next-Gen Front View.

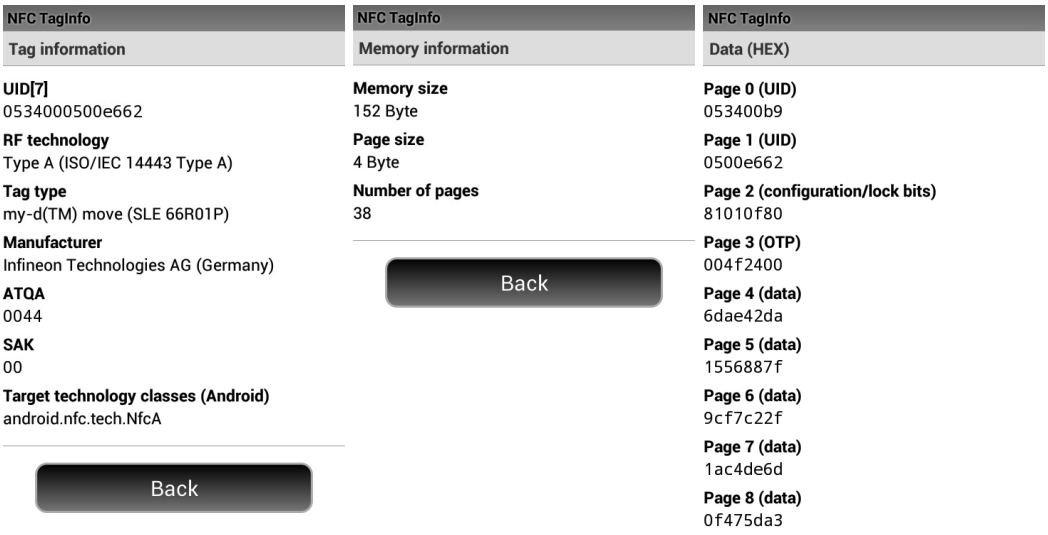


(c) Next-Gen Back View.

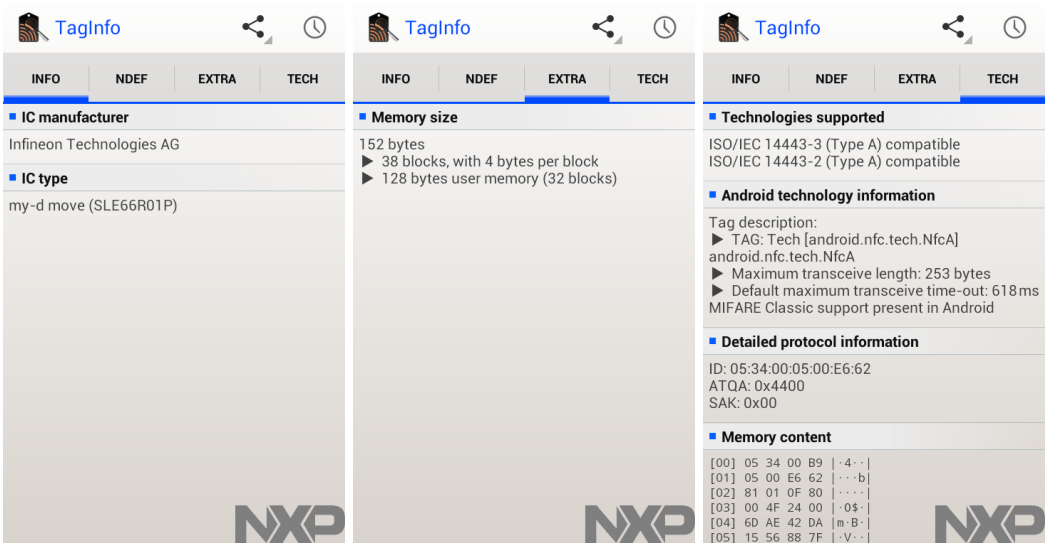
Figure 4.1: EZ-Link Standard Ticket.

4.2.1 Identification

Software For a first analysis of the underlying chip, a Google Nexus S smartphone running Android v4.1.2 together with the applications NFC TagInfo (v1.11) [70] and NFC TagInfo by NXP (v2.00) [71] is used. Both applications identify an ISO/IEC 14443 Type A compatible chip from Infineon Technologies AG: *my-d move SLE66R01P*. In order to verify the final result given by the applications, it is important to take a look at the details shown in Figure 4.2. According to ISO/IEC 14443-3 the Answer-To-Request Type A (ATQA) coded as 4400_h indicates a double-sized (7-byte) unique identifier. This ID is given in both applications as $0534000500e662_h$. The final Select-Acknowledge (SAK) coded as 00_h rules out compliance to ISO/IEC 14443-4, which is shown in the rightmost NFC TagInfo by NXP picture, where the section *Technologies supported* lists only layers 2 and 3. The first byte of the UID, 05_h, encodes the chip manufacturer. According to ISO/IEC 7816-6/AM1 [72] this value corresponds to Infineon Technologies AG. Following the my-d move short product information sheet, the nibble 3_h of the second second byte, 34_h, is used to code the chip family and indicates a my-d move or my-d move NFC, also known as SLE66R01P resp. SLE66R01PN [73]. The displayed memory size of 152 bytes matches both models. The suffix 'N' indicates an initialization following NFC Forum Type 2 Tag specifications, without the suffix the chip is a plain my-d move. Hence, the difference is only the memory content. According to NFC Forum's Type 2 Tag Operation Specification [74] page 03_h acts as a so-called Capability Container, storing information about the device. To indicate Tag Type 2 compliance this page has to start with $E1_h$, which is clearly not the case for the Standard Ticket that starts with 00_h. Thus, from the software side the chip type can be pinpointed.



(a) NFC TagInfo Screenshots (Inverted Color).

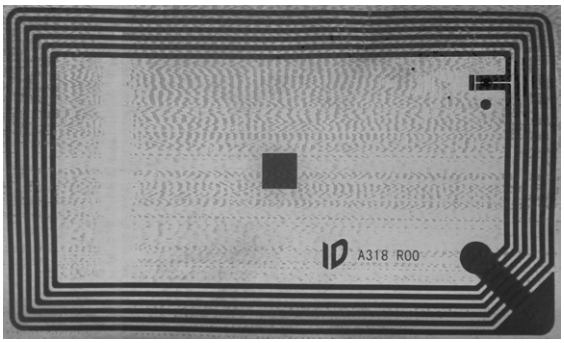


(b) NFC TagInfo by NXP Screenshots (Inverted Color).

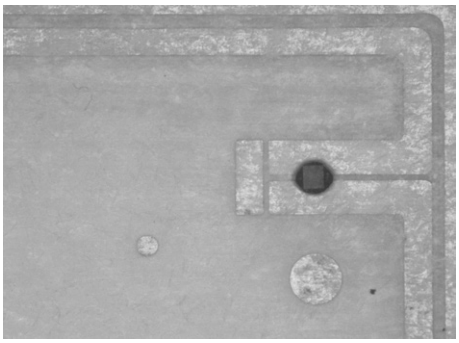
Figure 4.2: EZ-Link Standard Ticket Identification with an NFC-enabled Smartphone.

Hardware The Standard Ticket is a so-called paper ticket, meaning that its outermost package is made out of paper where usually a visual design and additional information are printed on. The layers of paper can be softened with some warm water and then removed, revealing a plastic inlay, inside which the IC and its 6-turn antenna are housed, as shown in Figure 4.3a. The string 'ID' in the lower right part of the inlay suggests that the inlay is a product of the Identive Group [75]. The close up views in 4.3b and 4.3c show a small, rectangular chip stuck in some epoxy resin to the two antenna connectors that point towards the inside of the inlay. Figure 4.3d shows the back side of the IC after it has been removed from the epoxy resin with fuming nitric acid.

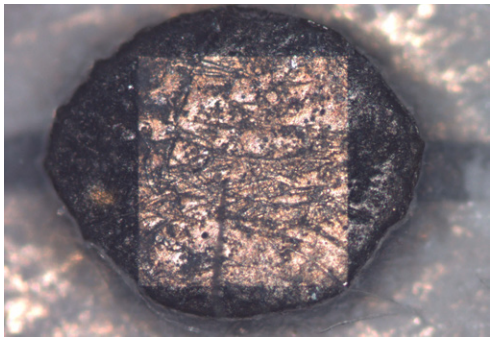
Figure 4.4a shows the front side of the exposed die that would normally face the antenna connectors. It provides an insight into the inner parts and wirings of the chip.



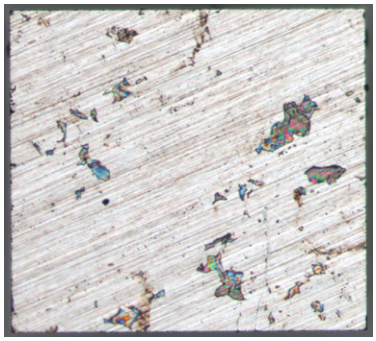
(a) Plastic Inlay.



(b) IC Close Up.



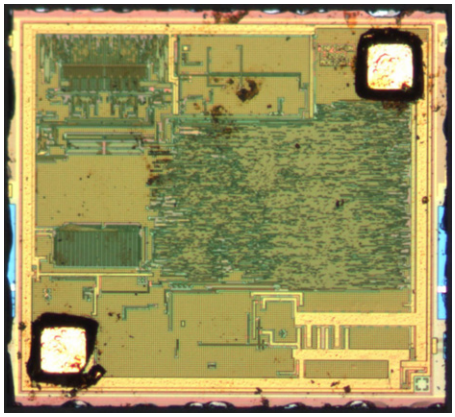
(c) IC Attached to Antenna.



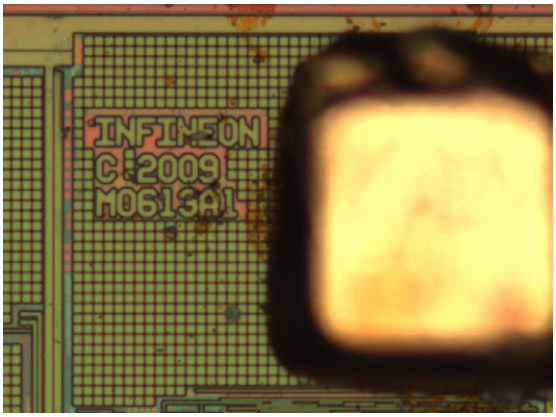
(d) IC Back Side.

Figure 4.3: EZ-Link Standard Ticket IC Decapsulation.

The bond pads in the lower left and upper right corner link the die with the antenna and clearly stand out in the picture. Next to the upper right one there is a further clue that supports the previously made conclusions about the chip model. Although the given year 2009 could point to different moments in an IC life cycle and the string 'M0613A1' remains unclear, the relation to Infineon is reinforced with this observation.



(a) Die Front Side.



(b) Bond Pad Close Up.

Figure 4.4: Infineon my-d move Die.

4.2.2 Properties

Having discovered the type of the chip, the short product information sheet in [76] documents its properties. Besides the compliance to ISO/IEC 14443-3 Type A and NFC Forum Type 2 Tag specifications, the chip mainly offers a 152-byte EEPROM memory containing special memory elements as well as general purpose memory sections, both of which can be accessed with a custom command set.

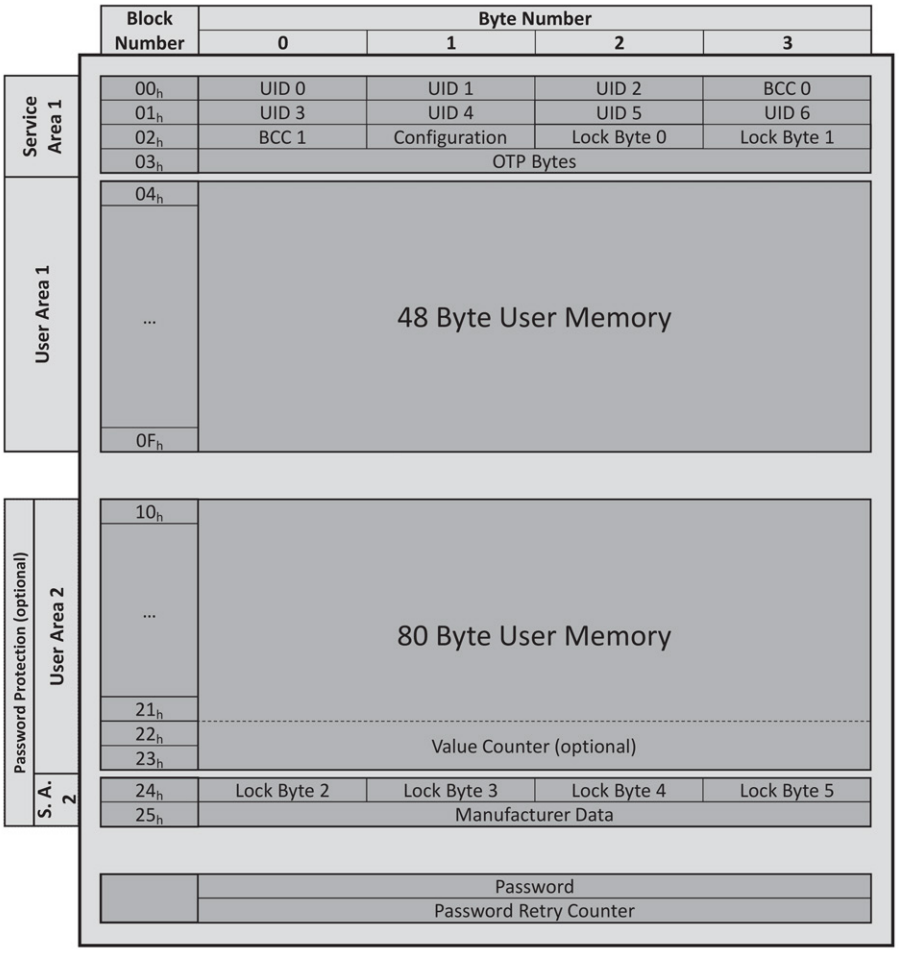


Figure 4.5: Infineon my-d move Memory Layout [76].

38 blocks indexed from 00_h to 25_h with four bytes each form the 152-byte memory of the my-d move, as shown in Figure 4.5. It is basically split up into two sections, both having a service and a user area. The service area contains special memory elements, whereas the user area provides general purpose memory storage. Apart from those two sections, a 32-bit password and its retry counter are stored separately and do not contribute to the 38 blocks. A 16-bit value counter, if activated, is laid over the last two user area blocks of the second section and enables corruption resistant counting. [73, 76]

In order to explain the memory layout in more detail the content of an EZ-Link Standard Ticket is read with the application NFC TagInfo and illustrated in Table 4.1.

Block [hex]	Content [hex]	
00	05 34 00 b9	S_1
01	05 00 e6 62	
02	81 01 0f 80	
03	00 4f 24 00	
04	6d ae 42 da	U_1
05	15 56 88 7f	
06	9c f7 c2 2f	
07	1a c4 de 6d	
08	0f 47 5d a3	
09	00 00 00 00	
0a	9d 5d fb 69	
0b	cb e8 4c 7e	
0c	85 37 9a 87	
0d	55 ef d0 82	
0e	a5 07 7d 94	
0f	02 bd 99 6f	
10	00 00 00 00	U_2
11	00 00 00 00	
12	00 00 00 00	
13	00 00 00 00	
14	00 00 00 00	
15	00 00 00 00	
16	00 00 00 00	
17	00 00 00 00	
18	00 00 00 00	
19	00 00 00 00	
1a	00 00 00 00	
1b	00 00 00 00	
1c	00 00 00 00	
1d	00 00 00 00	
1e	00 00 00 00	
1f	00 00 00 00	
20	00 00 00 00	
21	00 00 00 00	
22	00 00 00 00	
23	00 00 00 00	
24	00 00 00 00	S_2
25	32 c0 83 c2	

Table 4.1: Standard Ticket Memory Content.

Service Area 1 S_1 contains the UID of the chip including two integrity check bytes and a configuration byte followed by two lock bytes that enable write-protection of selected memory blocks. The last block of the service area is formed by one-time programmable (OTP) bytes. [76]

UID The 7-byte UID is used to uniquely identify the chip. It spans from the first byte of block 00_h to the first byte of block 02_h (no framed box). The two additional integrity check bytes are defined in ISO/IEC 14443-3 and are referred to as BCC bytes. In the example on the left, block 00_h contains the data $053400b9_h$, of which only 053400_h belongs to the UID. $b9_h$ is a BCC byte, which is always the XOR of previous UID bytes. Apart from the last stage, a cascade tag fixed to 88_h is added to the UID for BCC calculation. This gives $88_h \oplus 05_h \oplus 34_h \oplus 00_h = b9_h$. Block 01_h contains $0500e662_h$ and the corresponding BCC byte is stored as first byte on block 02_h. Since this is the last UID stage, the calculation yields $05_h \oplus 00_h \oplus e6_h \oplus 62_h = 81_h$. [53]

ConfigByte The second byte on block 02_h is the configuration byte (framed box). The short product sheet does not contain more detailed information about it, however configuration elements on these devices are usually used to (de-)activate special features like counters or password protection.

LockBytes The third and fourth byte on block 02_h (framed box) are the lock bytes for the User Area 1. Once switched to '1', they provide irreversible write-protection for selected blocks. While this feature is not clearly described in the Infineon my-d move short product sheet, it is covered more extensively in the data sheet of one of its competitors, NXP Semiconductors MIFARE Ultralight MF0ICU1. Both chips are identical regarding their four first blocks. The value of $0f80_h$ means that blocks 03_h, $0f_h$ and the lock bytes themselves are read-only, which is explained in more detail in Figure 4.6. In turn blocks 04_h to $0e_h$ can by read and written. The four lock bytes on block 24_h (framed box) behave similar, but protect blocks in User Area 2. Since they are all zeros, blocks 10_h to 23_h as well as the lock bytes themselves can be modified. [77]

OTPBytes The entire page 03_h is one-time programmable (framed box). The factory default value is all zeros and bits on that page can only be flipped to one but never back to zero. The content $004f2400_h$ is already application specific and cannot be modified due to the lock bytes.

User Area 1/2 U_1 provides 12 blocks or 48 bytes of general purpose memory that can be freely modified by the user. Through the lock bytes of Service Area 1 selected blocks can be irreversibly set to read-only, which has been done with block $0f_h$. U_2 contains 20 general purpose blocks or 80 bytes that can also be write-protected, however using the lock bytes of the second service area. This area is filled with zeros in the example. If the value counter is activated, the last two blocks of U_2 are occupied. [76]

Service Area 2 S_2 contains four lock bytes for User Area 2 and four manufacturer-specific bytes (framed boxes), which are set to read-only by default and contain information about the chip in a proprietary encoding. If activated, the password can be used to add an additional protective layer over Service Area 2 and User Area 2, either restricting read or read/write access. The retry counter can limit the number of unsuccessful authentication tries. [73, 76]

Other The 32-bit password is set to 00000000_h on a Standard Ticket, but is not made use of. The password retry-counter cannot be read. In addition a 16-bit value counter can be used to safely count minimizing the risk of corruption. However, this features is also not in use on a Standard Ticket. [73, 76]

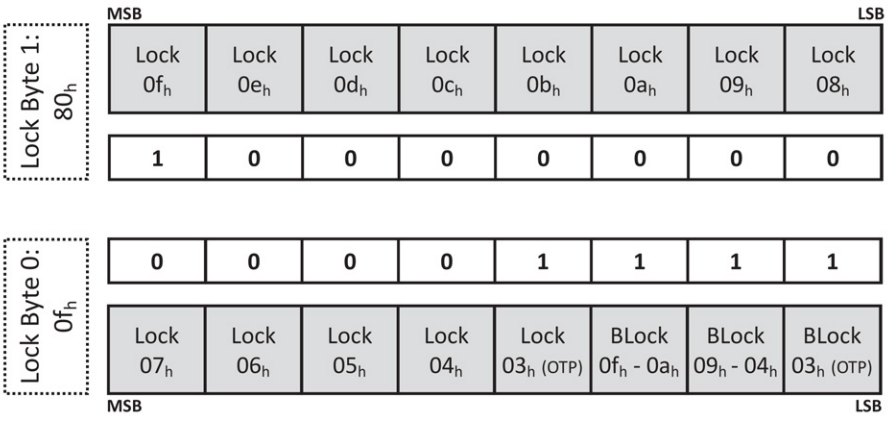


Figure 4.6: Infineon my-d move Service Area 1 Lock Bytes [77].

Figure 4.6 shows how the lock bytes in Service Area 1 are encoded. All blocks in User Area 1 as well as the OTP bytes on block 03_h can be write-protected separately. In order to protect a block, the corresponding lock bit has to be set to '1', hence a '0' indicates full access. Lock bits behave like one-time programmable bits and can only be set from '0' to '1' but not vice versa, thus once a block is protected it cannot be undone. The three special BLock bits are used to write-protect the lock bits themselves. Once these bits are set to '1', their corresponding lock bits cannot be changed anymore. This effectively finalizes the configuration of the lock bytes and is an important step before the card is finally used in the field. In the given example, it can be easily seen that blocks 03_h and $0f_h$ are write-protected as well as all lock bits are protected against further change. [77]

4.3 Implementation

Memory Analysis To illustrate how a Standard Ticket is used during traveling, a single trip is documented here together with the changing memory contents. The trip for this experiment started at Boon Lay MRT station and ended at Clementi MRT station, a section that is part of the green East-West-Line (see Figure 3.1). The ticket price was S$ 1.6 and already included S$ 0.1 ticket cost, leaving S$ 1.5 for the actual trip fare. A memory snapshot was taken after each of the following steps.

- Purchase of ticket at General Ticketing Machine.
- Entrance of MRT station through automatic gates.
- Exit of MRT station through automatic gates.

Block [*hex*]	**Content** [*hex*]
00	05 34 00 b9
01	05 00 e6 62
02	81 01 0f 80
03	00 4f 24 00
04	6d ae 42 da
05	15 56 88 7f
06	9c f7 c2 2f
07	1a c4 de 6d
08	0f 47 5d a3
09	00 00 00 00
0a	9d 5d fb 69
0b	cb e8 4c 7e
0c	85 37 9a 87
0d	55 ef d0 82
0e	a5 07 7d 94
0f	02 bd 99 6f

(a) After Purchase.

Block [*hex*]	**Content** [*hex*]
00	05 34 00 b9
01	05 00 e6 62
02	81 01 0f 80
03	00 4f 24 00
04	6d ae 42 da
05	15 56 88 7f
06	9c f7 c2 2f
07	1a c4 de 6d
08	0f 47 5d a3
09	00 00 00 00
0a	c2 35 90 1a
0b	f4 5b ab 3f
0c	fc 90 7d 51
0d	1b 86 0b 3d
0e	0f 40 68 fa
0f	02 bd 99 6f

(b) After Entering.

Block [*hex*]	**Content** [*hex*]
00	05 34 00 b9
01	05 00 e6 62
02	81 01 0f 80
03	00 4f 24 00
04	aa 04 e2 e3
05	5a 8e d1 0c
06	72 10 e8 a1
07	54 d0 37 a8
08	6d 09 19 b5
09	00 00 00 00
0a	c2 35 90 1a
0b	f4 5b ab 3f
0c	fc 90 7d 51
0d	1b 86 0b 3d
0e	0f 40 68 fa
0f	02 bd 99 6f

(c) After Exiting.

Table 4.2: Standard Ticket Memory Content During a Trip.

The first snapshot representing a freshly bought ticket is directly taken from Table 4.1, Tables 4.2b and 4.2c show the contents of the same card after entering resp. exiting the MRT station. Since Service Area and User Area 2 remain unchanged, they are omitted in the following tables. After entering the MRT station, memory blocks $\texttt{0a}_h$ to $\texttt{0e}_h$ changed, whereas after exiting, blocks $\texttt{04}_h$ to $\texttt{08}_h$ changed. Both modifications are highlighted in gray.

Travel Analysis Besides this low-level usage of the Standard Ticket, an interesting aspect mentioned on the website [69] was confirmed during our experiments. Although a specific route is selected during purchase at a General Ticket Machine, this route is not enforced by the system. Tickets that were bought at Joo Koon MRT station for the route

Joo Koon to Pasir Ris MRT station priced at S$ 2.3 (incl. S$ 0.1 ticket cost) could also be used for other trips on the same and on other MRT lines, e.g. Pioneer to Lakeside (East-West-Line), Lakeside to Bukit Gombak (East-West- to North-South-Line) or Bukit Gombak to Bukit Batok (North-South-Line). As long as the paid fare was big enough, as it is the case for a route going from one to the other end of an MRT line, any trip on the MRT system could be made.

Content Analysis To recover the meaning of the raw data stored on a Standard Ticket, multiple cards were compared after purchase and also during trips. Since no read protection is in place, the entire memory content can be read.

The one-time programmable content of block 03_h is read-only and fixed on all analyzed cards at any time to $004f2400_h$. Thus, it is assumed not to be travel related, since it is set at the purchase of the ticket and cannot be modified any further. The content of block $0f_h$ is also set to read-only. However, this block differs on every card that has been analyzed. Interpreting the block content of $02bd996f_h$ as a 4-byte integer with decimal value 45,980,015, it can be easily found in Figure 4.1c in the lower right corner. Together with the constant prefix '0001 0000' it becomes the serial number of the ticket printed on the back: 0001 0000 4598 0015.

Blocks 04_h to 08_h and $0a_h$ to $0e_h$ change separately from each other during travel and are thus referred to as data packet one and two. Both are 20 bytes in size and seem to change randomly after every reader interaction. No similarities could be found between any of the data packets on any card. Block 09_h remains at 000000_h at all times. Due to the random behavior of the involved data, travel related information is assumed to be scrambled or only referenced rather than being directly encoded in the raw bytes of the data packets.

4.4 Privacy Evaluation

Note Before continuing with the assessment of the Standard Ticket's properties and usage with respect to security and privacy concerns, a few extra statements have to be made. First, the following sections comprise discussions about underlying technologies of the Standard Ticket as well as practical experiments performed to test for implementation details. These discussions are always put into the context of a public and widely used system, since the Standard Ticket is used in public transport. This is why some points made here might not apply to other, e.g. more access restricted, systems. Second, since EZ-Link is a real-world system, the author wants to stress that during the experiments no damage, including financial loss, was done to the system itself and consequently to EZ-Link Pte Ltd. Third, the evaluation is solely based on publicly available information or reverse engineering results. Features and mechanisms currently not being used are discussed as well in order to give a comprehensive overview.

Uniqueness What comes inherently with RFID is the possibility to identify a given transponder. The uniqueness of an identifier always depends on the limits it is considered in. What might be unique in a closed system might not be globally. The Infineon my-d move ships with a 7-byte read-only identifier set by the manufacturer, of which one byte is used for the fixed manufacturer ID and four bits are used to set the chip family [73]. This leaves 44 Bits or about $17.59 \cdot 10^{12}$ possible numbers under the control of the manufacturer

to tell different my-d move chips apart. Additionally, there is a serial number printed and digitally stored on the card. Considering the current implementation with the prefix of '0001 0000' not being stored on the card, the 8-digit number provides 10^8 different identifiers under the control of the card issuer. Although the 32-bit number stored on the card can hold more values than that, without further implementation details known the decimal representation limits the number to eight digits. If the prefix is included, the number consequently increases to 10^{16} possibilities. And while the UID must be considered in a global context, the serial number applies to the EZ-Link system only. Since the number of issued Standard Tickets is not known to the author, it cannot be assessed whether these sources of uniqueness are sufficient and for how long. Nevertheless, both identifiers are a potential way to distinguish cards, which can also be used to enable more advanced security measures.

Anonymity The fact that RFID transponder can be identified from a distance without leaving any trace has repeatedly raised privacy concerns. One prominent example is supply chain management and the so-called Electronic Product Code or EPC. The purpose of EPC is to provide a globally unique and universal identifier for any physical object. It is maintained by EPCglobal and specified in the Tag Data Standard [78]. EPC is commonly used together with RFID tags that operate in the UHF range and reach operating distances of several meters when passively powered. The concerns are consequently based on the fact that these tags are incorporated in many items and can be uniquely identified from a considerable distance without attracting the owner's attention.

In [79] Garfinkel el al. discuss privacy threats regarding EPC and RFID. Items and their unique product code might be associated with their owner's identity. Unlike loyalty programs this association however can happen involuntarily and on an per-item-basis rather than with an entire product group. Consequently, shopping or product preferences can be inferred for example. The EPC uniqueness and the operating distance of UHF tags also enable movement or behavioral pattern tracking of their owners, if for example RFID reader are covertly set up at public places. And while there are other threats as well and also features that try to mitigate some of them, like the kill command that permanently deactivates a tag, this topic continues to be of interest.

The reason to mention this discussion, is that the prerequisites of the threats mentioned above are basically also given for RFID tags like the Standard Ticket. The UID and serial number, although much smaller and simpler than the EPC, can still provide a sufficient level of uniqueness. And although the analog interface of tags operating in the HF frequency band usually limits the communication distance to a few centimeters, long range readers that support ISO/IEC 14443 Type A and substantially increase the limited operating distance have successfully been demonstrated by Kirschenbaum et al. in [80], Hancke in [81] and Engelhardt et al. in [82]. In their papers the authors show that it is possible to actively communicate with an ISO/IEC 14443 Type A tag over an extended distance of approximately up to 25 cm, which is enough in practice when operating in crowded areas like an MRT train. When passively eavesdropping the communication between another reader and the tag, this distance increases to a couple of meters. Furthermore tags following the ISO/IEC 14443 Type A standard always respond with their UID to valid requests, regardless of who they come from. Together with an identity-UID association this could potentially enable movement or behavioral pattern tracking.

Nevertheless, the risks regarding the Standard Ticket in particular are moderate. Since its intended purpose are single MRT/LRT trips and a ticket with its UID is only used for

six top ups and at most for 30 days, a long term usage is not given and thus tracking is unlikely to be feasible let alone successful. Because of that, the threats mentioned above are hardly applicable in this case.

Apart from its UID also the entire memory content can be read without authentication, introducing a potential information leak. However, besides the two data packets, where further investigations are needed for a conclusion, no personal information is stored on a Standard Ticket, revealing no trivial privacy violation on the one hand but leaving the threat evaluation ultimately inconclusive on the other hand.

4.5 Security Evaluation

The discussion about security aspects is split up into two parts, one that focuses on the Standard Ticket's underlying chip and one that covers practical experiments to assess the proper usage of the ticket in the system.

4.5.1 Infineon my-d move

Write-Protection An important feature of the SLE66R01P is write-protection of the memory. Lock bytes are used to irreversibly change selected memory blocks to read-only, once their corresponding lock bit is set to one. A common practice would be to lock selected blocks and then finalize this process by locking the lock bytes themselves. If this last step is not done, a simple denial of service attack can be mounted, because left-out blocks that may need to be modified by the system can be write-protected subsequently. For instance, if the lock bytes in Service Area 1 were not read-only themselves, a Standard Ticket could easily be attacked by locking blocks 04_h to 08_h respectively $0a_h$ to $0e_h$, leading to a malfunction when entering or exiting through the automatic gates. This is because User Area 1 is used to store travel related data on blocks that are updated during every reader interaction. If those blocks were set to read-only, this update would fail. However, lock bytes on Standard Tickets are write-protected themselves, thus preventing the mentioned attack. Although the lock bytes in Service Area 2 are still writable, neither they themselves nor User Area 2 have been observed being used by the system.

OTP Lock bytes follow the concept of being one-time programmable, which in this case means that these bits are zero by default and once set to one, they cannot be changed back to zero. Consequently locked pages cannot be unlocked again, even if the lock bytes themselves remain writable. From a security perspective, this additionally limits the freedom of an attacker, even if the lock bytes are not locked themselves. Block 03_h applies this concept to an ordinary user memory block. Starting from all zeros, the content of each write command is OR'ed with the current block content, thus allowing bits only to flip to one. And while this mechanism could be used to count (e.g. flip a bit for every redeemed trip), it is not recommended to do so. Simply because of the fact that when a block remains modifiable, it cannot be properly protected from unauthorized write-access, potentially enabling denial of service attacks as described above. In case of the Standard Ticket, page 03_h is filled with constant data and protected from further change, thus preventing any malicious modification.

Password While lock bytes prevent memory pages from being modified, the password mechanism can restrict read as well as write access, but is limited to Service Area and User

Area 2. The password itself is 32-bit wide, which in practice is enough to make brute-force attacks inefficient. This is due to the slow communication interface and extended waiting periods after an incorrect password has been provided. The feature comes with a retry counter restricting the number of unsuccessful authentication attempts. In a my-d move product information sheet it is stated that if the retry counter is exceeded, further access to the memory is prevented [83]. Thus, a very simple denial of service attack is to flood the tag with invalid authentication attempts. If a system uses the password protected area, this will lead to a malfunction. As shown by Kirschenbaum et al. in [80] and Hancke in [81], this is possible with a custom long-range reader from a distance of approximately 25 cm. However, the biggest weakness of the password protection is its transmission in plain. If an attacker can sniff the communication between an official reader and tag performing a successful authentication, the password can simply be read from the logs. The feasibility of long range HF band eavesdropping attacks is discussed by Engelhard et al. in [82], which concludes that distances between one to three meters are realistic when targeting the fundamental frequency.

However, if the password is properly diversified for each ticket, it can add another obstacle to defend against denial of service attacks on the user memory area. Being the only authentication mechanism available on the my-d move, the password feature at least addresses the problem of unrestricted memory access that is inherent to many simple memory cards. Protection, however, is only available for User Area 2 and the value counter that is part of it. Again, since the Standard Ticket does not rely on the password feature, it is therefore not susceptible to any of the mentioned attacks.

User Memory What is being used on a Standard Ticket, is the user memory. The my-d move offers two areas, having 12 respectively 20 blocks to store data. They can all be write-protected with their corresponding lock bytes. Additionally User Area 2 can be read or read-write protected with the already evaluated password feature. Since the Standard Ticket only uses the first memory area, the travel related content can always be read without any restrictions. Even more, because the two data packets are not write-protected, travel related data can also be modified by anyone. This stems from the fact that the Standard Ticket can be used multiple times and enables a very straight forward denial of service attack that simply deletes the memory content and renders the ticket unusable. As stated previously, this is feasible from distances larger than defined in the standard.

Even though the data packets can be read, the meaning largely remains unknown. Their random nature and non-repetitive change during traveling suggest that travel related data is not stored with a simple encoding on the ticket. It is more likely that either some kind of scrambling is used, for instance travel data is stored encrypted, or that the two data packets are only references, for instance a (keyed) hash value, to the actual travel data stored in the back end system. Therefore, other approaches to continue the evaluation were chosen.

4.5.2 Back End System

Cloning A simple cloning attack is such a first step into a different direction. To test this attack and future ones, an own Android application has been developed that facilitates read and write operations performed on a Standard Ticket. It is run on a Google Nexus S smartphone with Android v4.1.2 and makes use of the NFC interface. During the

experiment, two Standard Tickets were bought with the intention to re-use the information from one card on the other. Of course this applies solely to elements that can be read and written, which are only the two data packets. The following list shows the steps of the experiment.

1. Buy two Standard Tickets at a General Ticket Machine.
2. Read the data packets from the first ticket and store them.
3. Overwrite the data packets of the second ticket with the stored ones.
4. Ride the MRT with the first and second ticket.

Using the first ticket worked, as it remained unchanged and simply represented a normal use case. However, trying to ride the MRT with the second ticket resulted in an error at the entry gates of the MRT station. Thus, the system is able to check whether the given data packets correspond to the card they are stored on and more specifically whether they match the card's UID and/or serial number. Therefore simple cloning attacks are prevented. The experiment also demonstrates that a denial of service attack, which modifies the data packets, is possible and renders the ticket invalid. This is can be considered a security policy violation as defined in the beginning of this thesis, since availability is a key requirement for a public transport system.

Reset An attack that also exploits the modifiable data packets in user memory is a reset or reuse attack. The idea is fairly simple and has for instance been presented by Benninger and Sobell at EUSecWest in 2012 [84]: record the data packets after the purchase of a ticket and use them again once a trip has been made. The following list shows the steps of the experiment.

1. Buy a Standard Ticket at a General Ticket Machine.
2. Read the data packets and store them.
3. Ride the MRT, both packets changed after the trip.
4. Write the previously stored packets to the ticket.
5. Ride the MRT again without paying.

Unlike the first experiment, this one succeeded. After purchasing a single trip from Boon Lay to Bukit Batok MRT, the data packets were saved with our custom Android application and the purchased trip was actually made. The changed data packets were saved after the trip and the original ones were restored. Then the same trip was successfully made again without reloading the ticket. During the experiment it was observed that the data packets on both trips changed to different values. Starting from the same memory content, the data packets ended up at entirely different values after both trips.

To illustrate this reset process, our custom Android application is shown in Figure 4.7 while it resets the ticket presented in Table 4.2. The leftmost picture depicts the memory content after the ticket purchase (see Table 4.2a). It is saved as 'State 1' in the application. The picture in the middle shows the difference between 'State 1' and the ticket content after the trip (see Table 4.2c), which is stored as 'State 2'. Clearly, the two data packets

differ between both states. In the figure, 'T' refers to the current memory content of the ticket and 'S' denotes the state it is compared to. A simple press of the 'Restore' button writes the currently selected state and consequently the previous memory content back to the ticket, as shown in the rightmost picture. This clearly illustrates the simplicity and ease of use regarding the practical implementation of the attack.

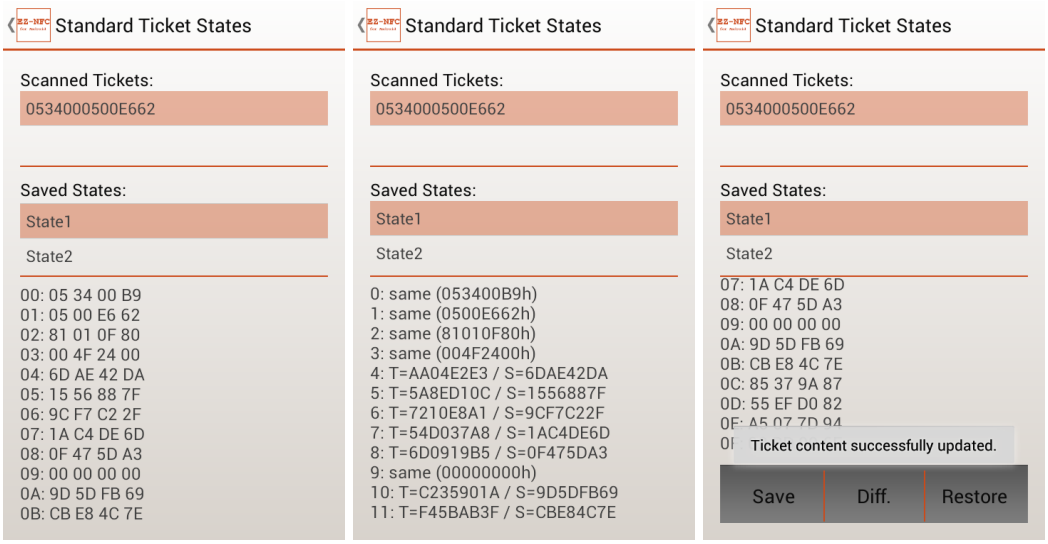


Figure 4.7: EZ-Link Standard Ticket Reset with Custom Android Application (Inverted Color).

An improvement of the attack can be derived from Section 4.3. There it is mentioned that beginning and end of a selected route are not enforced by the system, thus shifting the focus to the value that is loaded onto the Standard Ticket. Because of that, purchasing a route with a maximum fare ensures that any trip can be made with a reused ticket.

The tools that are needed for the attack include an NFC-enabled smartphone and an application that performs the reset. With an increasing support of NFC on today's smartphones, many commuters already fulfill this requirement without further investment. The development and usage of custom Android applications is free of charge, communication with an Infineon my-d move is publicly documented and supported conveniently out of the box from Android API level 10 onwards. Therefore, the requirements for the attack are minimal. Furthermore, resetting a Standard Ticket can be done in several seconds and hardly raises any suspicion in public, if performed with a smartphone.

The conditions that apply to the usage of a Standard Ticket, however, limit the practical impact of the attack. Since a trip is only valid on the day of purchase, at least one fare has to be paid everyday and latest after six top ups a new ticket has to be bought. However, due to the incentives, S$ 0.1 can be saved if the ticket is used to its full extent. Note that these assessments are done without any further knowledge of the back end system, which is in theory able to prevent this attack altogether.

Because of the implications that arise with practical experiments of that kind, no further investigation was done. Once more the author wants to stress that all trips were additionally paid for and thus no financial damage was done during the experiments. The sole purpose of the experiment was to identify possible security policy violations as defined in the beginning of this thesis, which has been achieved in this case.

4.5.3 Countermeasures

In the previous section it has been shown that the Standard Ticket suffers from a problem inherent to many simple memory cards: lack of authentication. The unrestricted read access to User and Service Area 1 of the my-d move allows an attacker to read the entire memory content that is used by the system. The current state of knowledge suggests that this has been accounted for in the system design, since the travel relevant data is not stored in plain on the ticket and thus could not be decoded or interpreted. However, the user memory, where this data is stored, again lacks a proper authentication mechanism that prevents unauthorized parties from writing to the memory. Consequently, it has been shown that a denial of service attack is possible. Sticking with the current solution, the password feature of the my-d move can be used to add another layer of security to the ticket, if the two data packets are moved to User Area 2. Although the password suffers from plain transmission, it can withstand brute-force attacks and prevent quick access by an attacker. This countermeasure hinders the previously mentioned scenario of an attacker covertly accessing a ticket for a short period of time in a crowded area. Note that the retry counter is not advisable to use, as it introduces yet another denial of service scenario, and that the password is recommended to be properly diversified for each ticket, for instance derived from a master key that is stored in the terminals and from the UID and/or serial number of the ticket.

The second security policy violation is caused by a successful reset attack of the Standard Ticket. It illustrates very well how the back end system must compensate the ticket's limitations. Since currently an attacker can update the memory content, the terminal has to query the back end system, whether a ticket has already been used or not. A similar check stems from the fact that due to the unrestricted read access, an attacker can completely clone a Standard Ticket with a custom RFID emulator. For the terminal, the clone is indistinguishable from the original. Such RFID emulators can be constructed with low-cost hardware. An overview is given by Verdult et al. in [85]. Although the password feature can again add a layer of security in this case, as it introduces another obstacle for an attacker to entirely read the content of the ticket, the problem itself lies rather in the terminal software and back end system, since the ticket reset checks have seemed to fail in our experiment. Without further knowledge about the back end system, however, a constructive solution for this specific problem cannot be given.

The limitations of simple memory cards like the my-d move introduces great challenges regarding their secure usage in a public transport system. Because of missing cryptographic mechanisms that can be used to achieve more advanced levels of security and privacy, switching to a more feature-rich smart card platform is consequently a recommended long-term solution.

4.6 Conclusion

In this chapter strong software and hardware based indicators are given that the underlying chip of an EZ-Link Standard Ticket is a my-d move SLE66R01P by Infineon Technologies AG. This is also supported by a media briefing held at Infineon Technologies Asia Pacific in early October 2013 [86]. It is observed that most of the chip's features are not used by the system and that additionally to the chip's UID a system-specific serial number is stored as well as printed on the ticket. An analysis of the ticket's memory content during traveling showed that travel related data is stored in two 20-byte data packets that are of

random nature. Thus, no trivial encoding of the data packets could be found.

Despite the facts that each ticket has two identifiers, that the entire memory content can be read without any authentication step and that all this can be done without the knowledge of the ticket owner from a considerable distance, the current research state does not exhibit any privacy violation with practical impact. This is due to the usage limitations of the ticket and the fact that according to the current state of knowledge no personal information is stored on a Standard Ticket.

From a security point of view, the Standard Ticket is properly put on top of the my-d move platform for the most part. Of course, many of the features, for instance the value counter, remain unused. However, write-protection and finalized lock bytes are properly made use of. Experiments showed that simple cloning of a Standard Ticket is detected, because the data packets can be related to the card they are stored on. However, the same experiment also illustrated that a simple denial of service attack can be mounted against the ticket by overwriting the data packets. This renders the Standard Ticket invalid and thus impairs the availability of the system. A short-term solution is to move the data packets to the password protected memory area.

In addition, the re-usage of the initial data packets after the purchase is not detected by the system. Although carrying some practical limitations, this enables attackers to use a Standard Ticket for multiple journeys without actually paying for them. This is undoubtedly a security policy violation that can easily be exploited with everyday equipment and without raising suspicion in public. However, the problem lies rather with the back end system than with the Standard Ticket itself. The compensation of the ticket's limitations by the back end system can in the long run be circumvented by changing to a smart card platform offering more advanced security mechanisms.

However, without knowing more about the data packets and consequently about the back end system, the final assessment regarding the Standard Ticket's security and privacy aspects remains partially inconclusive.

Chapter 5

ez-link Card

Similar to the previous evaluation, this chapter contains an introduction to the ez-link Card followed by a short section about the underlying chip and its properties. The part about its usage in the EZ-Link system includes a description of the CEPAS standard, which is closely examined regarding its privacy properties. During the security evaluation, the performed side-channel analysis is presented followed by a short conclusion.

5.1 Introduction

The ez-link card and the electronic purse that is stored on it are the core elements of the EZ-Link system. Unlike the Standard Ticket it can be used over an extended period of five years and can hold up to S$ 500 [6]. It can be bought at ticket offices in train stations for S$ 12, which includes S$ 7 that are already stored on the card and leaves S$ 5 as a non-refundable card price [87]. As already shown in Chapter 3.1, ez-link Cards can be charged in many ways and used to pay for various services and goods. They also differ regarding personalization, co-branding or membership association. But what all of them have in common is the electronic purse, a tool to maintain an account balance.

The concept of electronic payment within the EZ-Link system is referred to as Stored Value Facility, the idea behind which is quite straight forward. Customers pay real money to a third party and in turn get a digital value representing the paid amount transferred to their electronic purse. In the case of EZ-Link, this purse sits on top of a contactless smart card and basically allows authorized parties to in- and decrease the stored balance over a well-defined interface, which is specified in the CEPAS standard. This specification has been implemented on ez-link Cards since 2008 [88]. Customers can then charge their electronic account at merchants in exchange for services and goods. The merchants collect the amounts received and in turn exchange them for real money again at the third party that held it in the meantime.

Since this concept is not limited to contactless smart cards, there are actually two separate topics that need to be addressed in this chapter. First, there is the current smart card platform that runs an actual implementation of the electronic purse as a smart card applet on every ez-link Card. Consequently a short discussion about the platform and its properties is needed. But more importantly, because the electronic purse is also implemented on other platforms, for example on the NETS FlashPay card [4] and on NFC-enabled SIM cards [48], the specification of the purse must be evaluated in detail.

(a) Front View. (b) Back View.

Figure 5.1: ez-link Card.

Visual Appearance Figure 5.1 shows the front and back side of a non-personalized, non-co-branded, non-membership ezlink Card. The visual appearance of the front side is dominated by the EZ-Link logo and a colorful design, which has been observed in colors like orange, blue and black. However no difference was noticed based on the color of the cards. The back side is featured with legal and customer related messages like where to turn to for further information and assistance. Beneath that there is a series of logos, starting from the left with the Singapore public transport and ending to right with the Contactless e-purse Application logo.

Furthermore there is a decimal number printed on the bottom of the back side. It is the so-called Card Application Number (CAN), which is part of the CEPAS standard and explained in more detail in Section 5.3.1. The meaning of the string beneath the last four digits of the CAN is unknown.

5.2 Platform

This section contains a short discussion about the ez-link Card's underlying chip viewed from a software and hardware perspective. Unlike the Standard Ticket, the focus is not put on the identification of the actual chip type but rather on the discovery of additional information about the implementation.

5.2.1 Identification

Software For a first analysis of the underlying chip, again a Google Nexus S smartphone running Android v4.1.2 together with NFC TagInfo (v1.11) [70] and NFC TagInfo by NXP (v2.00) [71] is used. The results shown in Figure 5.2 give some first insights. Additionally the ez-link Card is also analyzed on a Microprox MP300 TCL2 universal RFID reader. The results are shown in Tables 5.1 and 5.2.

The ez-link Card implements ISO/IEC 14443 Type B up to layer 4 as illustrated in the rightmost picture in Figure 5.2. The content of a full Answer-To-Request Type B (ATQB) is given in Table 5.1. The PUPI is randomly generated, since always different values have been observed during the analysis. This can be seen on the screenshots, where the PUPIs are $\texttt{9d0c8542}_h$ (in the figure given in reverse byte order) and $\texttt{97ab8782}_h$, and in the table, where it is $\texttt{30b580bf}_h$. Also included in the ATQB is the Protocol Information field containing the supported transmission parameters of the card. The most important

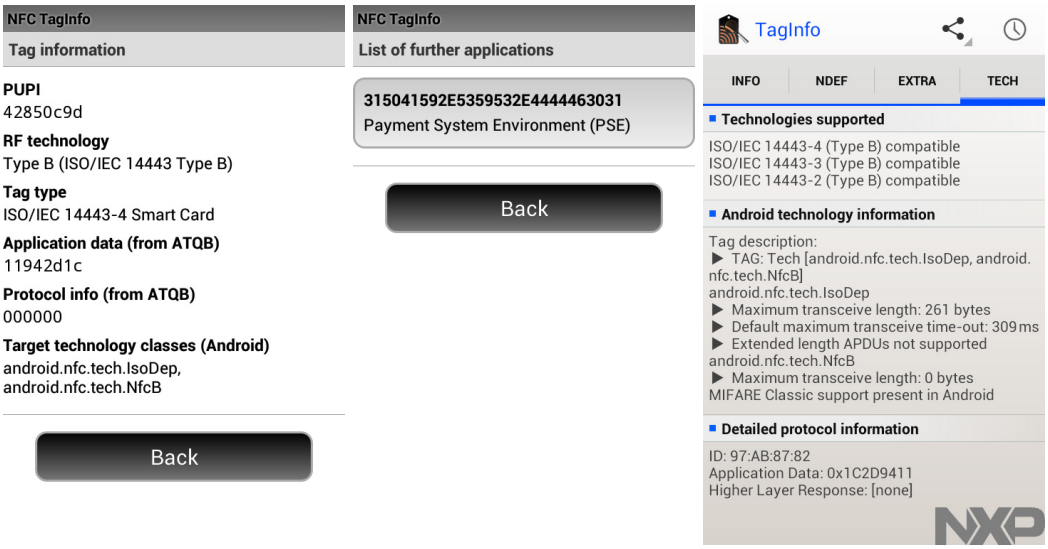


Figure 5.2: ez-link Card Identification with an NFC-enabled Smartphone (Inverted Color).

facts are that the card supports a data rate of up to 847 kbit/s in both directions as well as a maximum transmission frame size of 128 bytes and as expected is compliant to ISO/IEC 14443-4. Also, a standardized coding of the Application Data field is used. Consequently, $\texttt{1c}_h$ is the AFI, which codes the application family of the ez-link Card. According to ISO/IEC 14443-3, the nibble $\texttt{1}_h$ is mapped to the transport sector (mass transit, bus, airline, etc.) whereas the lower nibble $\texttt{c}_h$ codes a proprietary subfamily. The bytes $\texttt{2d94}_h$ are the Type B CRC checksum of the Application Identifier (AID), whereas the byte $\texttt{11}_h$ indicates that there is in total one application installed on the smart card that corresponds to the given AFI. [53]

AQTB $\texttt{5030b580bf1c2d9411f77185}_h$			
$\texttt{50}_h$	$\texttt{30b580bf}_h$	$\texttt{1c2d9411}_h$	$\texttt{f77185}_h$
Fixed for ATQB	PUPI	Application Data	Protocol Info

Table 5.1: ISO/IEC 14443-3 Answer To Request Type B Decoded [53].

As the middle picture in Figure 5.2 shows, this application is inside a so-called Payment System Environment (PSE) directory definition file. This structure is part of the file system defined in the EMV specification. The idea behind it is to simplify application selection by arranging the files of all payment card applications in a well-defined group hierarchy with the PSE file on top. This file always has the reserved AID '1PAY.SYS.DDF01', which is given in its ASCII-encoded hexadecimal representation ($\texttt{315041592e5359532e4444463031}_h$) in the same picture. [89]

The next indicator that reveals further details about the smart card platform is the ATR. It is again recorded with the MP300 and shown in Table 5.2. The initial character $\texttt{3b}_h$ denotes a direct transmission convention whereas the format character $\texttt{7e}_h$ specifies three interface bytes TA_1, TB_1, TC_1 as well as 14 historical bytes to come. Amongst other things, the interface block $\texttt{940000}_h$ denotes a maximum external clock frequency of 5 MHz. [58]

Since historical bytes typically depend on the operating system of the smart card, their meaning is not completely revealed. The first part 57443565_h can be interpreted as the ASCII-encoded string 'WD5e', whereas the following two bytes $f45f_h$ remain unclear. These leading six bytes are the same on all analyzed ez-link Cards. The last eight bytes, $b20d52715b031d3f_h$, are the so-called Card Serial Number (CSN), which is part of the CEPAS standard and explained in more detail in Section 5.3.1.

ATR $3b7e94000057443565f45fb20d52715b031d3f_h$			
$3b_h$	$7e_h$	940000_h	$57443565f45fb20d52715b031d3f_h$
Initial Byte	Format Byte	Interface Byte	Historical Bytes

Table 5.2: ISO/IEC 7816-3 Answer To Reset Decoded [58].

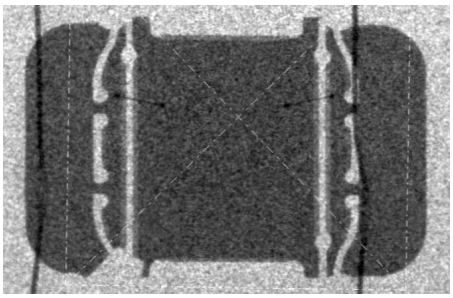
Another interesting aspect from the software perspective is the operating system the smart card runs. As indicated in [90], the CEPAS compliant card operating system TimeCOS ® is running on ez-link Cards. It is developed by Watchdata, whose international headquarters are located in Singapore. The string 'WD5e' found in the historical bytes is assumed to be linked to the company. Furthermore it is stated that Watchdata supplies CEPAS compliant smart cards that passed a Common Criteria EAL5+ certification. [91]

Hardware The ez-link card's outermost package is made out of plastic and features the visual design shown at the beginning of this chapter. In order to identify at which approximate location of the card the IC resides, a strong flashlight can be used. If it is turned on and held against the card, the light can be seen on the other side of the card as it partially shines through it. The antenna close to the edges of the card and the IC can then both be recognized as dark shadows in the bright area where the light illuminates the card. The rectangle in Figure 5.3a marks where the IC is approximately located on an ez-link Card. Figure 5.3b shows an X-ray CT scan of this area. The IC is placed close to the rectangular middle piece of the large dark gray shape that is framed by two metal connectors with rounded edges. The two black vertical lines that touch these connectors are the wires of the antenna. The two small black horizontal lines that seem to join the middle piece with the antenna connectors are bonding wires attached to the die of the IC.

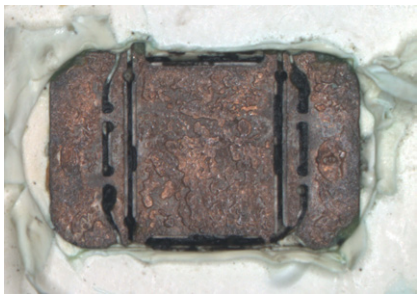
To remove the IC from the plastic card, a Stanley knife can be used to cut the area where the IC is located and to carefully remove the plastic layers of this smaller piece. This works quite well, since the plastic material is fairly brittle. Another approach is illustrated in Figure 5.3c, where fuming nitric acid was used to carefully dissolve some of the plastic and residues were cleaned with acetone afterwards. As a result, the metal structure identified in the X-ray CT image is now accessible. The IC is stuck underneath it, covered in black epoxy resin, similar to the Standard Ticket. The epoxy cover is also dissolved with fuming nitric acid revealing the die front side as illustrated in Figure 5.3d. It can be clearly seen that out of the seven bond pads, only two are used to connect the antenna to the IC. This might indicate that the chip supports a dual interface (contact-based and contactless).

When zooming in on the die as illustrated in Figure 5.3e, neither internal structures nor any labels or marks can be identified on the entire die. Instead, a pattern of horizontal lines appears as shown in Figure 5.3f. A very likely explanation for the structure of this topmost layer is shielding, which is a way of defending against invasive attacks on smart

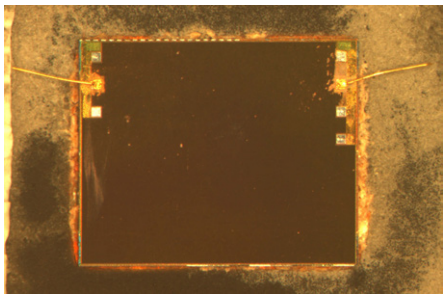
(a) Plastic Shell.



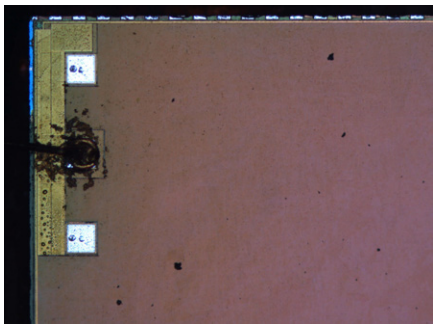
(b) IC X-ray.



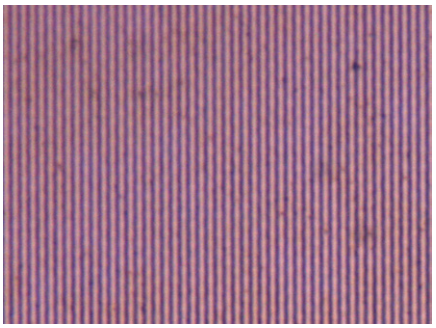
(c) IC Exposure.



(d) Die Front Side.



(e) Die Top Left Corner.



(f) Die Surface Magnified.

Figure 5.3: ez-link Card IC Decapsulation.

cards and explained in [57]. If the die is not shielded, the internal structure of the circuit is susceptible to optical inspection, to measurements of charge levels during operation or even to physical access of wires on the die. This can be prevented by placing one ore more metalization layers, also called meshes, on top of the die and monitoring them for integrity. For instance, shields can be built in a way that they supply power to components on the chip. Consequently, if they are removed by an attacker, the chip simply stops working. Another solution is to use opaque shields and detect their removal with a phototransistor.

In 2010 Christopher Tarnovsky presented the deconstruction of an Infineon SLE 66 series processor. As one of the first steps, he removed a shielding mesh on the top metal layer of the die that had a very similar look and structure as the one shown in Figure 5.3f. This is an indicator that shielding might be used in this case. [92]

5.3 Implementation

So far, the concept of the EZ-Link system has been addressed and the role of the electronic purse identified. It is used to hold an account balance and act as a gate keeper so that only authorized parties can access and consequently modify the balance. In order to facilitate interoperability, the design of the purse is defined in the Singaporean Contactless e-purse Application Specification or CEPAS. Since the electronic purse is the core application on the ez-link Card, its specification is discussed in this section.

5.3.1 Contactless e-purse Application

The Singapore standard SS 518:2006, also known as CEPAS, is the further development of the national Specification for Stored Value Card Application known as SS 468:1999. CEPAS defines a command set for e-purse applications, of which two versions exist (v1.0 and v2.0). Since all analyzed ez-link Cards implement the latest version, the discussion will focus only on v2.0. The main commands defined in the standard are used to decrease (Debit) or increase (Credit) the purse balance as well as to retrieve status information about the purse (Read Purse). Because of the definition of additional smart card commands, the specification largely follows ISO/IEC 7816-4 but also references ISO/IEC 9797-1 regarding block cipher operation as well as other smart card and e-purse specifications that can be found in the national standards SS 372, SS 467:2002 and SS 484:2000. The goal of CEPAS is to enable atomic operations such that debit and credit transactions can be made with one command only. This means that updates to the purse are either applied completely or not at all and that an inconsistent state must not occur. The reduced amount of commands and their atomic nature aim to make transactions more robust and thus more suitable for contactless environments. Furthermore the specification tries to facilitate implementation on a traditional file system as well as a fixed sized sector structure. [93]

Card There are two identification numbers for each card, the 8-byte Card Serial Number (CSN) set by the card manufacturer and the 8-byte Card Application Number (CAN) set by the card issuer. Furthermore, in CEPAS v2.0 a 2-byte issuer ID that is associated with every purse is part of the CAN. [93]

Purse Although the command interface of the purse is in the main focus of the CEPAS standard, some properties of the purse are defined in addition to it. A purse consists of three items, a purse file, which contains the actual electronic purse, a key file, which stores the keys needed for the cryptographic operations during the purse commands, and a transaction log that keeps track of past transactions. According to the standard, a purse file shall contain the following elements. [93]

- *File ID* ... short file identifier of the purse file.
- *Purse Balance* ... monetary value stored in a 3-byte signed integer.
- *Last Transaction* ... record of the last debit or credit transaction.
- *Creation Time Stamp* ... date and time when the purse was created.
- *Maximum Value* ... biggest allowed value of the purse balance (minimum is zero).

Furthermore, a purse file is only accessible through the defined purse commands and initial personalization commands, the latter of which are not part of the CEPAS standard, since purse initialization is the responsibility of the card issuer. An error detection mechanism shall ensure the integrity of the purse, of which multiple can exist per smart card. [93]

A key file contains a set of key records, each consisting of a 16-byte key used for the Triple DES cipher (2TDES, two unique keys) and a corresponding key type, which defines what the key is allowed to be used for. There is one type for each elementary operation performed during purse commands: debit, autoload, credit and signature. However, a real distinction is only made between credit, autoload and the rest. Credit keys are the only ones that are used to increase the purse balance. Autoload is an optional feature that automatically reloads the purse balance during a debit transaction, if the amount deducted exceeds the current balance. This can only be done with an autoload key, which is other than that identical to a debit key regarding its usage. Apart from their names, there is no difference specified between a debit and a signature key. Each key file is addressed with a 5-bit short file identifier (SFI) and each key record is selected with a 1-byte key number. Thus, in order to choose a key, the SFI together with the key number must be provided. This tuple is labeled in general (Kf, Kn), with Kf being the key file SFI and Kn being the key number. Keys with a certain type have a corresponding prefix, for example DKf is the SFI of a debit key and SKn refers to the key number of a signing key. It is important to note that key files are not bound to a specific purse file but are selected individually in each command instead. [93]

Apart from the purse and the keys, there is also a transaction log specified in the standard. It is used as a cyclic historical buffer that stores the last debit and credit transactions in a first-in-first-out manner. Table 5.3 shows the structure of a transaction log entry. [93]

Transaction Log Record			
Type	Amount	DateTime	User Data
1 byte	3 bytes	4 bytes	8 bytes

Table 5.3: CEPAS Transaction Record Format [93].

Debit, Credit The debit and credit transactions are very similar to each other and both trigger multiple tasks with one message exchange. The purse balance is de- or increased, a receipt of the transaction is included in the answer and a new entry is added to the transaction log. Before issuing one of the commands, an 8-byte random number must be requested from the card with a standardized Get Challenge command. This is necessary, since session keys are used for the cryptographic operations rather than the static card keys mentioned above. Both the terminal and the card supply random numbers in order to create them, which is depicted in Figure 5.4. [93]

In the case of a debit transaction, the type of the fixed key can be debit, autoload or signature. In contrast to that, a credit command only uses dedicated credit keys. Under the assumption that the terminal has the same set of keys that is stored on the card, the simple exchange of random numbers is sufficient for the session key creation. In the debit case, the 8-byte terminal random number 'Term-Rnd' is the first block that is encrypted by the 3DES cipher in Cipher-Block-Chaining (CBC) mode. This mode requires that every input to the cipher is XOR'ed with the previous output of the cipher. Since this

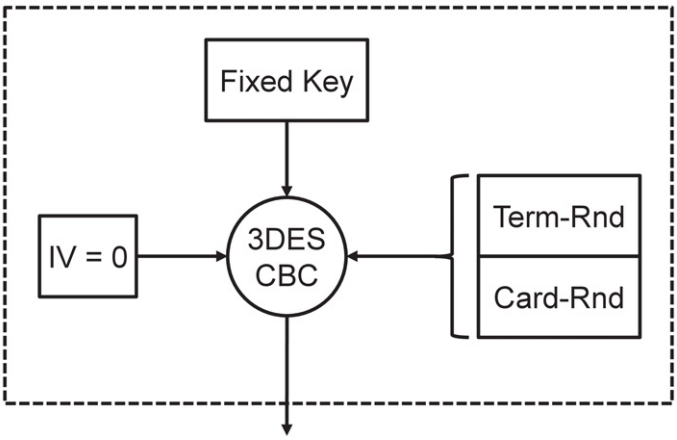


Figure 5.4: CEPAS Session Key Creation from Random Numbers [93].

would not work in the first round, a so-called initial vector (IV) is used, which is zero for every session key creation. The second 8-byte block that is encrypted in the debit case is the card random number 'Card-Rnd'. In the credit case, the same procedure is followed except that the first encrypted block is the card random number whereas the second block is the terminal random number. [93]

Since multiple tasks are triggered with debit and credit commands, their structure is quite complex. In Table 5.4 the details of the debit and credit transactions APDUs are given, which look very similar for the most part. However, there are important differences. Note that for a debit command, in total two keys must be specified, whereas a credit transaction needs three keys. Parameter P2 of the debit command specifies the debit options, with which a deduction and autoload of the purse, a disabling of the purse expiry check or even a deactivation of the purse itself can be triggered. For a credit transaction, parameters P1 and P2 specify the second credit key tuple (CKf_2, CKn_2). The command data starts in both cases with Pf, the short file identifier of the purse file, followed by the key identifiers for either the debit key (DKf, DKn) or the first credit key (CKf_1, CKn_1). Similar, an additional signing key is given with SKf and SKn after that. The terminal random number R_T and the user data part TR_{User} of the transaction log record (see 5.3) surround the so-called command cryptograms. They are encrypted using an initial vector of zero with session keys derived from DKf, DKn or CKf_1, CKn_1 and contain the actual transaction details. [93]

At the beginning there is in both cases a 4-byte number called Terminal Reference Parameter (TRP), which is not specified in the CEPAS standard and can be freely chosen by the implementation. It is followed by an ISO/IEC 14443-3 Type B CRC, which is for a debit command calculated over TRP, debit options, Pf, SKf, SKn and the transaction log record. For a credit command, the same fields except the debit options are used. SKf and SKn are redundant to the ones given in the command data. The transaction header TR_{Head} contains the transaction type, amount and time (see 5.3) and together with the user data TR_{User} forms the complete transaction log record. The biggest difference between a debit and a credit command is that for the latter everything in the cryptogram, except for the TRP and the transaction time, is additionally encrypted directly with the second credit key given by CKf_2 and CKn_2. No session key is derived for this additional operation. [93]

Command	CLA	INS	P1	P2	L_c	Data	L_e
Debit	90_h	34_h	RFU	Options	25_h	see 5.4b	18_h
Credit		36_h	CKf_2	CKn_2			

(a) Command APDU.

Command	1 byte	1 byte	1 byte	1 byte	1 byte	8 bytes	16 bytes	8 bytes
Debit	Pf	DKf	DKn	SKf	SKn	R_T	see 5.4c	TR_{User}
Credit		CKf_1	CKn_1					

(b) Command Data.

Command	4 bytes	2 bytes	1 byte	1 byte	8 bytes
Debit	TRP	CRC	SKf	SKn	TR_{Head}
Credit					

(c) Transaction Cryptogram.

Command	3 bytes	5+8 bytes	8 bytes
Debit	New Balance	Signed Cert.	Counter
Credit			

(d) Response APDU.

Table 5.4: CEPAS Debit and Credit Commands in Detail [93].

If the internal updates are successful, the card responds in both cases with a receipt cryptogram, which is encrypted with the same session key as the command cryptogram, but with an 8-byte counter value as the initial vector. This receipt contains the updated purse balance, a so-called signed certificate and the same 8-byte counter value. The certificate consists of the encrypted transaction header TR_{Head} using a session key derived from the signing key specified by SKf and SKn. However, this session key is not created using the concatenated random numbers but rather using the debit options, the updated purse balance, the TRP and again the 8-byte counter data. The initial vector, however, is zero. This signed certificate is then repeated in the receipt, one time taking the five most significant bytes and one time all eight bytes. The counter data actually consists of three individual counters. The purse transaction counter has three bytes and is incremented by one during each debit and credit command, thus rendering the card unusable after 2^{24} or 16,777,216 transactions. The 3-byte add-value counter is only incremented during credit and autoload operations whereas the 2-byte modify purse counter gets incremented if purse data is directly updated by an external command. [93]

In addition to the already quite complex operations that accompany debit and credit transactions, both commands can be also used to perform maintenance tasks. The debit command can be used to deactivate the purse by setting a flag in the debit options. The credit operation can act as a write command to update raw data in the purse elementary file. For this purpose the transaction record field is used to store the updated data and its destination. [93]

Read Purse In contrast to debit and credit commands, the purpose of the read purse command is to retrieve information about the purse itself rather than to perform a task with it. In order to do so, there are actually three different sets of parameters to access

two separate sources of information. The first two sets, 'Status (w)' and 'Status (w/o)', retrieve purse status information including the implemented CEPAS version or creation and expiry dates. The 'Status (w)' variant additionally triggers the creation of a reduced version of the last receipt cryptogram, which is appended to the purse status. Here the last signed certificate is encrypted together with the 8-byte counter data using a session key derived from a fixed key chosen with Kf and Kn in the command parameters (see Figure 5.9). The session key creation is identical to the debit command. The third set, 'Log', is used to access the transaction log and read out its content. [93]

Command	CLA	INS	P1	P2	L_c	Data	L_e
Status (w/o)	90_h	32_h	Pf	RFU	00_h	see 5.5b	00_h
Status (w)					$0a_h$		00_h
Log					01_h		see 5.5c

(a) Command APDUs.

Command	1 byte	1 byte	8 bytes
Status (w/o)	-	-	-
Status (w)	Kf	Kn	R_T
Log	Record Offset	-	-

(b) Command Data.

Command	variable size
Status (w/o)	min. 63 bytes status info
Status (w)	min. 81 bytes status info
Log	L_e bytes of the transaction log

(c) Response APDUs.

Table 5.5: CEPAS Read Purse Command in Detail [93].

Table 5.5 shows the read purse command with its different parameters. The 'Status (w/o)' variant of the read purse command does not request the cryptogram appendix and therefore does not trigger any cryptographic operation. Consequently sending a Get Challenge command beforehand is not required. Since no additional data is needed in the command APDU, the Data field is missing, but L_c is still given as zero. Because of this, the 'Status (w/o)' APDU structure does not match any of the cases illustrated in Figure 3.5 and thus does not follow ISO/IEC 7816-4, which states that for a zero-length Data field L_c is absent. In practice, this could be circumvented by using an extended L_e field and for instance request 65,536 bytes instead of 256. However, this might differ from terminal to terminal. Because the answer can vary in its length, L_e is chosen to be zero to indicate that the terminal simply expects all available data. However, a minimum of 63 bytes is always sent. [93]

The only real difference from the first variant to the others is the content of the Data field. In the 'Status (w)' case, it contains the identifiers Kf and Kn, which can point to a debit, autoload or signature key, and the 8-byte terminal random number R_T, which is again used for the session key creation. Of course, a Get Challenge is mandatory here before sending the read purse command. Since the answer is extended by the cryptogram appendix, a minimum of 83 bytes can be expected with an L_e of zero. Other than that, this

variant includes the exact same purse status information as the 'Status (w/o)' case. The second source of information accessible with the read purse command is the transaction log. This third variant can be used when including only one byte in the Data field of the command APDU. It is an offset and defines the number of transaction records that should be skipped when reading the log. The number of bytes to read from the log can be specified with the L_e field. Again, no cryptographic operations are involved and a preceding Get Challenge is not necessary. [93]

5.3.2 EZ-Link Purse

Now that the underlying standard of the electronic purse that is stored on an ez-link Card has been explained, this section will focus on the actual implementation of this CEPAS compliant purse.

Fare Paying for public transport is a major use case of the ez-link Card, but not all means of transport use the purse in the same way. This is because traveling with the MRT or LRT is different from going by bus or taxi and due to its CEPAS compliance the purse can adjust to that. The following conclusions were drawn after using different means of transport and analyzing the transactions on the ez-link Card.

When using the MRT/LRT system, access to the trains is controlled by gates that only open, when an electronic payment card is presented with sufficient value on it. Thus, the system remembers at which gate a commuter entered and deducts an automatically calculated fare at the exit gate. Since it is not possible to enter or exit the station other than through these gates, fare evasion is very well prevented. Consequently one trip adds only one debit transaction to the log.

The situation is different on a bus. Commuters are advised to enter the bus in the front, where the driver can make sure that everyone checks in with their card having sufficient value on it. Otherwise, a traditional cash payment is possible as well. However, when commuters leave the bus, there is no way of enforcing that everybody checks out again. Therefore, every time a commuter enters the bus, the maximum fare from the current to the last station is deducted, causing one debit transaction to be added to the log. It is the commuters' responsibility now to check out again, since a refund is given, if the actual fare is smaller than the one previously deducted. The partial refund adds another transaction, increasing the balance again. The whole procedure is done to prevent fare evasion also on the bus.

Paying for a taxi ride is similar to making purchases in convenience stores, it is just a general deduction with a custom value specified by the driver or merchant. Naturally, this adds only one debit transaction to the log.

EZ-Online Since analyzing the wireless communication between an authorized terminal and an ez-link Card is limited to places where these terminals are, for instance MRT stations, taxis or convenience stores, alternatives are welcome for a first look.

EZ-Online is a web-based service that allows reloading an ez-link Card conveniently from home, all that is needed is the official EZ-Online reader that is plugged via USB into the computer [41]. The reader used during this analysis turned out to be the model ACR122U by Advanced Card Systems Ltd., which is an off-the-shelf and inexpensive contactless smart card reader.

In [8] Kerschbaum et al. discuss the process of EZ-Online and introduce the idea of recording the communication going over the USB interface, with which the RFID reader is connected to the PC. They show that it is possible to see the plain communication and to get an insight of how an ez-link Card is used by the back end system, since the software running on the PC and the reader simply act as a communication relay between the card and the back end system. This means that no cryptographic keys have to be stored in the application or on the reader in order to provide this top-up service.

The software that implements the EZ-Online process is a Java application called JHunter. When examining its communication with the reader, the read purse commands can be observed. Table 5.6 shows the captured 'Status (w)' and 'Log' variant of the read purse command.

Command	CLA	INS	P1	P2	L_c	Data	L_e
Status (w)	90_h	32_h	03_h	00_h	$0a_h$	$\boxed{14}_h$ $\boxed{03}_h$ $\boxed{a4f52d7f31a22cac}_h$	00_h
Log					01_h	00_h to $1d_h$	10_h

Table 5.6: Read Purse Commands USB Traffic Capture.

The recorded commands clearly show that the SFI of the purse file, Pf, is 03_h. The 'Status (w)' command indicates that a debit, autoload or signature key is located at Kf = 14_h and Kn = 03_h, which is emphasized with framed boxes in the table. After these IDs there is an 8-byte random number as expected. The 'Log' command is used to read transaction log entries separately, which is why the offset given in the Data field varies between 00_h and $1d_h$. This indicates that at least 30 transaction records are stored on the card.

Purse Identifier CEPAS allows multiple purses to reside on one smart card. To distinguish them, the purse short file identifier Pf is used. In the above example, EZ-Online uses Pf = 03_h. In order to find all available purses and thus all valid identifiers, a brute-force approach is used to simply try all 256 possibilities for Pf with the help of the 'Status (w/o)' command. Although an SFI has only five bits, all values are included in the search to be as comprehensive as possible. The result is that only a purse SFI of 03_h triggered a valid response, all other values yielded a status code of $6b00_h$. Since the meaning of the status codes is not specified in the CEPAS standard, the general meaning defined in ISO/IEC 7816-4 and shown in Table 5.7 is used. Consequently, the value $6b00_h$ indicates wrong parameter bytes P1 and/or P2. Since Pf is given as P1 in the command APDU, this fits the result and suggests that there is only one purse installed on the analyzed ez-link Card with a purse file identifier of 03_h. This purse is consequently used in all following investigations.

Code	**Meaning**
6982_h	Security status not satisfied
6985_h	Conditions of use not satisfied
$6a80_h$	Incorrect parameters in the command data field
$6a82_h$	File or application not found
$6b00_h$	Wrong parameters P1-P2
9000_h	Normal processing

Table 5.7: Selected Status Codes According to ISO/IEC 7816-4 [59].

Key Identifier A similar experiment was done in order to reveal the keys available to the EZ-Link purse. Since each key is addressed by a 5-bit key file identifier Kf and a 1-byte key number Kn, there are 8,192 possibilities for the tuple (Kf, Kn). Again, to be as comprehensive as possible, all eight bits of the key file SFI and therefore all 65,536 values are included in the search. In order to identify the valid combinations, a brute-force approach is used to try all possible tuples for the read purse ('Status (w)'), debit and credit command. In the read purse case this is straight forward, since an invalid tuple does not trigger an answer. However, a valid debit or credit command cannot be performed, since the actual values of the keys are not known and thus no valid cryptogram can be created. This is why both commands eventually return an error code. Nevertheless, the card has to check whether a correct key is chosen in order to perform the decryption of the cryptogram, which is why the card still gives usable feedback about the key choice. Other than that the commands are crafted to look valid to the card. The terminal random number as well as the transaction user data contain random values and for the debit command the debit options are set to 00_h. The signing key identifiers (SKf, SKn) were found to not affect the results given in Table 5.8, regardless of their value. This may be because the signing key is used during a later stage of the debit and credit operations, which is not reached due to the card aborting earlier.

The results of the experiment are shown in Table 5.8. Overall, 10 keys were discovered in four key files with SFIs 12_h, 13_h, 14_h and 15_h, each containing either two or three keys. All other combinations yielded a status code of $6a82_h$ for each of the commands. According to Table 5.7, this error code suggests that there was no corresponding key file found.

Considering the *read purse command* case, a debit, autoload or signature key must be provided in the 'Status (w)' variant. Out of the 10 keys, eight triggered a valid response of purse information followed by a status code of 9000_h indicating a successful operation. Keys (14_h, 01_h) and (14_h, 02_h) yielded the error code $6a80_h$. Since credit keys are not allowed to be used for a read purse command, this is a first indicator suggesting that those two keys are credit keys.

<table>
<tr><th colspan="2">Keys</th><th colspan="3">Responses</th></tr>
<tr><th>Kf</th><th>Kn</th><th>ReadPurse</th><th>Debit</th><th>Credit</th></tr>
<tr><td>14_h</td><td>01_h</td><td rowspan="2">$6a80_h$</td><td rowspan="2">6985_h</td><td rowspan="2">6982_h</td></tr>
<tr><td>14_h</td><td>02_h</td></tr>
<tr><td>14_h</td><td>03_h</td><td rowspan="9">9000_h</td><td rowspan="9">6982_h</td><td>6985_h</td></tr>
<tr><td>12_h</td><td>01_h</td><td rowspan="8">$6b00_h$</td></tr>
<tr><td>12_h</td><td>02_h</td></tr>
<tr><td>13_h</td><td>01_h</td></tr>
<tr><td>13_h</td><td>02_h</td></tr>
<tr><td>13_h</td><td>03_h</td></tr>
<tr><td>15_h</td><td>01_h</td></tr>
<tr><td>15_h</td><td>02_h</td></tr>
</table>

Table 5.8: Key Identifiers Discovered on an ez-link Card.

For the *debit command*, the keys (14_h, 01_h) and (14_h, 02_h) yielded the error code 6985_h, which indicates a violated usage condition. All other keys trigger the error code 6982_h. Since a debit operation can be performed with a debit, autoload or signature

key, the uniform error code of the eight lower keys in the table correspond to the results obtained from the read purse command. The usage condition error of the other two keys again suggests that they are credit keys, since those kind of keys are not allowed to be used for debit operations.

For the *credit command*, the identifiers of the credit keys 1 and 2 were kept identical during the brute-force tests and did not vary independently. This limited the number of tries to 65,536 also in this case. Similar to the eight keys specified in the debit command, keys (14_h, 01_h) and (14_h, 02_h) caused an unsatisfied security state (code 6982_h), which suggests that they are indeed eligible for credit operations. However, unlike in the debit case, the other eight keys do not yield a consistent status code. Only key (14_h, 03_h) caused a usage condition error, the others triggered a wrong parameter P1/P2 error. A reason for that may be the fact that according to the CEPAS standard credit keys must not reside outside issuer key files. If a non-issuer key file is specified in a credit command, an error is returned. Consequently, it is possible that the card first checks whether the given file is indeed an issuer key file (and returns $6b00_h$ if not) and afterwards checks whether the given key is actually a credit key. This would explain the differences to the debit command results under the assumption that file 14_h is an issuer key file while the rest is not.

Card Analysis As shown by Kerschbaum et al. in [8], the EZ-Online application gathers the transaction log, which is referred to as 'Rec_Short' in the paper, and sends it to the back end system. The EZ-Link server then replies with more detailed transaction records, which are labeled as 'Rec_Long' and displayed by the Java application. This conversion by the server is considered to be a fixed part of recovering the transaction history of an ez-link Card by the authors.

The following paragraphs show that the 'Rec_Short' entries gathered directly from the card by the read purse 'Log' command already contain sufficient information to construct the 'Rec_Long' version without the need of the EZ-Link server and thus of an active Internet connection. In addition, also the information collected with the 'Status (w)' variant can be entirely interpreted and displayed in a human readable form. This refinement of the ez-link Card information leak is already used in practice by Eric Butler and his contributors in their publicly available and open source Android application FareBot (v2.50) [35, 36].

To illustrate how the information leak can be exploited, an own Android application has been developed that uses the discussed read purse commands to retrieve and display the status information and transaction log of an EZ-Link electronic purse. It is important to note that the application shown here has been developed solely by the author prior to the knowledge of the FareBot open source code base. Consequently, neither code nor ideas were taken from it. The application development, the interpretation of the data returned by the card as well as the discussion in the following paragraphs are only based on the CEPAS standard and own experiments. The Android application presented here is hosted on a Google Nexus S smartphone running Android v4.1.2. In addition to the features of FareBot v2.50, it includes a detailed explanation of the purse status and the debit options, it always shows the last credit transaction in a human readable form and lists the cryptogram, CRC as well as the entire response content in their hexadecimal representation.

Table 5.9 shows the response of an ez-link Card to the 'Status (w)' read purse variant (Kf = 14_h, Kn = 03_h). All bytes belonging together for interpretation reasons are already separated in different rows and are listed in order of appearance, starting with the CEPAS

version and ending with the CRC. In order to illustrate how to derive information from the raw bytes, the three upper pictures in Figure 5.5 show the same response but processed by the Android application.

Purse Information (hex)	
CEPAS Version	`02`
Purse Status	`01`
Purse Balance	`000471`
Autoload Amount	`000000`
CAN	`1000120010164466`
CSN	`b20d52715b031d3f`
Expiry Date	`21f4`
Creation Date	`1964`
Last Credit Trans. TRP	`001311ff`
Last Credit Trans. Header	`7500138823016d3c`
Transaction Log Size	`1e`
Issuer Data Length	`20`
Last Trans. TRP	`00170100`
Last Trans. Record	`30ffff762331433a4842432d424e4320`
Issuer Data	`02000000010000020200000000000000` `00000000000000000000000000000000`
Last Trans. Debit Options	`00`
Last Trans. Signed Cert.	`4e23121657abbc96`
Counter Data	`f5d4fdb726b0b616`
CRC	`c109`

Table 5.9: Read Purse 'Status (w)' Command Response [93].

The first byte of the response, $\texttt{02}_h$, encodes the CEPAS version the card implements and can be directly converted to a decimal number, indicating support for version v2.0 of the standard. The second byte, $\texttt{01}_h$, contains purse status information and indicates that the purse is enabled and the autoload feature is currently deactivated, as shown in Figure 5.5. The next three bytes code the purse balance and can be directly converted to a decimal number, yielding $\texttt{000471}_h = 1137_d$. This is actually the number of cents, thus the balance is S$ 11.37. The following three bytes represent the autoload amount, which would be automatically added to the purse balance in case of an autoload event. Since this feature is currently not in use, the amount is zero. The two following groups of eight bytes are the main identification numbers of the ez-link Card. The first one is the CAN, which although given in hexadecimal representation can be read decimally: 1000120010164466_d. It can also be found in Figure 5.1b on the back side of the ez-link Card. The second ID is the CSN, which is simply an 8-byte random number. It is included in the historical bytes of the ATR shown in Table 5.2. The two following groups of bytes encode the purse creation and expiry date. Similar to the concept of the Unix time, the numbers count the time from a defined reference date. Experiments showed that the reference point in time for EZ-Link is 'Sun, 01 Jan 1995 00:00:00'. The values $\texttt{21f4}_h = 8692_d$ and $\texttt{1964}_h = 6500_d$ represent the number of days that passed since then. Because this reference date is not defined in the standard, it may vary for other CEPAS implementations. [93]

Figure 5.5: EZ-Link Purse Display with Custom Android Application (Inverted Color).

Transaction Log Entries (hex)			
Type	Amount	DateTime	User Data
30	ffffd9	2330d9db	4842462d46525020
31	ffffad	232d680b	5356432031373900
76	00000a	232d68b7	5356432031373900

Table 5.10: Read Purse 'Log' Command Responses.

Transaction Types	
Type	Description
30_h	MRT Debit
31_h	Bus Debit
75_h	Credit
76_h	Bus Refund
$A0_h$	Debit General

Table 5.11: Trans. Types.

Since a comparison to other implementations and further processing in software is much easier, the original time stamps given by the ez-link Card are converted into Unix time stamps, which simply count the number of seconds since 'Thu, 01 Jan 1970 00:00:00' following Coordinated Universal Time (UTC). Consequently the EZ-Link reference date is given as the Unix time stamp 788918400 and has to be added to all date conversions as an offset. Equation 5.1 shows how to derive a Unix time stamp for the purse creation and expiry date. The factor 86400 represents the number of seconds per day. [93]

$$\text{Days} \cdot \text{SecondsPerDay} + \text{Offset} = \text{Unix Timestamp} \tag{5.1}$$

$$6500 \cdot 86400 + 788918400 = 1350518400 \mathrel{\widehat{=}} \text{Thu, 18 Oct 2012 00:00:00} \tag{5.2}$$

$$8692 \cdot 86400 + 788918400 = 1539907200 \mathrel{\widehat{=}} \text{Fri, 19 Oct 2018 00:00:00} \tag{5.3}$$

Surprisingly the given validity time span of the purse is six years, which contradicts the five year period that is mentioned on the website [6]. The next two groups of bytes in the response are related to the last credit transaction performed with the card. The four bytes $\texttt{001311ff}_h$ are the Terminal Reference Parameter, the meaning of which is determined by the implementation. The following 8-byte group is the transaction header, which is a transaction record as shown in Table 5.3 but without the user data. Consequently the type $\texttt{75}_h$ indicates a credit transaction. $\texttt{001388}_h$ encodes the amount that is added to the balance and can be converted in the same way as the purse balance, which yields S$ 50.00. The value $\texttt{23016d3c}_h = 587296060_d$ represents the date and time of the transaction and counts the seconds from the previously stated EZ-Link reference date. Thus it can be similarly converted to a Unix time stamp using Equation 5.4. [93]

$$\text{Seconds} + \text{Offset} = \text{Unix Timestamp} \tag{5.4}$$

$$587296060 + 788918400 = 1350518400 \mathrel{\widehat{=}} \text{Sun, 11 Aug 2013 09:47:40} \tag{5.5}$$

The next byte $\texttt{1e}_h = 30_d$ indicates how many transactions are saved in the transaction log. After that byte $\texttt{20}_h = 32_d$ gives the length of the issuer specific data, which is surrounded by information about the last transaction. There is again the TRP, given as $\texttt{00170100}_h$, and the transaction record, but this time including the user data. As experiments have shown, the byte $\texttt{30}_h$ indicates a debit operation for an MRT trip. The amount $\texttt{ffff76}_h = -138$ is interpreted as a two's-complement representation and is thus negative. It again gives the amount in cents, which yields a balance decrease of S$ -1.38. The time of the transaction is given as 'Mon, 16 Sep 2013 16:37:14'. The user data of transactions has been observed to be typically in ASCII encoding, which yields $\texttt{4842432d424e4320}_h$ = 'HBC-BNC'. After comparing the actual trip to the data given in the card response, the string 'HBC' indicates the Harbour Front whereas 'BNC' refers to the Boon Lay MRT station (see 3.1). Following this transaction record is the issuer specific data block with 32 bytes, the meaning of which is not specified in the standard. The single byte $\texttt{00}_h$ after that encodes the debit options of the last transaction. As shown in Figure 5.5, the lower nibble specifies the modus operandi of the debit operation. Besides the normal deduction of the purse ($\texttt{0}_h$) there is for instance also a deduction with a partial refund ($\texttt{1}_h$). The two most significant bits can be used to disable the expiry date checks as well as the autoload feature. [93]

The 'Status (w/o)' response would normally end here. For the 'Status (w)' case, however, there are ten additional bytes appended to the response. The first eight bytes,

$\texttt{4e23121657abbc96}_h$, are the encrypted signed certificate of the last transaction followed by the also encrypted eight byte long counter data $\texttt{f5d4fdb726b0b616}_h$. The last two bytes, $\texttt{c109}_h$, are an ISO/IEC 14443-3 Type B CRC calculated over the unencrypted signed certificate and counter data. The two ciphertexts differ between protocol runs, since session keys are used for encryption. In contrast, the CRC stays constant, since it is calculated over unencrypted data. The entire content of the response in hexadecimal representation is given in the upper right-most picture in Figure 5.5 under the title 'Raw Content'. [93]

Table 5.10 shows three out of the 30 transaction records that can be retrieved with the 'Log' variant of the read purse command. The raw bytes of each entry are shown in the table, whereas the processed results are again illustrated in Figure 5.5. The first transaction corresponds to lower left-most, the second to the lower middle and the third to the lower right-most picture. The meaning of the type and user data fields was derived by manually noting details of transactions and comparing the notes with the entries presented here. Table 5.11 lists all transaction types whose meaning could be revealed. The amount and date can be converted in the same way as applied to the purse status response. In all analyzed transactions the user data field always contained an ASCII-encoded string, which makes it easier to interpret. The user data field of the first entry (MRT Debit) 'HBF-FRP' contains the start and end station of the trip, which indicates a journey from Harbour Front or 'HBF' to Farrer Park or 'FRP' MRT station. The user data fields of the second (Bus Debit) and third (Bus Refund) entry contain the bus number. The string 'SVC 179' indicates the bus service number 179, which starts its loop at Boon Lay Interchange and heads for Nanyang Technological University campus. The transaction type 'Credit' appears in the last credit transaction header shown in Table 5.9 and denotes an increase of the purse balance. Since ez-link Cards can also be used to pay for other services than public transport, there is also the category 'Debit General', which has been observed for example when paying in convenience stores. [93]

5.4 Privacy Evaluation

This section revolves around the read purse command and discusses a potential information leak caused by it. As shown in the previous section, all that is needed to access the transaction history of an ez-link Card is an inexpensive RFID reader, which eliminates the need of an Internet connection to the EZ-Link server and makes an attack easier to execute in practice. To illustrate this and to assess the impact on the privacy of customers, the next section discusses results of a real world experiment that puts the attack to the test. After that countermeasures as well as general privacy enhancements of public transportation systems based on RFID technology are reviewed.

5.4.1 CEPAS e-Purse

In the previous section the CEPAS standard and its three main commands have been analyzed. In order to successfully perform a credit or debit operation, the key set on the card must be known to derive the correct session keys. Without them, the terminal cannot send a valid request, thus the card will answer with an error status instead of performing the desired action. This authentication mechanism ensures that only authorized parties can in- and decrease the purse balance.

In contrast to that there is the read purse command. It can be executed by the terminal without proving to the card that it is authorized to do so. In fact, anyone can successfully retrieve all the information provided through the read purse commands, since there is no access restriction specified in the standard. This is exactly what the Android application running on a smartphone with an NFC interface does and what any other inexpensive RFID reader like the ACR122U can do as well. It reads the purse and transaction information and displays it in a human readable way without the need of further resources like an Internet connection.

Similar to the privacy discussion of the Standard Ticket, the fact that there is no access restriction to prevent unauthorized parties to read ticket information and that this can be done over a wireless channel without attracting the owner's attention raises concerns over misuse of the gathered data. But unlike the Standard Ticket, where this data is limited to card identifiers and two scrambled data packets, information retrieved from an ez-link Card is comprehensive, easy to interpret and personal. This clearly constitutes a privacy violation as defined in the beginning of this thesis.

An attacker can access purse information for instance with a skimming attack that has already been discussed for the Standard Ticket. In crowded areas, which are typically found in public transport especially during rush hour, pointing a custom made RFID reader, e.g. hidden inside a briefcase, towards another commuter's wallet is a realistic scenario. In addition, every terminal that performs debit or credit transactions with the ez-link Card can also read the transaction history without leaving any traces. The consequences of such an information leak can be diverse and largely depend on the intention and creativity of an attacker. For example, the easy access to the current purse balance of an ez-link Card as well as to its expiry and activation status enables a very quick assessment of whether to steal it or not. The biggest portion of the data, however, is made up by all the saved transactions. Together with two unique IDs (CAN, CSN) this is the optimal basis for creating a tracking profile of the card holder.

Note The following sections contain comprehensive transaction data from an actual ez-link Card, which has been used by the author only. Therefore, the privacy of nobody else is impaired through the publication of this data. In the case of the author, it is his belief that the illustrations that are based on the data and the points made in the discussions accompanying them are vital for an open discussion about the privacy issues and thus greatly outweigh the privacy impairment that is implied by them. Furthermore, the techniques described here solely serve an illustrative purpose and shall demonstrate what can potentially be derived from such a data set. Since the data set stems from the author's card, it is limited in its size. In addition, the complete tracking profile is known beforehand. Consequently, the intention is not to assess the effectiveness of the applied techniques.

Analysis of travel behavior and activity identification on a much bigger set of public transport travel data have been done by Chakirov et al. in [94, 95]. For their first paper listed, they had access to one full day of travel records stemming from the EZ-Link system in September 2010. The already preprocessed set contained approximately 3.6 million journeys (equals one or more single, coherent trips) and 1.8 million unique card IDs. For their second work, they had access to data registered during one full week in April 2011 again stemming from the EZ-Link system. They also refer to the Household Interview Travel Survey (HITS) 2008, which is a survey conducted by the LTA every couple of

years to assess Singaporeans' traveling patterns. The comparison of HITS 2012 to 2008 is briefly summarized in a press release in [96] and in the Land Transport Master Plan 2013 published by the LTA [97]. All of the mentioned work can aid with interpreting travel data collected from an ez-link Card. Consequently, it will be referred to in the following paragraphs.

Short-Term Profiling In order to illustrate possible privacy impairments caused by the discovered information leak, one transaction log of the author's ez-link Card containing 30 entries is analyzed here. The time span covered by the log is approximately five days. Note that the information used here was retrieved by reading the ez-link Card only once, which is a matter of mere seconds and feasible to do in practice by an attacker without the consent of the card holder. Table A.1 in Appendix A shows the list of transactions, which are already converted into a human readable format.

To put the list into a more practical context, an attacker could have collected it in the morning of Thursday, April 11^{th} 2013. This is a realistic scenario, since crowded trains and buses during rush hour by default cause small distances between commuters, which is one way to covertly read an ez-link Card without the permission of the card holder. Since the transaction log is limited to 30 entries, the last day included is Saturday, April 6^{th} 2013.

The goal now is to reconstruct as much information as possible from the collected transactions. For that purpose, trips with the MRT are especially beneficial, since the start and end station of the trip are given in the user data field of the transaction entry. The only effort is to map the given abbreviations to real MRT stations. This can be done most reliably by experiments, but consulting web sites like [98] can also be a good starting point. The MRT trips form a solid frame, to which more details can be added. Transactions from bus trips for instance solely contain the bus service number in the user data field. Only with the paid fare or traveling time and an additional source of information it is possible to derive the approximate start and stop of the bus ride. Such an additional source can be a fare table like the one hosted by the Public Transport Council [99], which lists all fares depending on the distance traveled. A time table, which contains information about the traveling times between two stops, can also provide valuable input, although varying traffic has to be considered.

Figure 5.6 shows the results for Saturday after the reconstruction process has been applied to the eight transactions done on that day. Important locations are always marked by circled numbers, which are given in brackets in the text, e.g. (5). The chronologically first log entry is an MRT trip from Clementi (1) or 'CLE' to Choa Chu Kang (2) or 'CCK'. The next two entries stem from the bus service number 927, which starts its loop at Choa Chu Kang Interchange to head for the Singapore Zoo. Following the approach in [95], single trips with a time span of less than an hour between them are considered to belong to one journey. The actual fare of the bus ride is S$ 1.27, if the partial refund of S$ 0.39 is also taken into account. The fare table reveals that there is only one stop that would yield a fare of S$ 1.27 when starting from Choa Chu Kang Interchange and that is 'Singapore Zoo' (3). The estimated travel time given in the fare table between the two stops is 23 minutes, which fits the actual time of 21 minutes and 40 seconds (between the debit and partial refund transaction) very well. Given the fact that there is roughly a five hour gap to the next transaction in the list, a visit to one of the establishments of the zoo is very likely. [100]

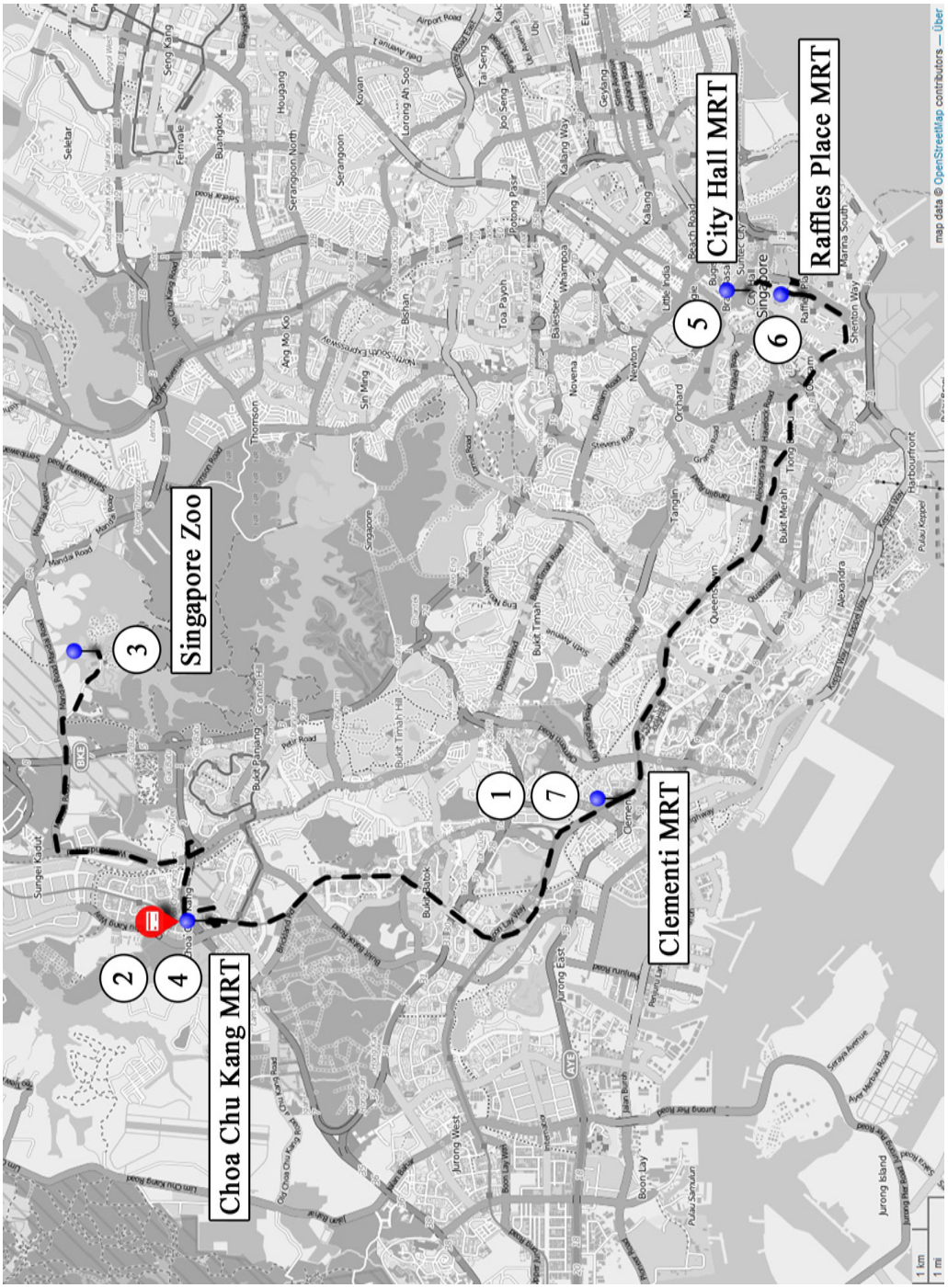


Figure 5.6: Short-Term Profiling: Reconstruction of Saturday, April 6^{th} 2013.

The next two transactions are again stemming from bus service 927, but this time the overall fare is S$ 1.23. Reading a bit ahead in the transaction list reveals that the upcoming entries are again located at Choa Chu Kang, thus the assumption is that the commuter traveled back to the interchange with line 927. Again after querying the fare table, the only stop that would cause a fare of S$ 1.23 to the interchange is 'After Singapore Zoo', which is still in proximity to the zoo. The given estimated travel time of 19 minutes again fits the actual time of 18 minutes and 32 seconds very well. Back at Choa Chu Kang station (4), the commuter recharged the ez-link Card with S$ 50.00 at a general ticket machine. This is indicated by the user data string 'GTM' and illustrated by a small credit card sign in the figure. Throughout the analysis there have been multiple cases where one station is referred to by more than one abbreviation, which is also the case here. 'CCB' as well as 'CCK' refer to Choa Chu Kang station, although probably to different parts of it. 'CCB' possibly indicates the bus interchange whereas 'CCK' is the label for the MRT station. [100]

The next transaction still belonging to the journey that started at the Singapore Zoo is an MRT trip, from Choa Chu Kang to City Hall (5) or 'CTH'. The time gap of roughly five and a half hours to the last transaction on that day suggests some evening activities in the downtown area of Singapore. The last journey, an MRT trip from Raffles Place (6) or 'RFP' back to Clementi (7), where the first trip started in the morning, might be an indicator that the commuter lives in the Clementi area. This is also supported by the fact that Monday to Thursday the first and last transactions are always MRT trips from and to Clementi MRT station. Additionally, according to the HITS 2008 survey results, the majority of home activities start between 16:00 and 23:00 and have a duration that ranges between 9 and 14 hours with a distinct peak between 12 and 13 hours [95]. Given the gaps of approximately 8.5h (Sat-Mon), 14h (Mon-Tue), 13h (Tue-Wed) and 8h (Wed-Thu), they fit this model well enough.

Clementi MRT station (1) is also shown in Figure 5.7, which illustrates transactions on work days. Since there is no entry on Sunday, the figure depicts the rest of the transactions in the list. All the first trips in the morning head for Boon Lay MRT (2) or 'BNL'. After that there is always a trip with bus service 179 for a fare of S$ 0.30. Starting from Boon Lay Interchange service 179 heads for Nanyang Technological University campus. Unfortunately the very low fare of S$ 0.30 does not correspond to the fares of line 179 as given in the fare and time table. A possible reason might be that there was a discount given. Taking the travel times on Monday (21 min, 51 sec) and Wednesday (22 min, 58 sec), the bus stop that fits this time best is 'Lee Wee Nam Library' on NTU campus. In contrast, the travel times on Tuesday (17 min, 07 sec) and Thursday (17 min, 30 sec) suggest stops 'Hall 1' or 'Opposite Hall 6'. Regardless of the times not matching, all exit stops are located on NTU premises (3). [102]

The time gap of approximately 9.5 hours between the bus rides in the morning and the next transactions, which are in the evening, suggests that the commuter works in that period of time. This is supported by the results of the HITS conducted in 2008. According to the survey, the majority of work activities start between 06:00 and 11:00 o'clock and have a duration between 7 and 11 hours with a distribution peak between 8.5 and 9 hours. [95]

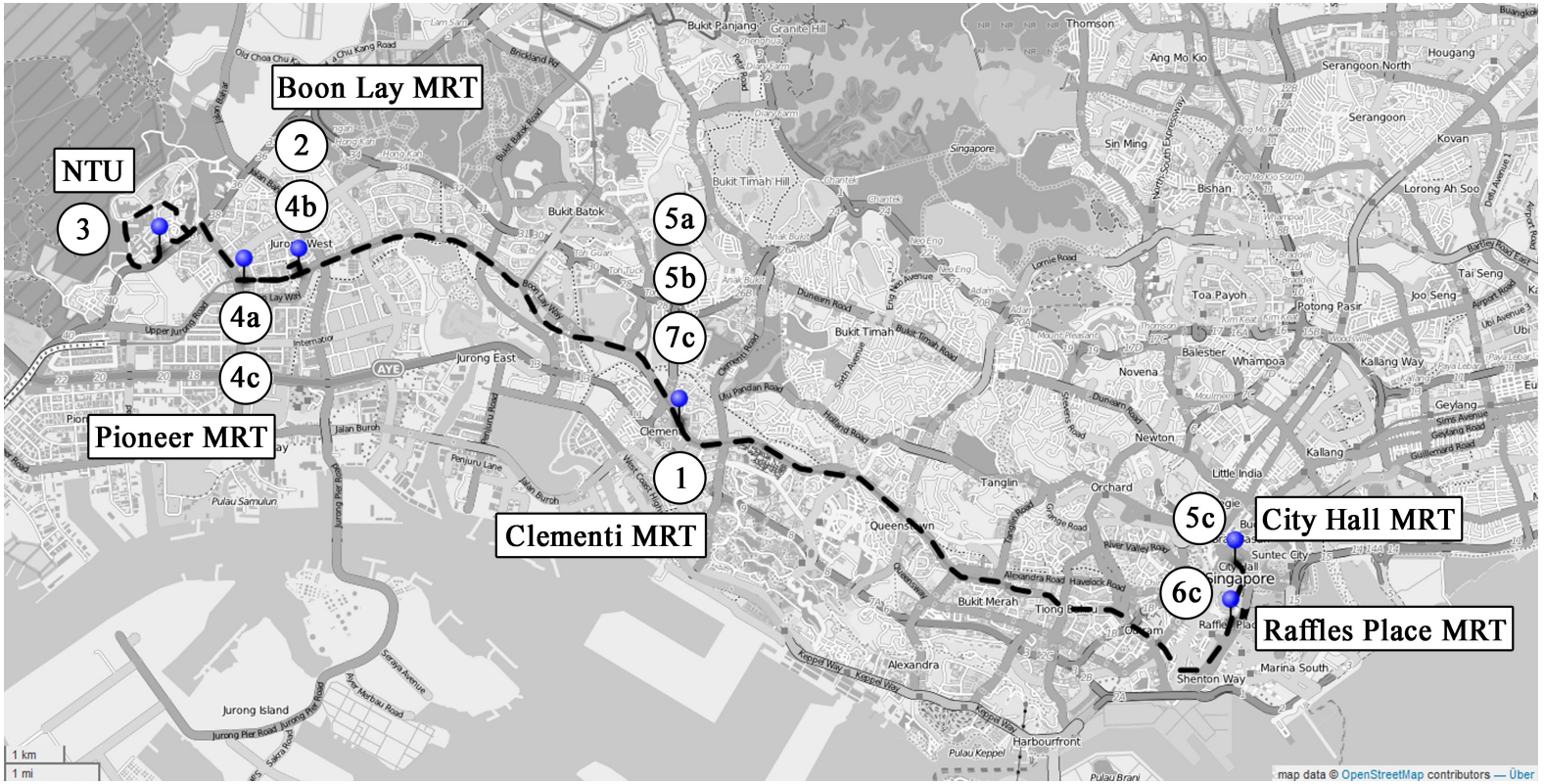


Figure 5.7: Short-Term Profiling: Reconstruction of Workdays.

Another indicator for the work place really being on NTU campus (3) are the first entries after the time gap. They are again bus rides with line 179, but this time there are differences between the week days. On Monday the fare is S$ 0.83, on Tuesday and Wednesday it is S$ 0.73, which is linked to the next transactions after the bus ride. On Monday the next entry is an MRT trip starting at Boon Lay (4b), to which the only bus stop that yields a S$ 0.83 fare is 'Innovation Center'. Since there is no partial refund given, the travel time cannot be assessed adequately. On Tuesday and Wednesday the next entries are MRT trips starting at Pioneer MRT (4a, 4c) or 'PNR', which is located near the bus stop 'Blk 649A'. Taking the fare of S$ 0.73 and the travel times (8 min, 41 sec and 10 min, 15 sec), the best match for the entry stop is 'Hall 4'. Although the results of the entry bus stop are not precise, all possibilities are again located on NTU campus. [102]

Now it is time to recap and finish the reconstruction of the week days. The tour on Monday started in Clementi (1), continued to Boon Lay (2), to NTU campus (3), back to Boon Lay (4b) and finally to Clementi (5b). With the previous conclusions that the commuter's home is in the Clementi area and the work place is at NTU, this fits very well. The tour on Tuesday is very similar and differs only in the way back, which goes over Pioneer (4a) back to Clementi (5a). The trips after work on Wednesday are a bit different, although they also involve Pioneer (4c). From there an MRT train was taken to City Hall (5c) or 'CTH' and right after that to Raffles Place (6c) or 'RFP'. The gap of roughly five hours to the next trip again indicates some evening activities in downtown Singapore. An MRT trip from Raffles Place (6c) back to Clementi (7c) late at night completes the tour.

The previous paragraphs showed that a comprehensive reconstruction of a commuters travel history is possible and that activities can be inferred from it during the process. Most importantly, the locations of the work place and home can potentially be inferred, which is considered to be sensitive personal information. However, the main requirement is a consistent and coherent transaction record. This implies regular usage of the public transportation system, since trips not payed for with an ez-link Card can hardly be detected. For instance walking or using a private car renders the transaction log incoherent, thus activities are hard to infer from it.

With 7.097 million daily passenger trips on average and a population of 5.312 million in 2012, Singapore's public transportation system is undoubtedly frequently used [1]. In [94] Chakirov et al. also discuss journey consistency. When considering commuters with more than one journey record per day, 89.7 % of journeys that follow a preceding one, start less than 1 km away from the location the commuter previously alighted (In this context one journey equals to one or more consecutive trips). This clearly indicates that the majority of commuters more frequently using public transport have consistent journey records. When considering the distance between the first and last station of the day, which is critical for inferring the home location, the last station lies in a 1 km radius of the first station in 61 % of the cases. Again, the majority of commuters shows consistent travel records also in this regard.

All in all, this illustrates that a short-term travel profile can be constructed from the data stored on an ez-link Card and that sensitive personal information can be inferred from it. Naturally, the profiling gets more accurate, the more data is available, since outliers for instance can be identified better.

Long-Term Profiling The previous paragraphs illustrate that it is possible to reconstruct a movement profile of a commuter and derive additional information from it by only taking one transaction log of 30 entries into account. Because every ez-link Card also includes its two unique IDs (CAN, CSN) into the read purse response, tracking a card over multiple readings and also in a set of readings from different cards is easily possible. This is the basis that enables long-term profiling of a commuter.

To illustrate this problem, an experiment with the ez-link Card of the author was conducted over a period of approximately six months. The goal was to record all transactions done with the card over this period of time in order to get a comprehensive data set and a solid basis for a long-term usage profile. As demonstrated in the previous section, in the author's case one transaction log covers a time span of less than a week. That's why the log was read two times per week on average over the entire period of six months.

In total 562 transactions could be collected over a period of 188 days, from March 13^{th} 2013 to September 16^{th} 2013. Out of these days there were 120 with and 68 without transactions. The average number of transactions per day is therefore 2.99 considering all 188 days or 4.68 if only those 120 days are taken into account. Figure 5.8 shows the distribution of the transactions over the given period of time in a histogram. The distribution of the transactions per day shows distinct patterns over time, which can be attributed to different phenomena.

One of the more apparent observations are two gaps in the histogram, one from May 25^{th} to June 3^{rd} 2013 and one from August 12^{th} to 23^{rd} 2013. Since the rest of the histogram shows a very regular usage of the ez-link Card with maximum gaps of four days, the time spans of 10 and 12 days clearly mark prominent events. In this experiment these periods have no transactions, because the time was spent abroad. Another observation that accompanies these gaps but is not necessarily linked to them are the days before they occur. While the average number of transactions per day is below five, there were 20 transactions registered on August 4^{th} and 17 on May 24^{th}. The reason for that is sightseeing, which causes a higher number of irregular trips aggregated in a short period of time and spread over large parts of Singapore that typically exhibit points of interest.

This observation might be used to distinguish a tourist from a residential commuter, which tends to show a more regularly distributed transaction history with less trips per day that are more localized and aggregated around frequently traveled spots, for instance home and work place. The results from a short-term profiling can certainly help distinguishing residents and tourists and might even be sufficient due to the fundamental differences in travel behavior.

Additionally, the number of transactions per day reveal another interesting aspect. When comparing this metric for the months March and April to the rest of the time span, there is a significant difference. The average number of transactions per day for March to April is 4.53 taking days with no transactions into account and 5.05 taking only days that show at least one transaction. For the rest of the time span (large gaps in May and August excluded) these numbers are 2.91 and 4.47 respectively. Clearly, for days having one or more transactions the average numbers per day, 4.47 and 5.05, do not differ significantly, however when considering days with no data as well, the difference between 2.91 and 4.53 is considerable. This suggests a more even distribution in March and April compared to the other months and can also be seen in the histogram. From May onwards, the distribution looks more fragmented than in the beginning and there is a specific reason for that.

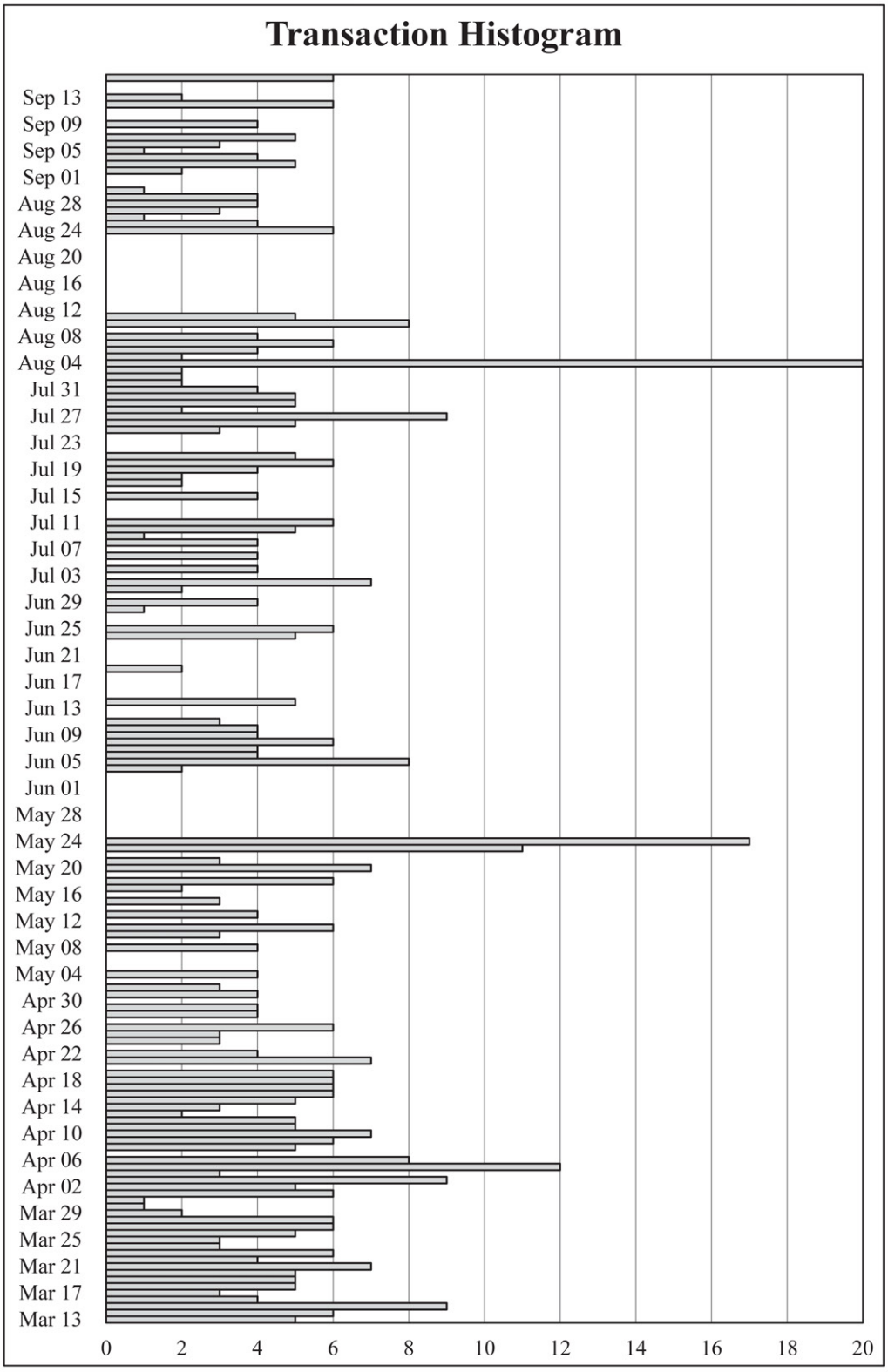


Figure 5.8: Long-Term Profiling: Transaction Histogram.

In March and April, MRT and bus were the primary means to travel from home to work. From May onwards, the location of the accommodation changed and a shuttle bus was taken to work that could not be paid with an ez-link Card. Thus, it is not showing up in the transactions. While in the first two months, MRT and bus trips provided a base level of transactions, the other months are fragmented with days where no travel payments were necessary, therefore containing no entries at all. The list of transactions in Table A.2 in Appendix A shows an excerpt of the transaction log during the month of May. In contrast to the list from April, information is scarce. There are no transactions in the morning, only in the evening. On Monday, bus 179 is taken from NTU campus to Pioneer station and from there the MRT is taken to Bukit Batok or 'BBT'. However, from there a private shuttle bus is taken, which is indicated by the user data string 'CFDB' referring to the transport company ComfortDelGro Corporation.

Since there is hardly any information available on the exact routes of these private shuttle buses and since they can cover extended areas with a fixed fare regardless of the distance traveled, it is significantly harder to reconstruct a travel profile let alone infer the approximate area of accommodation. The missing entries for Tuesday and Thursday reflect what has been discussed previously. Wednesday and Friday bring no new information to the table. The same situation arises when a commuter uses multiple ez-link Cards or other means of transportation, like a private car. All of that contributes to a fragmentation of the histogram, which makes profiling significantly harder.

All in all, a long-term profile enables an attacker to infer more details about the commuter, since different usage patterns can be observed even with a simple analysis tool like a histogram. Regularly using the MRT and bus system accounts for an even distribution of transactions, whereas usage of a private car or shuttle bus may fragment the distribution. Also, sightseeing potentially causes a different pattern compared to an average resident. To infer as much as possible from the data set, a long- and short-term profiling approach should be combined.

5.4.2 Countermeasures

The previous section showed that the unrestricted access to comprehensive information about the electronic purse stored on an ez-link Card and about transactions performed with it impairs the privacy of the card holder. Consequently, a first step must be to restore this access restriction and seal the leak, which is discussed in the first part of the section. After that, more general measures to ensure privacy in public transport systems by design are reviewed.

Immediate Action As a response to the issue, users of the EZ-Link system can take some measures to significantly reduce the impact of the information leak. Since access restriction is the key aspect, consequently ez-link Cards must be handled with greater care. As already shown, their contactless nature makes them susceptible to attacks from longer distances than defined in the standards and often promoted by manufacturers and operators. To counteract this issue, there are a big variety of RFID blocking cases, badge holders and wallets on the market. However, experience showed that some of the products do not work as expected and that customers usually do not have the equipment or knowledge to verify this prior to purchase. For technically inexperienced customers, asking for a demonstration or following recommendations by third parties with good reputation are possible solutions. More experienced customers can perform a quick self-assessment

of a product, for which smartphones with an NFC interface can be of great help. As a last resort, wrapping the card in tinfoil and trying to use it afterwards can also lead to a working solution.

However, shielding an ez-link Card from radio-frequency signals significantly impairs the efficiency and convenience introduced with RFID systems, since the card has to be removed manually from the RFID blocking equipment for every transaction. This effort and prolonged waiting times caused by it reduce the usability of the system. In addition, this approach only counteracts covert tries to read an ez-link Card without consent and does not protect from social engineering efforts or malicious point-of-sales operators. In this regard, it is important to be aware that anyone with access to the card (even for a second) can read purse information and transaction data. Adopting a critical attitude towards people actively asking for a card is also a protective measure.

To specifically counteract the analysis of transaction data and the inference of additional information from it, a good measure is to break transaction consistency. As shown in the previous section, a key requirement for travel analysis are coherent transactions forming journeys that are easy to reconstruct. As Table A.2 shows, a fragmented list of transactions is significantly harder to process. Because of that, another countermeasure is to randomly use multiple ez-link Cards, since consecutive transactions get spread over different cards, which adds another obstacle in analyzing the card data. However, note that this does not protect the card from being accessed and that information like the purse balance or last top-up amount is still accessible.

Privacy By Design In [103] Sadeghi et al. discuss ways to ensure user privacy in transport systems that are relying on RFID tickets. To achieve this goal, the authors give a set of necessary requirements, of which the basic ones state that the following actions shall only be done by authorized parties.

- Access to user-related data.
- Identification of tokens.
- Tracing of tokens.

Considering the CEPAS standard and the analysis of an EZ-Link purse implementation shown in the previous sections, none of the above requirements are fulfilled. User-related data can be read by unauthorized parties using the read purse command set. This enables unrestricted access to purse information and past transactions performed with it, which have shown to be clearly related to the card holder. Although the platform-dependent PUPI of an ez-link Card is generated randomly, the unique 8-byte CSN is included in the Answer-To-Reset, which can be easily retrieved. In addition, it is possible to access the 8-byte CAN and CSN with a read purse command. Note that both of these unique identifiers are platform-independent, since they are part of the CEPAS standard, and allow unauthorized identification and thus tracing of an ez-link Card. This clearly violates the second and third requirement.

The biggest problem is that most of the issues stem from the CEPAS standard and its implementation. A first protective measure would be to add an authentication step to the read purse command, in which the reader has to prove that it is authorized to access purse and transaction information. On the one hand this is feasible to implement, since such an authentication mechanism is already part of the other CEPAS commands. On

the other hand, without also changing the CEPAS standard the implementation would immediately loose its CEPAS compliance. Since changing a standard is considered a time-consuming and expensive process, adapting the implementation is more realistic as a short-term solution. A compromise would be to add the authentication step to all applications making use of the read purse command, but still let the card return dummy data for every unauthenticated but CEPAS compliant read purse command. This way, real purse information can only be retrieved in an authorized way. However, this still requires to update the software on all issued ez-link Cards and still only remedies the unrestricted access to user-related data. Identification and tracing are still possible, since the basic authentication principle used in the CEPAS standard is based on the fact that the reader can uniquely identify a card before the authentication process. This is necessary in order to derive the card specific key set. Consequently, a satisfactory long-term solution must incorporate more radical changes to the overall system.

In [103] Sadeghi et al. present a privacy-preserving e-ticket system based on symmetric-key authentication, secure key storage with physically unclonable functions (PUF) and a rerandomizable encryption scheme that tackles the problem of token identification and traceability. The constraints of RFID tags play a vital role in this process, since ticket cost directly relates to computing capabilities and consequently to the overall system design. One very important example of this trade-off is the choice between symmetric and asymmetric ciphers. Symmetric-key cryptography tends to run fast and to be cheap to implement, but also limits the realization of sophisticated privacy and security techniques. Asymmetric or public-key cryptography on the other hand tends to be computationally intense and costly to put on an RFID tag, but enables feature-rich mechanisms.

One example that makes use of asymmetric cryptography to design a privacy-preserving ticketing system is given by Kerschbaum et al. in [8]. The authors introduce anonymous card authentication with a zero-knowledge proof and even privacy-preserving billing and data mining by the operator with partially homomorphic encryption techniques. However, to realize this feature-rich system design, challenges regarding implementation and efficiency have to be overcome.

The balance of user privacy and operator interests, as addressed in [8], is an important issue in systems that aggregate a considerable amount of user data, as it is the case in public transport. Monitoring of personal behavior, marketing and advertising as well as improper protection of the data collection and disclosure of data to third parties are just some of the concerns that arise. However, tasks like schedule and fare adjustments as well as network extensions and load analysis are reasonable examples that also rely on a comprehensive travel data set. In [104] for instance, Lee et al. present a bus service reliability analysis using smart card data stemming from the EZ-Link system. A general overview of the usage of smart card data in public transit is given by Pelletier et al. in [105].

5.5 Security Evaluation

This section discusses security aspects of the CEPAS standard. Based on the observations, a side-channel analysis scenario is developed targeting the cryptographic keys used during the purse operations. The proof-of-concept first-order DEMA attacks are then performed on measurements of an ez-link Card's electromagnetic emanation.

5.5.1 CEPAS e-Purse

The CEPAS standard relies on symmetric cryptography to secure purse operations. Mutual authentication between the terminal and the card is based on the fact that both parties have the same set of keys and are therefore able to derive the correct session keys used in the protocol run. With a simple exchange of random numbers, these temporary keys are derived from the static key set. This way, all encrypted data that is transmitted over the insecure RFID channel is created with temporary keys that are only valid for one exchange of command and response. The result of an en- or decryption with a static key is never transmitted in plain and can therefore not be observed by an attacker. This is a protective measure for the static keys stored on the card.

How a terminal shall obtain the static card keys is not specified in the CEPAS standard. A common approach is to use a key derivation algorithm that takes a master key and card specific information (like a unique ID) and creates a card-specific set of keys. This step is recommended in the CEPAS standard, but not explicitly defined. According to the recommendation, the CAN and CSN can be used as card specific information and the master key should be stored in a secure access module (SAM). The advantage of such an approach is that a terminal can easily obtain all necessary keys without maintaining a large database. In addition, a properly designed key derivation algorithm ensures that every card gets a different key set and that a compromised card does neither affect other cards nor reveal the master key. [93]

The different key types that are associated with specific purse operations allow a high level of flexibility regarding key management. Switching between keys is easy, since they are selected explicitly in every single command. A simple change of the command parameters is sufficient to select a new key. In addition, it is possible to enforce different policies for different keys. For example, credit keys must only be stored in issuer key files, whereas all other keys can reside in any key file. Although the CEPAS standard consistently follows this concept, it also allows the reduction of the entire key set to only one key. It is up to the implementation whether the pool of different keys to choose from is taken advantage of or not. As the experiments in Section 5.3.2 showed, there are ten keys available to the EZ-Link purse, two of which are assumed to be credit keys. Although this does not suggest a key set reduction, it could not be fully clarified whether the system actually makes use of the entire key set, since access to official terminals is usually very limited. In order avoid this limitation, the security evaluation was continued with side-channel analysis, which only requires the card. [93]

5.5.2 Side-Channel Analysis

Performing a differential side-channel analysis on a target essentially poses the requirement of a known algorithm, access to a varying input or output, a static secret that should be recovered and a side-channel that is measured. In order to attack the static keys stored on an ez-link Card, an analysis scenario that fulfills all of these requirements has to be defined first.

Scenario When issuing a read purse ‘Status (w)’ command, the card creates a reduced version of the receipt cryptogram, which consists of the encrypted signed certificate of the last transaction and the encrypted content of the three internal counters. The creation of this reduced cryptogram is depicted in Figure 5.9. Prior to the read purse ‘Status (w)’ command, the terminal has to issue a Get Challenge in order to collect an 8-byte random

number from the card. Embedded in the command, the terminal specifies a static key used to derive the session key from as well as its own 8-byte random number. In this case, the fixed key can be a debit, autoload or signature key. Since 3DES operates on blocks of eight bytes, the two random numbers have to be processed consecutively. The card starts with the terminal random number and encrypts its own one as the second block. Because the initial vector is zero, the first cryptographic operation performed by the card is a 3DES encryption of a plaintext chosen by the terminal. The output of this encryption is XOR'ed with the card random number and again 3DES-encrypted with the same key. The concatenated outputs are then the current session key, which is illustrated in the figure.

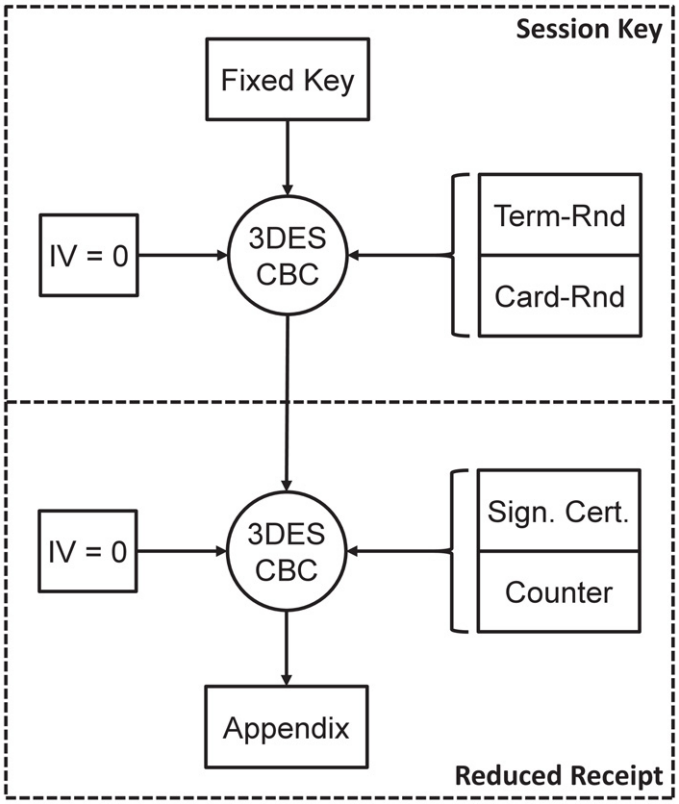


Figure 5.9: Creation of the Reduced Transaction Receipt Cryptogram (Appendix) During the CEPAS Read Purse Command [93].

This is already a very favorable situation for performing differential side-channel analysis. An attacker can select a static key and not only knows the varying input but can even control it. The 3DES cipher is used in 2-key mode and consequently consists of a DES encryption with the first eight key bytes, a DES decryption with the second eight key bytes and again an encryption with the first eight bytes of the key. Therefore, the well-studied DES algorithm has to be attacked twice in order to recover the entire 16-byte key. This two stage process starts with the DES encryption of the terminal random number and continues with the DES decryption of the previous result. The processing of the card random number is not even necessary to consider.

The result of the session key creation is not revealed to the terminal but is directly used in the next 3DES operation. Here the signed certificate and the counter data are

encrypted to form the appendix, which can be extracted from the command response. If a side-channel attack is successful, the recovered key has to be verified. The first possibility is to try and perform an operation with it. For instance, if a debit key is recovered, a debit command can be sent to the card and if accepted signals a successful attack. The second possibility is to perform a read purse 'Status (w)' command, derive the potential session key and try to decrypt the appendix in the response. The problem is that the signed certificate is not known, since it is never transmitted in plain. However, a successful decryption might be indicated by the counter data, which is expected to not have a random nature (which would very likely be the result of an incorrect decryption) but rather a structured appearance. Analyzing a newly purchased ez-link Card that hasn't been used for any transactions yet has the advantage that the counters are expected to be near their initial values, which might be easier to detect.

If a successful side-channel attack has been set up, the read purse 'Status (w)' command allows the recovery of all debit, autoload and signature keys on the card, since the fixed key can be individually selected during each command by specifying the Kf and Kn parameters. The only exception are credit keys. In order to recover them, the SCA scenario has to be applied to the credit command. This is realistic, since again the session key creation is attacked. The only difference is that the card random number is encrypted first in this case. Since the card generates random values, a differential side-channel analysis attack can be performed as usual. To do so, an attacker issues an invalid credit request, since without a valid credit key the cryptogram that is part of the parameters cannot be correctly calculated. However, the card must derive the real session key in order to check the validity of the request. Although the card returns an error status code, the session key creation happening before the error response can be attacked. Since there is no difference mentioned in the standard between the two credit keys needed for each credit operation, each of them can be recovered the same way simply by adapting the CKf_1 and CKn_1 identifiers in the command parameters.

One reason why the read purse command is favorable to use, especially during the equipment setup and measurement fine-tuning phase, is that it allows to selectively turn the cryptographic operations on and off. When issuing a 'Status (w/o)' command, the card does not perform any cryptographic calculations at all. When sending a 'Status (w)' command, the same operations are performed as in the previous case but complemented with encryptions. This way, it is very easy to identify the actual cryptographic parts in the measurements, because both command variants can be compared and parts that do not show up in the 'Status (w/o)' measurements but do in the 'Status (w)' signal are very good candidates to examine further.

The scenario that can be constructed from the CEPAS standard fulfills the requirements defined previously. An encryption and decryption round of DES is attacked to recover the full 16-byte 2TDES key, which is a chosen static card key. The input to the DES operations is known and random. Furthermore it is possible to attack all existing keys on a card and to verify potential key candidates. The next step is to select an appropriate side-channel and set up a measurement environment.

Setup Measuring the power consumption of a device that is powered from and communicates over a contactless interface poses various challenges. The current consumption can usually not be measured directly, since the chip does not provide supply and ground pins to the outside world. Instead a high frequency electromagnetic field delivers energy to the card through inductive coupling. The reader emanates a field with a particular strength

and if a tag is present, the field induces a voltage in its antenna, which is then rectified and stabilized in order to be used as a supply voltage of the circuit. Through the principle of inductive coupling, the tag drains energy from the reader field, which in turn gets a bit weaker compared to the original field strength. As any other electric circuit, the card emanates an electromagnetic field as current flows through the circuit. In the case of a CMOS circuit, the emanation varies with the processed data.

There are typically two measurement approaches to record signals that carry side-channel relevant information in an RFID setup. The first exploits the small variations of the reader field due to the changing current consumption of the tag. This is possible because of the coupling of the antennas, which causes the reader field to be weaker, if the tag drains more energy. For this technique the reader field itself is measured. The second approach is measuring the electromagnetic emanation of the chip, which is followed in this experiment.

The main issues in both cases are caused by the strong reader field, which is orders of magnitude stronger than the small variations caused by the tag. When measuring the reader field or the EM emanation directly, the input range of the oscilloscope must be large enough to capture the entire signal swing of the reader field. Consequently, the small variations on top of it suffer from the resulting low resolution and large quantization error. However, only these weaker parts contain the information exploited during a side-channel attack. Therefore, a preprocessing step is necessary to reduce the impact of the reader field and thus enable measuring only the small variants with sufficient resolution. This can be done by suppressing the reader field frequency of 13.56 MHz together with its harmonics and demodulating the signal down to the baseband region. In this experiment, the field is only suppressed but not demodulated. An appropriate demodulation circuit is presented by Kasper et al. in [106].

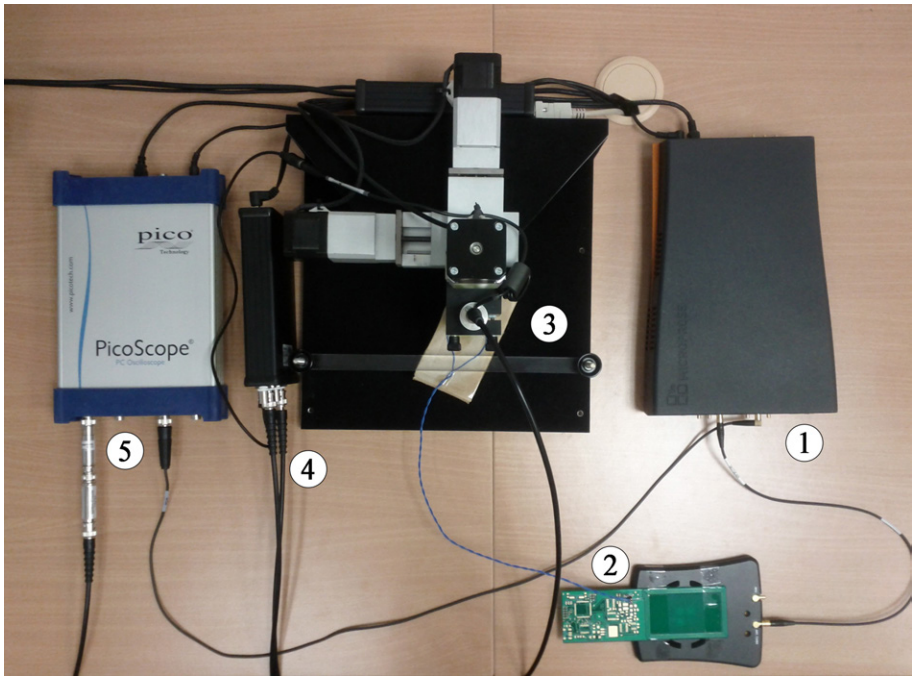
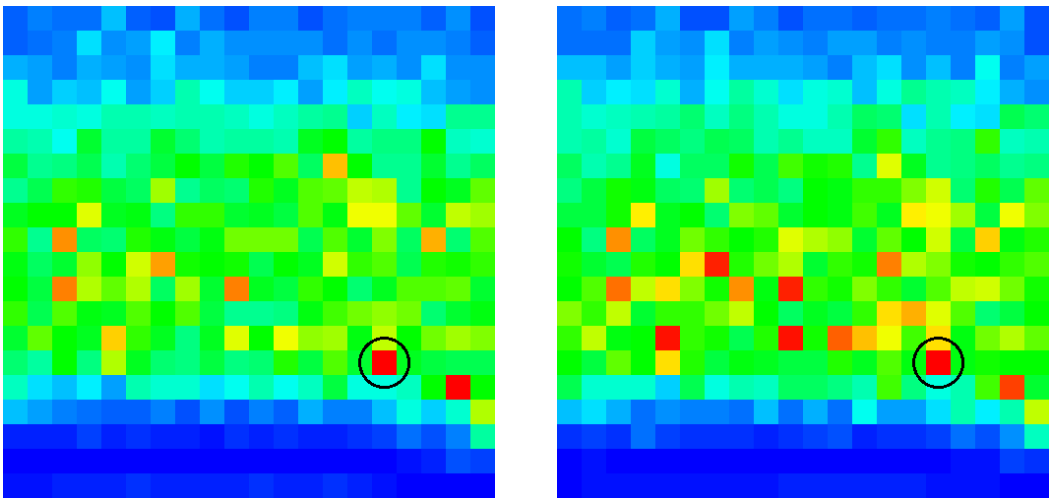


Figure 5.10: Measurement Setup for Differential EM Analysis.

In the experiments presented here, the electromagnetic emanation of the ez-link Card is measured. Figure 5.10 shows the measurement setup. A computer not shown in the picture controls the entire measurement process and also preforms the side-channel attacks afterwards with the software suite Inspector v4.6 by Riscure. To record a trace, a Micropross MP300 TCL2 universal RFID reader (1) is used to send a read purse command and trigger a 3DES operation. In order to further minimize the influence of the radiated reader field, the chip contained in an ez-link Card has been removed and connected to a separate antenna via a twisted cable of approximately 35 cm length (2). Although being part of a printed circuit board, as shown in the figure, the ID-1 antenna is the only part that is used from it. The chip together with the connected cable is then placed under the EM probe (3), which is mounted on an XYZ-table in order to freely and precisely move it over the surface of the chip package. The probe has a sensitivity of 20 mV/μT @ 1 MHz, a bandwidth of 1 GHz and a spatial resolution of 1 mm^2. The chip itself is facing the probe with its back side. The acquired signal is then fed into a multi-notch filter that suppresses all 13.56 MHz components up to the 4th harmonic (4). After that the signal is bandwidth-limited by a passive 50 MHz low-pass filter and finally recorded by a PicoScope 5203 oscilloscope (5). From there the digitized measurement is transfered back to the computer that initiated it. [107]

Measurements A first important step is setting up the measurement environment to achieve the best possible trace quality. Therefore the XYZ-table is used to scan the surface of the chip package in a fixed plane to find a location where the EM emanation is most prominent. The read purse 'Status (w)' command is used to stimulate the card. For each read purse request the time of 4.78 ms between the last reader and the first card modulation is recorded. The majority of the chip surface is scanned in a 20x20 array, only small parts at the sides of the chip that are blocked by the wires connecting the antenna are skipped. For each point in the array, one trace is measured and converted into the frequency domain to analyze the spectral intensity of its signal components. This metric is then used to plot the pictures shown in Figure 5.11 and to assess which location offers the strongest EM emanation and is thus best to choose for further investigations.



(a) 10 MHz to 12 MHz Band. (b) 15 MHz to 18 MHz Band.

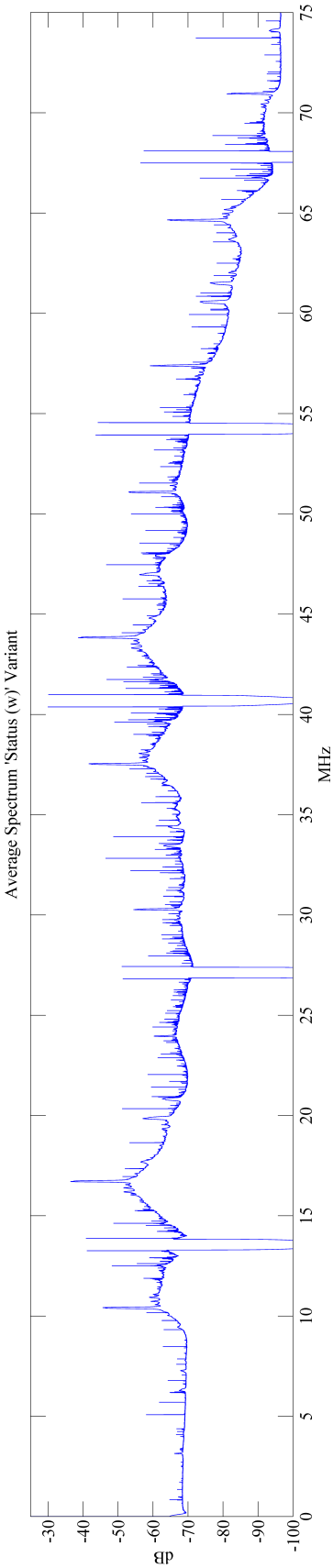
Figure 5.11: XY Measurements of the Read Purse 'Status (w)' Command.

Since no demodulation circuit is used to process the measured signal, the electromagnetic emanation of the chip is modulated onto the carrier frequency of 13.56 MHz, as shown in Figure 5.12. This is why the spectral intensity of the 10 MHz to 12 MHz as well as the 15 MHz to 18 MHz band is analyzed. In this process, the average amplitude of all spectral components of the given band is calculated and displayed as a color-coded data point on the location where the trace was recorded. A blue color indicates a lower value while a red color represents a higher value. Each frequency band covers one of the modulation sidelobes, which are symmetrically located left and right of the fundamental carrier frequency. The fact that the upper sidelobe is selected by a 3 MHz window instead of a 2 MHz one that covers the lower sidelobe does not perceivably influence the spectral intensity plots and thus the decisions based on them.

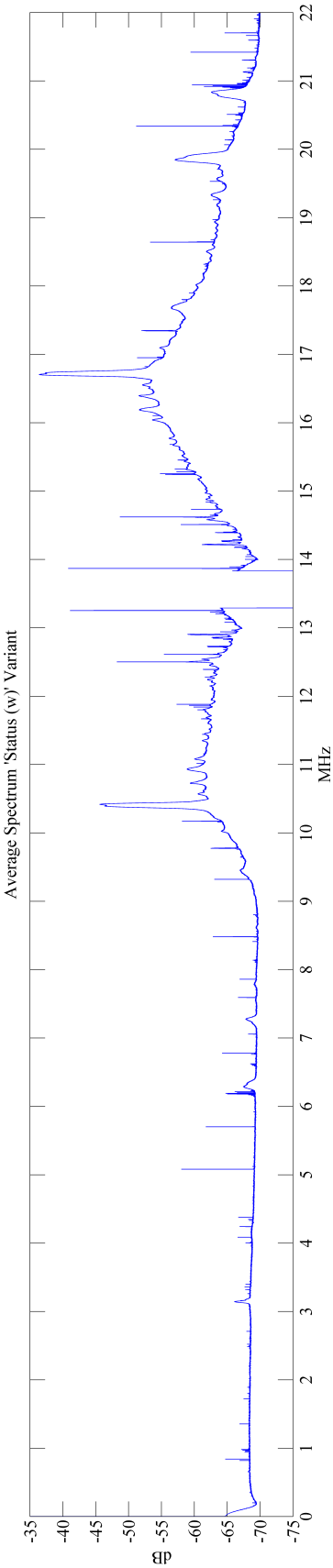
As can be clearly seen in the plots, the selected frequency bands are weak on the top and bottom of the chip, which is indicated by large blue areas. The middle part shows a solid signal level illustrated by a green area, inside which orange and red hotspots appear. The overall structure of the picture can be explained by the location of the die at the center of the chip package. Since the IC package is between the EM probe and the actual chip die and a sampling rate of 250 MS/s is used, the location of the hotspots hardly map to any specific internal circuit structures. As concluded by Heyszl et al. in [65], direct access to the die as well as a very high measurement resolution >1 GS/s are recommended for localized EM measurements. Nevertheless, the identified hotspots provide a sufficiently strong signal that further investigations can be conducted. The circle in the figures indicate the location that was selected, because it showed a continuously strong signal for all modulation sidelobes.

Because of the advantageous property to selectively turn the cryptographic operations on and off, measurements of the 'Status (w/o)' as well as the 'Status (w)' variant of the read purse command are performed. Again a sampling rate of 250 MS/s is used to record 10,000 traces for each command. As already mentioned, it takes the card 4.78 ms to respond after a 'Status (w)' command has been sent. In contrast, a 'Status (w/o)' response is received after 1.92 ms. The trace post-processing only involves applying an inverse notch filter provided by the measurement software that equalizes the frequency band distortions introduced by the multi-notch filter in the measurement setup.

Figure 5.12 shows the spectral composition averaged over all 10,000 traces of the 'Status (w)' read purse command. The 'Status (w/o)' spectrum is not shown, since there is no observable difference to the 'Status (w)' variant. The suppression of the 13.56 MHz carrier and its harmonics by the multi-notch filter can be clearly seen as big drops in the spectrum. The signal attenuation starting at 50 MHz caused by the low-pass filter is also clearly shown. The modulation of the signal emanated by the chip onto the HF frequency of 13.56 MHz is especially strong at the first and third carrier harmonic, but nevertheless can be seen at every harmonic. Approximately 3.15 MHz away in each direction from the harmonics two strong sidelobes are visible (most prominent: 10.41 MHz, 16.71 MHz, 37.53 MHz, 43.83 MHz). Their location around the fundamental carrier component of 13.56 MHz is the reason for the frequency band choices during the XY scan. An indicator that also the second harmonic of the signal emanated by the chip is modulated onto the HF field are yet another two sidelobes 6.3 MHz away from the harmonics of the carrier, which is twice the distance of the stronger sidelobes. This can be seen in Figure 5.12b, where the first 22 MHz of the spectrum are displayed in more detail. Note that the peak below 20 MHz is the upper sidelobe belonging to the 13.56 MHz carrier, whereas the peak above 20 MHz is already the lower sidelobe belonging to the second carrier harmonic 27.12 MHz.



(a) Full.



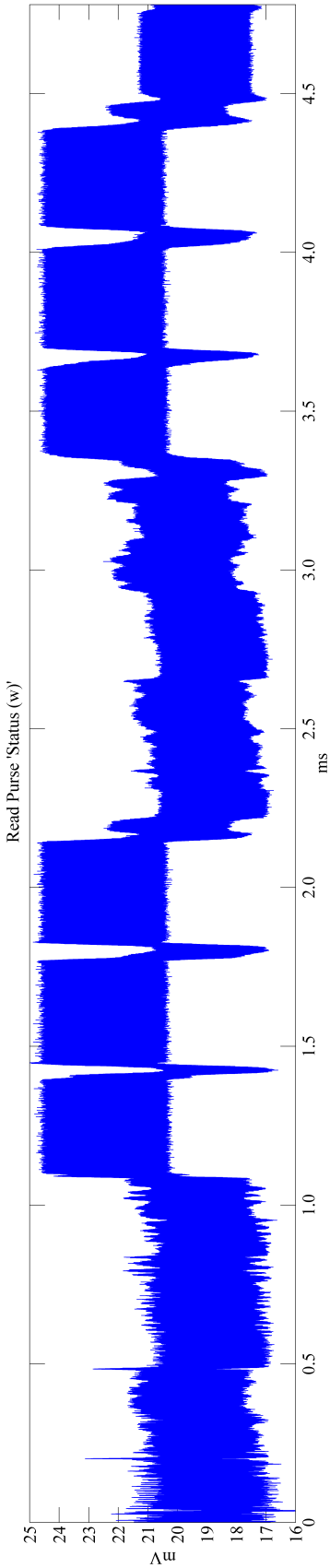
(b) Zoom.

Figure 5.12: Signal Spectrum of 'Status (w)' Read Purse Command.

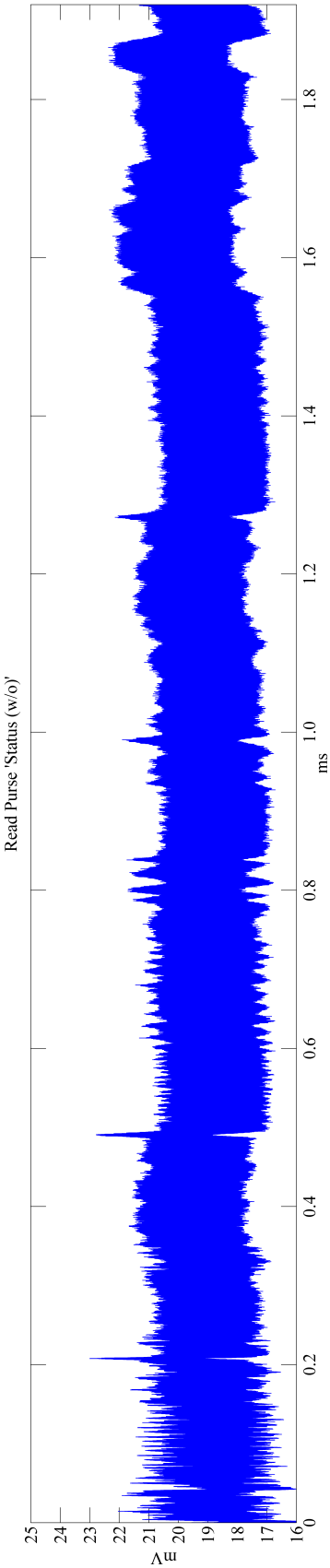
The region of the spectrum below 10 MHz shows a small peak around 3.15 MHz, which is the assumed fundamental chip emanation frequency, one around 6.30 MHz, which would consequently be the second harmonic of it, and one around 7.3 MHz, which is already the modulated second harmonic (lower sidelobe) of the chip emanation belonging to the 13.56 MHz carrier. The small peak at 17.7 MHz remains unclear.

Figure 5.13 illustrates both read purse commands in time domain representation. Since the recorded traces showed acceptable alignment in both cases, the absolute value of the traces was taken and averaged afterwards to form the signals shown in the pictures. Here the difference between the two cases is very prominent. Note that the time scale is different and that the 'Status (w/o)' trace with 1.92 ms is significantly shorter than the 'Status (w)' trace with 4.78 ms. Pattern matching techniques showed that the shorter 'Status (w/o)' variant appears in segments before, in between and after the two very distinctive pattern that dominate the look of the 'Status (w)' trace. This is a very strong indicator that these two pattern are related to the cryptographic operations. The facts that there are two big 3DES operations, one to derive the session key and one to create the appendix, and that the pattern themselves are very similar to one another fit this picture very well.

A 3DES operation consists of two 3DES encryptions, which in turn comprise three DES calls each. A pattern consists of one shorter and two longer plateaus with a duration of approximately 300 μs and 320 μs respectively. Between them there are two peaks, the first of which is about 50 μs and the second of which is about 60 μs. The question, which part of the pattern corresponds to the 3DES encryptions, is accompanied by the uncertainty whether the cipher is implemented in software of hardware. In general, the DES algorithm is hardware oriented and can thus be more efficiently implemented in hardware than in software. In a survey published in 2007 by Eisenbarth et al. [108], implementations of different (a-)symmetric ciphers are compared. A hardware DES that trades throughput for a smaller area can encrypt one block of plaintext in 144 clock cycles. In comparison a software DES running on an 8-bit microcontroller is listed with 8,633 clock cycles per encryption. This clearly states that even a deliberately slow hardware implementation is still much faster than a software one. In [57] Rankl et al. list an average encryption/decryption duration of 130 μs for one input block processed by a dedicated 3DES hardware unit on a smart card running with 3.5 MHz. This corresponds to 455 clock cycles and is somewhat close to three consecutive calls of the 144-cycles solution stated earlier. Assuming that the 3DES is implemented in software as three consecutive calls of the DES, a total of at least 51,798 clock cycles would be needed to encrypt two blocks. This is already a very high number, but still neglects the time spent storing and loading data. If the encryptions should fit the length of around 1.05 ms for an entire pattern, the clock frequency must be around 49.33 MHz, which is already an unusually high internal smart card clock that can hardly be increased to account for the still missing data loading and storing tasks. This observation shifts the focus to a dedicated DES hardware implementation, which can be found on many smart card platforms today. A 3DES encryption needing only 455 cycles can fit in different parts of the pattern. If one 320 μs plateau corresponds to one encryption, the required clock frequency would be around 1.42 MHz. If a 50 μs peak is considered, the clock would be 9.1 MHz. Given these results and factoring in an additional overhead for loading and storing data as well as potential side-channel countermeasures, a hardware implementation of the 3DES operation is more likely.

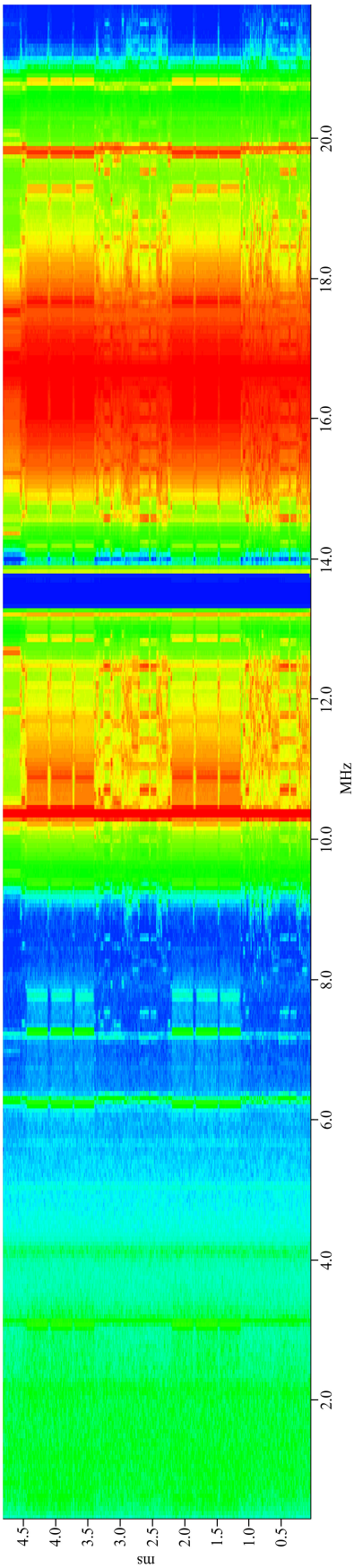


(a) With Cryptographic Operation.

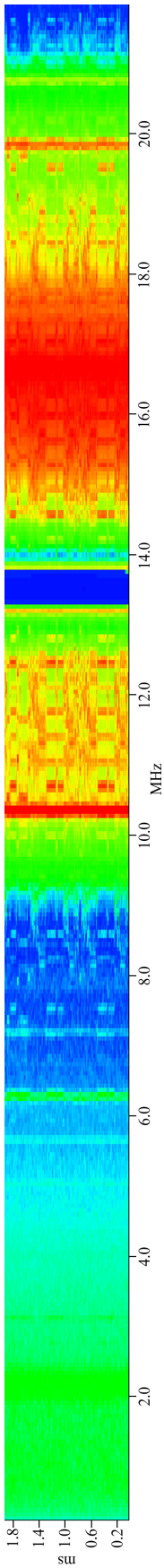


(b) Without Cryptographic Operation.

Figure 5.13: Time Domain Comparison between Read Purse Commands.



(a) 'Status (w)'.



(b) 'Status (w/o)'.

Figure 5.14: Spectrogram Comparison between Read Purse Commands.

The summary of the findings gathered during this initial trace inspection process are very well depicted in the spectrogram in Figure 5.14. In these pictures, the spectrum of the signal is shown as it evolves over time. For this purpose, the signal is split into several fragments and the spectral components for each of them are calculated. The amplitude of these spectral components are color-coded and placed at the corresponding time slot in the picture, where a frequency band of approximately 0 MHz to 22 MHz is shown. A low spectral amplitude is illustrated with a blue color and a high one with a red color. The figure shows the spectrogram for both read purse commands. It can be clearly seen that the signal emanated by the chip is modulated onto the high-frequency carrier, which leads to red regions of high spectral amplitude symmetrically placed ±3.15 MHz around the 13.56 MHz carrier. Two distinct pattern appearing in the 'Status (w)' command can be clearly seen in those frequency ranges shown in 5.14a. Since they do not appear in the 'Status (w/o)' case shown in 5.14b, they are assumed to be related to the cryptographic operations. Furthermore, it can be seen that the signal of the 'Status (w/o)' command fits before and in between those distinct pattern shown in the 'Status (w)' signal. Also, the significant difference in signal strength between the modulated chip emanation around the HF carrier (red parts around 13.56 MHz) and the weak unmodulated emanation in the base band (green parts around 3.15 MHz and 6.3 MHz) is illustrated.

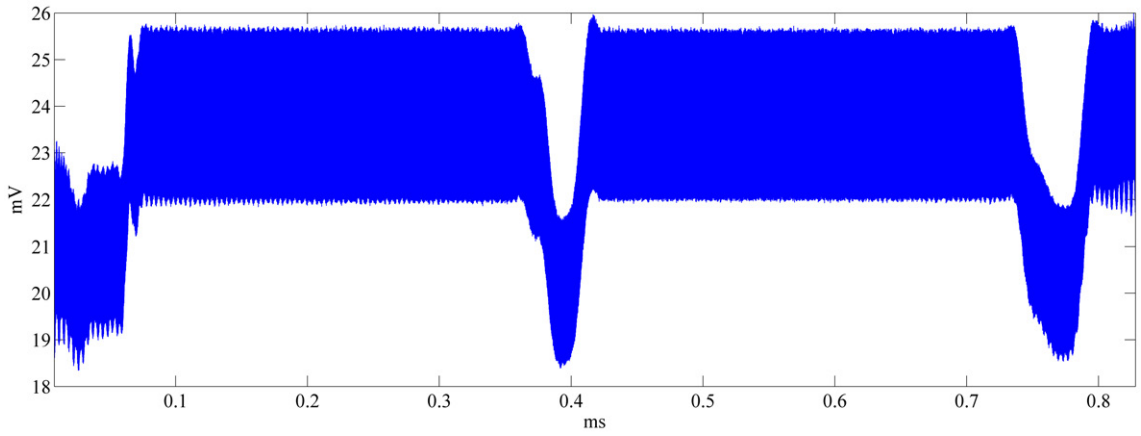


Figure 5.15: Attacked 'Status (w)' Trace Section.

Attacks In order to assess the side-channel analysis resistance of the smart card that hosts the EZ-Link purse, two attacks were performed on the part of the traces that are assumed to belong to the first 3DES operation or in other words to the session key creation. The exact time frame of 820 μs is shown in Figure 5.15. It comprises the first two plateaus and peaks of the first pattern, since those are possible parts where a 3DES encryption might take place. For the attack 500,000 traces were recorded. The only post-processing step was applying an inverse notch filter provided by the measurement software to equalize the frequency band distortions introduced by the multi-notch filter in the measurement setup. The traces showed sufficient alignment, thus this step was skipped.

The attacks themselves fall into the category of first-order correlation power analysis of the chip's electromagnetic emanation. They target the first DES encryption of the 3DES-based session key derivation. The difference between the attacks is the used power model. For the first one, the Hamming distance of the round register, $\mathrm{HD}(R_i, R_{i+1})$, is attacked, whereas for the second one the Hamming weight of the S-box output, $\mathrm{HW}(S_{out,j})$,

is targeted. This choice is explained in more detail in Section 3.4. The values of the correlation coefficients for all eight key bytes and both power models are shown in Table 5.12.

	K_i	ρ_{1st}	H_{1st}	ρ_{2nd}	H_{2nd}	d
HD(R_i, R_{i+1})	1	0.0077	05_h	0.0076	09_h	0.01299
	2	0.0076	34_h	0.0074	24_h	0.02632
	3	0.0078	22_h	0.0075	24_h	0.03846
	4	0.0082	32_h	0.0076	16_h	0.07317
	5	0.0070	18_h	0.0069	$1a_h$	0.01429
	6	0.0080	13_h	0.0077	38_h	0.03750
	7	0.0077	$1f_h$	0.0074	$1a_h$	0.03896
	8	0.0074	11_h	0.0074	35_h	0.00000
HW($S_{out,j}$)	1	0.0076	09_h	0.0075	$1b_h$	0.01316
	2	0.0085	32_h	0.0080	28_h	0.05882
	3	0.0078	$0c_h$	0.0077	39_h	0.01282
	4	0.0076	$2f_h$	0.0074	28_h	0.02632
	5	0.0078	$2c_h$	0.0077	$3f_h$	0.01282
	6	0.0078	$3b_h$	0.0074	$2d_h$	0.05128
	7	0.0072	10_h	0.0072	$1f_h$	0.00000
	8	0.0085	$1a_h$	0.0079	30_h	0.07059

Table 5.12: Results of the DEMA Attack After 500,000 Traces.

K_i denotes the number of the key byte. Since the first DES encryption of the 3DES algorithm is attacked, the first eight key bytes of the full 16-byte 3DES key are listed here. ρ_{1st} and ρ_{2nd} denote the highest and second-highest correlation coefficients, whereas H_{1st} and H_{2nd} label their corresponding key hypotheses. The metric given in equation 5.6 is used to normalize the distance between the first and second correlation coefficient. This metric indicates how significant the results are. Values towards '1' mean that the corresponding key hypothesis clearly stands out from the rest and is therefore likely to be correct. Values towards '0' indicate that no key hypothesis sticks out and that the results are inconclusive.

$$d = \frac{\rho_{1st} - \rho_{2nd}}{\rho_{1st}} \ , \ 0 \leq d \leq 1 \tag{5.6}$$

As can be seen in the table, the distance is always below 0.1 and therefore very close to zero, which indicates that no key hypothesis stands out. This renders the results inconclusive and does not reveal any vulnerability of the smart card to a first-order correlation power analysis. If the first attack would be successful, the next step would be to target the DES decryption of the 3DES algorithm in order to recover the second eight key bytes of the full 16-byte key.

5.6 Conclusion

In this chapter the smart card platform that constitutes the basis of the ez-link Card has been examined and although not identified revealing manufacturer and model, indicators have been presented that point towards what is commonly referred to as a high-security

chip. The shielding mesh as the top metal layer of the die as well as the smart card's resistance to first-order correlation power analysis attacks support this assumption. Furthermore the results suggest that the 3DES cipher is implemented as a dedicated hardware module. The only connection to a specific manufacturer was given in early October 2013 at a media briefing held at Infineon Technologies Asia Pacific, where Infineon was named as a major supplier for CEPAS compliant smart cards since 2009 [86].

The Contactless e-purse Application Specification defines the main properties of the electronic purse found on today's ez-link Cards. The focus lies on three commands that allow the purse to be increased, decreased and queried for status and transaction information. Although a mutual authentication is required to modify the purse balance, a terminal is not required to proof its authenticity when requesting status and transaction information. The data gathered via this leak has been shown to be comprehensive, easy to interpret and personal to the card holder. The fact that it can be accessed by anyone from a considerable distance and without the knowledge or consent of the owner constitutes a clear privacy violation as defined in the beginning of this thesis. The profiling techniques applied to a real-world data set illustrate the potential dangers arising from the data leak and shall stimulate an open discussion about this privacy impairment. Possible short-term solutions like radio-frequency blocking equipment and the usage of multiple ez-link Cards at a time are an important immediate response, since open source tools to exploit the information leak are already publicly available. Privacy enhancements on system design level can help mitigate the problem in the long run.

Apart from well-defined security concepts like mutual authentication or the usage of freshly generated session keys, CEPAS does not specify how to actually tackle broader challenges like key management. By giving recommendations, for instance to diversify keys for each card or to store master keys in SAMs, it leaves much to the implementation. The well-integrated idea of having multiple keys for different tasks enables flexible and fine-grained key management, but is at best an optional feature the implementation chooses to take advantage of, since the standard allows the reduction of the key set to one key. The discovery of 10 key records spread over four key files, however, does not indicate such a severe key set reduction for an EZ-Link purse. Nevertheless, the exact usage of all discovered keys remains inconclusive.

Furthermore, it has been shown that CEPAS facilitates differential side-channel analysis. A scenario allowing to target any key used by the purse has been presented and shown to be feasible by conducting a first-order correlation power analysis attack on the chip's electromagnetic emanation. Especially the unrestricted access to purse information has been found to be useful regarding measurement equipment setup and fine-tuning as well as identification of cryptographic operations in the traces. This leads to the conclusion that side-channel analysis resistance must be considered when implementing the CEPAS standard. The side-channel attacks on the current smart card platform did not reveal weaknesses in this regard.

Chapter 6

Conclusion

This chapter summarizes the findings and contributions of this thesis and based on that gives ideas and promising topics for further research.

6.1 Summary

Following the goals defined at the beginning of this thesis, the starting point of the investigations were the two contactless smart cards used in the EZ-Link system. The different directions these two topics developed in match the intention of laying the proper groundwork for future research projects.

For the underlying smart card chip of the Standard Ticket, strong indicators were given that reveal the model and manufacturer. Based on this discovery, the focus was put on the chip's features and their usage by the EZ-Link system. It could be shown that basic mechanisms are properly configured and used by the system, while more sophisticated features are not made use of. The lack of strong authentication inherent to many memory cards poses the threat of denial of service attacks. This was shown during an experiment to also apply to the Standard Ticket. The presented attack renders a ticket invalid and thus impairs the availability of the system. Because of limitations like these, the back end system has to compensate what cannot be ensured by the smart card and has consequently been tested for basic security checks. During these experiments it was possible to reuse a Standard Ticket for multiple journeys without actually paying for them, which is a clear security policy violation. In contrast, the current state of knowledge does not suggest any privacy violations with practical impact. This is due to the usage restrictions of a Standard Ticket defined by the system operator and that based on current knowledge no personal information is stored on a ticket. However, without knowing the exact meaning of the ticket content, the final assessment regarding the Standard Ticket's security and privacy aspects remains partially inconclusive.

For the smart card chip constituting the basis of an ez-link Card indicators were presented that point towards a high-security smart card controller. The shielding mesh discovered during the decapsulation of the chip as well as its resistance to the presented first-order differential side-channel analysis attacks support this assumption. Consequently, the focus was shifted to the implementation of the electronic purse hosted on the card. The Contactless e-purse Application Specification, which defines basic properties of the purse and how to access it, was examined in particular. Although cryptographic authentication mechanisms are part of the standard, information about the purse and past transactions

can be collected by unauthorized parties. The data that is leaked has been shown to be comprehensive, easy to interpret and personal to the card holder. The fact that it can be accessed by anyone from a distance long enough to have practical impact and without the knowledge or consent of the owner constitutes a clear privacy violation. A publicly available open source implementation that can be used to exploit the information leak adds to the urgency of the problem. To illustrate the issue, a comprehensive data set was gathered from an actual ez-link Card over a period of six months and techniques to process the data and infer sensitive personal information from it were discussed. In addition, it was shown that a side-channel analysis scenario can be well arranged with the CEPAS standard. Although the conducted attacks did not reveal the cryptographic keys stored on the analyzed ez-link Card, they are a proof-of-concept showing that the platform's side-channel analysis resistance must be considered when implementing the CEPAS standard on it.

Except for the side-channel analysis equipment, the tools necessary for the assessment of both smart cards are low-cost, off-the-shelf and easy to use. The Android software that has been developed and the Google Nexus S smartphone running it illustrate the minor requirements. This renders the presented attacks to be easily realizable in practice, which poses an immediate threat to the system and requires a rapid response to the problems. Therefore, the results presented in this thesis have been communicated to the affected parties prior to publication and possible short- as well as long-term solutions are given in the corresponding chapters of this thesis.

6.2 Future Work

The biggest question when it comes to the Standard Ticket is, what the data packets stored in its memory are made of. This is consequently one of the topics further work could tackle. In addition, more advanced experiments, for instance ticket cloning with an RFID emulator, and reverse engineering of the back end system might reveal promising new research directions.

Regarding the ez-link Card, there are multiple topics to analyze in more detail. The decapsulation of the chip could be continued by removing the shielding mesh and taking a closer look at the metal layers below. An exposed die can also be used in more invasive attacks like fault analysis or localized EM measurements. Side-channel analysis techniques could be again applied to these new measurements and upon discovery of countermeasures may be improved to more advanced techniques like high-order attacks.

Considering the CEPAS standard, identifying more implementations of it on other platforms and applying the findings of this thesis to them could give a more comprehensive insight in the actual realization of the electronic purse. This is interesting, because the standard leaves a considerable freedom to the purse implementation. This could be accompanied by reverse engineering of the back end system in order to shed light on topics like key derivation algorithms or secure key storage in terminals.

As mentioned on their website [48], the progressing advancement of EZ-Link in the area of NFC-enabled smartphone platforms opens up a completely new environment in which the electronic purse is embedded. This is also a potential new research direction. Also, the efforts of analyzing the web based services, e.g. EZ-Online, by Kerschbaum et al. [8] could be continued.

Appendix A

Profiling Data Set

[118]	[Thu Apr 11 08:47:32 SGT 2013]	[+00.12]	[SVC 179]
[049]	[Thu Apr 11 08:30:02 SGT 2013]	[-00.42]	[SVC 179]
[048]	[Thu Apr 11 08:28:25 SGT 2013]	[-01.17]	[CLE-BNL]
[048]	[Thu Apr 11 00:27:06 SGT 2013]	[-01.35]	[RFP-CLE]
[048]	[Wed Apr 10 19:14:46 SGT 2013]	[-00.73]	[CTH-RFP]
[048]	[Wed Apr 10 19:08:08 SGT 2013]	[-01.04]	[PNR-CTH]
[118]	[Wed Apr 10 18:26:43 SGT 2013]	[+00.10]	[SVC 179]
[049]	[Wed Apr 10 18:16:28 SGT 2013]	[-00.83]	[SVC 179]
[118]	[Wed Apr 10 08:35:09 SGT 2013]	[+00.12]	[SVC 179]
[049]	[Wed Apr 10 08:12:11 SGT 2013]	[-00.42]	[SVC 179]
[048]	[Wed Apr 10 08:10:32 SGT 2013]	[-01.17]	[CLE-BNL]
[048]	[Tue Apr 09 19:23:34 SGT 2013]	[-00.62]	[PNR-CLE]
[118]	[Tue Apr 09 19:06:01 SGT 2013]	[+00.10]	[SVC 179]
[049]	[Tue Apr 09 18:57:20 SGT 2013]	[-00.83]	[SVC 179]
[118]	[Tue Apr 09 09:18:13 SGT 2013]	[+00.12]	[SVC 179]
[049]	[Tue Apr 09 09:01:06 SGT 2013]	[-00.42]	[SVC 179]
[048]	[Tue Apr 09 09:00:00 SGT 2013]	[-01.17]	[CLE-BNL]
[048]	[Mon Apr 08 19:04:22 SGT 2013]	[-00.52]	[BNL-CLE]
[049]	[Mon Apr 08 18:13:54 SGT 2013]	[-00.83]	[SVC 179]
[118]	[Mon Apr 08 08:40:56 SGT 2013]	[+00.12]	[SVC 179]
[049]	[Mon Apr 08 08:19:05 SGT 2013]	[-00.42]	[SVC 179]
[048]	[Mon Apr 08 08:17:28 SGT 2013]	[-01.17]	[CLE-BNL]
[048]	[Sat Apr 06 23:55:27 SGT 2013]	[-01.35]	[RFP-CLE]
[048]	[Sat Apr 06 18:16:15 SGT 2013]	[-00.64]	[CCK-CTH]
[117]	[Sat Apr 06 17:33:04 SGT 2013]	[+50.00]	[CCB GTM]
[118]	[Sat Apr 06 17:28:51 SGT 2013]	[+00.04]	[SVC 927]
[049]	[Sat Apr 06 17:10:19 SGT 2013]	[-01.27]	[SVC 927]
[118]	[Sat Apr 06 12:17:08 SGT 2013]	[+00.39]	[SVC 927]
[049]	[Sat Apr 06 11:55:28 SGT 2013]	[-01.66]	[SVC 927]
[048]	[Sat Apr 06 11:03:35 SGT 2013]	[-01.27]	[CLE-CCK]

Table A.1: List of Coherent EZ-Link Transactions.

[118]	[Fri May 17 18:48:57 SGT 2013]	[+00.10]	[SVC 179]
[049]	[Fri May 17 18:44:40 SGT 2013]	[-00.83]	[SVC 179]
[048]	[Wed May 15 18:53:01 SGT 2013]	[-00.54]	[PNR-BBT]
[118]	[Wed May 15 18:33:18 SGT 2013]	[+00.10]	[SVC 179]
[049]	[Wed May 15 18:25:43 SGT 2013]	[-00.83]	[SVC 179]
[160]	[Mon May 13 19:31:16 SGT 2013]	[-01.30]	[CFDB]
[048]	[Mon May 13 18:53:38 SGT 2013]	[-00.54]	[PNR-BBT]
[118]	[Mon May 13 18:34:56 SGT 2013]	[+00.10]	[SVC 179]
[049]	[Mon May 13 18:25:43 SGT 2013]	[-00.83]	[SVC 179]

Table A.2: List of Fragmented EZ-Link Transactions.

The list given in Table A.1 contains 30 coherent transactions performed with the ez-link Card of the author over a period of five days, from Saturday, April 06^{th} 2013, to Thursday, April 11^{th} 2013. The list given in Table A.2 contains nine fragmented transactions spanning over one week, from Monday, May 13^{th} 2013, to Friday, May 17^{th} 2013, performed with the same ez-link Card. The entries are already converted into a human readable format. Note that all numbers are in decimal representation. An entry starts with the transaction type, which is followed by the date and time, the amount and the user data already shown as a string in ASCII encoding. The entries are listed with the latest one first in descending order.

Bibliography

[1] Land Transport Authority of Singapore. Singapore Land Transport: Statistics In Brief 2013. URL: http://www.lta.gov.sg/content/dam/ltaweb/corp/PublicationsResearch/files/FactsandFigures/Stats_in_Brief_2013.pdf. Report. Accessed on Oct 14 2013.

[2] Monetary Authority of Singapore. Stored Value Facility Guidelines. URL: http://www.mas.gov.sg/singapore-financial-centre/payment-and-settlement-systems/payment-media/stored-value-facilities.aspx. Article, June 2006. Accessed on Oct 14 2013.

[3] Monetary Authority of Singapore. H1 2013 Retail Payment Statistics for Selected Payment Systems in Singapore. URL: http://www.mas.gov.sg/~/media/MAS/Singapore%20Financial%20Centre/Why%20Singapore/Payment%20and%20Settlement%20Systems%20redirect%20pages/H1%202013%20Retail%20Payment%20Statistics.pdf. Report. Accessed on Oct 14 2013.

[4] Monetary Authority of Singapore. Stored Value Facilities. URL: http://www.mas.gov.sg/singapore-financial-centre/payment-and-settlement-systems/payment-media/stored-value-facilities.aspx. Article. Accessed on Oct 14 2013.

[5] EZ-Link Pte Ltd. EZ-Link Launches Chingay 2014 Commemorative ez-link Cards set. URL: http://www.ezlink.com.sg/corporate/news.php?id=84. Press Release, February 2014. Accessed on Feb 24 2014.

[6] EZ-Link Pte Ltd. ez-Link Card Key Features. URL: http://www.ezlink.com.sg/ez-link-card/key-features.php. Article. Accessed on Oct 14 2013.

[7] Ross Anderson. Why Cryptosystems Fail. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 215–227. ACM, 1993.

[8] Florian Kerschbaum, Hoon Wei Lim, and Ivan Gudymenko. Privacy-Preserving Billing for e-Ticketing Systems in Public Transportation. *IACR Cryptology ePrint Archive*, 2013:206, 2013.

[9] Karsten Nohl and Henryk Plötz. Mifare, Little Security Despite Obscurity. Presentation at Chaos Computer Congress, 2007.

[10] Karsten Nohl, David Evans, Starbug, and Henryk Plötz. Reverse-Engineering a Cryptographic RFID Tag. In *USENIX Security Symposium*, pages 185–194, 2008.

[11] Gerhard de Koning Gans, Jaap-Henk Hoepman, and Flavio D. Garcia. A Practical Attack on the MIFARE Classic. In *Smart Card Research and Advanced Applications*, pages 267–282. Springer, 2008.

[12] Gerhard de Koning Gans. Analysis of the MIFARE Classic used in the OV-Chipkaart project. *Master's thesis, Radboud University Nijmegen*, 2008.

[13] Flavio D. Garcia, Gerhard de Koning Gans, Ruben Muijrers, Peter Van Rossum, Roel Verdult, Ronny Wichers Schreur, and Bart Jacobs. Dismantling MIFARE Classic. In *Computer Security-ESORICS 2008*, pages 97–114. Springer, 2008.

[14] Flavio D. Garcia, Peter van Rossum, Roel Verdult, and Ronny Wichers Schreur. Wirelessly Pickpocketing a Mifare Classic Card. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 3–15. IEEE, 2009.

[15] Wee Hon Tan. Practical Attacks on the MIFARE Classic. *Master's thesis, Imperial College London*, 2009.

[16] Harald Welte. Reverse Engineering a real-world RFID payment system. URL: `http://events.ccc.de/congress/2010/Fahrplan/attachments/1772_easycard.pdf`. Presentation at Chaos Computer Congress, 2010. Accessed on Oct 17 2013.

[17] Timo Kasper, Michael Silbermann, and Christof Paar. All You Can Eat or Breaking a Real-World Contactless Payment System. In *Financial Cryptography and Data Security*, pages 343–350. Springer, 2010.

[18] Steve Bono, Matthew Green, Adam Stubblefield, Ari Juels, Avi Rubin, and Michael Szydlo. Security Analysis of a Cryptographically-Enabled RFID Device. In *14th USENIX Security Symposium*, volume 1, page 16, 2005.

[19] Roel Verdult. Security analysis of RFID tags. *Masters thesis, Radboud University Nijmegen*, 2008.

[20] David Oswald and Christof Paar. Breaking Mifare DESFire MF3ICD40: Power Analysis and Templates in the Real World. In *Cryptographic Hardware and Embedded Systems–CHES 2011*, pages 207–222. Springer, 2011.

[21] Timo Kasper, Ingo von Maurich, David Oswald, and Christof Paar. Cloning Cryptographic RFID Cards for 25$. *5th Benelux Workshop on Information and System Security, WisSec 2010, Nijmegen, the Netherland*, 2010.

[22] Michael Hutter, Stefan Mangard, and Martin Feldhofer. Power and EM Attacks on Passive 13.56 MHz RFID Devices. In *Cryptographic Hardware and Embedded Systems-CHES 2007*, pages 320–333. Springer, 2007.

[23] Michael Hutter, Marcel Medwed, Daniel Hein, and Johannes Wolkerstorfer. Attacking ECDSA-enabled RFID devices. In *Applied Cryptography and Network Security*, pages 519–534. Springer, 2009.

[24] Timo Kasper, David Oswald, and Christof Paar. EM Side-Channel Attacks on Commercial Contactless Smartcards Using Low-Cost Equipment. In *Information Security Applications*, pages 79–93. Springer, 2009.

[25] Thomas Korak, Thomas Plos, and Michael Hutter. Attacking an AES-Enabled NFC Tag: Implications from Design to a Real-World Scenario. In *Constructive Side-Channel Analysis and Secure Design*, pages 17–32. Springer, 2012.

[26] Patrick YK Chau and Simpson Poon. Octopus: An E-Cash Payment System Success Story. *Communications of the ACM*, 46(9):129–133, 2003.

[27] Stuart GM Bailey and Nadia Caidi. How much is too little? Privacy and smart cards in Hong Kong and Ontario. *Journal of information science*, 31(5):354–364, 2005.

[28] Graham Greenleaf and Robin McLeish. Hong Kongs Privacy Enforcement: Issues Exposed, Powers Lacking. *Privacy Laws & Business International Report*, 116:25–28, 2012.

[29] Rina CY Chung. Hong Kong's "Smart" Identity Card: Data Privacy Issues and Implications for a Post-September 11th America. *Asian-Pacific L. & Pol'y J.*, 4:442–524, 2003.

[30] Andrew Lee, Timothy Lui, and Bryon Leung. Security Analysis of the Octopus System. URL: `http://courses.ece.ubc.ca/412/previous_years/2007_1_spring/modules/term_project/reports/2007/security_analysis_of_octopus_smart_card_system.pdf`. Report. Accessed on Oct 14 2013.

[31] L.S.K. Sim, E.A.C. Seow, and S. Prakasam. Implementation of an Enhanced Integrated Fare System for Singapore. In *Proc. RTS Conference, Singapore*, 2003.

[32] Xiaobo Yang. Advanced Public Transport System in Singapore. In *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, volume 2, pages 1660–1663. IEEE, 2003.

[33] United Nations International Community. The Universal Declaration of Human Rights, 1948.

[34] Alan F. Westin. Privacy and Freedom. *New York: Atheneum*, 1967.

[35] codebutler. FareBot. URL: `https://play.google.com/store/apps/details?id=com.codebutler.farebot`. Android Application. Accessed on Feb 18 2014.

[36] Eric Butler. codebutler / farebot. URL: `https://github.com/codebutler/farebot`. Article. Accessed on Feb 18 2014.

[37] EZ-Link Pte Ltd. Where to Use. URL: `http://www.ezlink.com.sg/ez-link-card/where-to-use.php`. Article. Accessed on Oct 24 2013.

[38] Transit Link Pte Ltd. About Us. URL: `http://www.transitlink.com.sg/Aboutus.aspx`. Article. Accessed on Dec 07 2013.

[39] Land Transport Authority of Singapore. Train System Map 2012. URL: `http://www.lta.gov.sg/content/dam/ltaweb/corp/PublicTransport/img/MRT_SysMp_Dec12.jpg`. Picture, 2012. Accessed on Oct 28 2013.

[40] EZ-Link Pte Ltd. Islandwide Top-up points. URL: `http://www.ezlink.com.sg/top-up/island-wide-points.php`. Article. Accessed on Oct 24 2013.

[41] EZ-Link Pte Ltd. EZ-Online. URL: `http://www.ezlink.com.sg/top-up/ez-online.php`. Article. Accessed on Oct 24 2013.

[42] EZ-Link Pte Ltd. Top and Tap. URL: `http://www.ezlink.com.sg/top-up/top-tap.php`. Article. Accessed on Oct 24 2013.

[43] EZ-Link Pte Ltd. EZ-Reload. URL: `http://www.ezlink.com.sg/top-up/ez-reload.php`. Article. Accessed on Oct 24 2013.

[44] Singapore Tourist Pass. Why Singapore Tourist Pass. URL: `http://www.thesingaporetouristpass.com.sg/why-singapore-tourist-pass/`. Article. Accessed on Oct 25 2013.

[45] EZ-Link Pte Ltd. EZ-Link Season Pass. URL: `http://www.ezlink.com.sg/services/season-pass/`. Article. Accessed on Oct 25 2013.

[46] PAssion Card. About PAssion Card. URL: `http://www.passioncard.com.sg/aboutpa.aspx?bg=aboutpa&menuid=443`. Article. Accessed on Oct 25 2013.

[47] People's Association. About Us. URL: `http://www.pa.gov.sg/about-us.html`. Article. Accessed on Oct 25 2013.

[48] EZ-Link Pte Ltd. FAQs : Introduction to NFC. URL: `http://www.ezlinknfc.com/introduction-to-nfc/`. Article. Accessed on Dec 06 2013.

[49] Klaus Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley & Sons, Inc., New York, NY, USA, 2 edition, 2003.

[50] International Organization for Standardization (ISO). ISO/IEC 7810: Identification cards – Physical characteristics. (2003).

[51] International Organization for Standardization (ISO). ISO/IEC 14443-1: Identification cards – Contactless integrated circuit cards – Proximity cards – Part 1: Physical characteristics. (2008).

[52] International Organization for Standardization (ISO). ISO/IEC 14443-2: Identification cards – Contactless integrated circuit cards – Proximity cards – Part 2: Radio frequency power and signal interface. (2010).

[53] International Organization for Standardization (ISO). ISO/IEC 14443-3: Identification cards – Contactless integrated circuit cards – Proximity cards – Part 3: Initialization and anticollision. (2011).

[54] International Organization for Standardization (ISO). ISO/IEC 14443-4: Identification cards – Contactless integrated circuit cards – Proximity cards – Part 4: Transmission protocol. (2008).

[55] International Organization for Standardization (ISO). ISO/IEC 18092: Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1). (2013).

[56] International Organization for Standardization (ISO). ISO/IEC 21481: Information technology – Telecommunications and information exchange between systems – Near Field Communication Interface and Protocol -2 (NFCIP-2). (2012).

[57] Wolfgang Rankl and Wolfgang Effing. *Smart card handbook*. Wiley.com, 2010.

[58] International Organization for Standardization (ISO). ISO/IEC 7816-3: Identification cards – Integrated circuit cards – Part 3: Cards with contacts – Electrical interface and transmission protocols. (2006).

[59] International Organization for Standardization (ISO). ISO/IEC 7816-4: Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange. (2013).

[60] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, volume 31. Springer, 2007.

[61] Hubert Kaeslin. *Digital Integrated Circuit Design: From VLSI Architectures to CMOS Fabrication*. Cambridge University Press, 2008.

[62] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *Advances in Cryptology–CRYPTO99*, pages 388–397. Springer, 1999.

[63] Ronald L. Rivest, Adi Shamir, and Len Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[64] Auguste Kerckhoffs. La Cryptographie Militaire. *Journal des Sciences Militaires*, pages 161–191, 1883.

[65] Johann Heyszl, Dominik Merli, Benedikt Heinz, Fabrizio De Santis, and Georg Sigl. Strengths and Limitations of High-Resolution Electromagnetic Field Measurements for Side-Channel Analysis. In *Smart Card Research and Advanced Applications*, pages 248–262. Springer, 2013.

[66] National Institute of Standards and Technology. Special Publication 800-67: Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher. Revision 1. *National Institute of Standards and Technology*, 2012.

[67] Lawrence C Washington and Wade Trappe. *Introduction to Cryptography: with Coding Theory*. Prentice Hall PTR, 2002.

[68] John Gilmore. Cracking DES: Secrets of Encryption Research. *Wiretap Politics & Chip Design by the Electronic Frontier Foundation (ISBN 1565925203)*, 1998.

[69] Transit Link Pte Ltd. New Standard Ticket. URL: `http://www.transitlink.com.sg/PSdetail.aspx?ty=cat&Id=2`. Article. Accessed on Dec 07 2013.

[70] NFC Research Lab. NFC TagInfo. URL: `https://play.google.com/store/apps/details?id=at.mroland.android.apps.nfctaginfo`. Android Application. Accessed on Oct 16 2013.

[71] NXP Semiconductors. NFC TagInfo by NXP. URL: `https://play.google.com/store/apps/details?id=com.nxp.taginfolite`. Android Application. Accessed on Oct 16 2013.

[72] International Organization for Standardization (ISO). ISO/IEC 7816-6: Identification cards – Integrated circuit cards – Part 6: Interindustry data elements for interchange. (2004).

[73] Infineon Technologies AG. my-d move (NFC). SLE66R01P(N). Short Product Information. November 2011.

[74] NFC Forum. Type 2 Tag Operation Specification, Version 1.1. (2011).

[75] Identive Group. 45 x 76 mm, 80 mm pitch. 13.56 MHz Transponder Inlay (HF) ISO 14443/15693. URL: `http://www.identive-group.com/images/pdfs/datasheets/en/Identive_45x76_80_PET-AI_ACA_V2.pdf`. May 2012. Accessed on Nov 14 2013.

[76] Infineon Technologies AG. my-d move. SLE66R01P. Short Product Information. July 2009.

[77] NXP Semiconductors. MF0ICU1 - MIFARE Ultralight contactless single-ticket IC. Product data sheet. December 2010.

[78] EPCglobal GS1 AISBL. GS1 EPC Tag Data Standard 1.7. May 2013.

[79] Simson L. Garfinkel, Ari Juels, and Ravikanth Pappu. RFID privacy: An Overview of Problems and Proposed Solutions. *Security & Privacy, IEEE*, 3(3):34–43, 2005.

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

[81] Gerhard P. Hancke. Practical Eavesdropping and Skimming Attacks on High-Frequency RFID Tokens. *Journal of Computer Security*, 19(2):259–288, 2011.

[82] Maximilian Engelhardt, Florian Pfeiffer, Klaus Finkenzeller, and Erwin Biebl. Extending ISO/IEC 14443 Type A Eavesdropping Range using Higher Harmonics. *ITG-Fachbericht-Smart SysTech 2013*, 2013.

[83] Infineon Technologies AG. my-d move - SLE66R01P. Product Brief. 2009.

[84] Corey Benninger and Max Sobell. NFC for Free Rides and Rooms (on your phone). Presentation at EUSecWest, 2012.

[85] Roel Verdult, Gerhard de Koning Gans, and Flavio D. Garcia. A Toolbox for RFID Protocol Analysis. In *RFID Technology (EURASIP RFID), 2012 Fourth International EURASIP Workshop on*, pages 27–34. IEEE, 2012.

[86] Zafar Anjum for Computer World Singapore. Infineon to supply Next Generation Security Chips for Singapore's CEPAS cards. URL: `http://www.computerworld.com.sg/resource/industries/infineon-to-supply-next-generation-security-chips-for-singapores-cepas-cards/`. Article, October 2013. Accessed on Dec 18 2013.

[87] Transit Link Pte Ltd. Use for Stored Value Cards FAQs. URL: `http://www.transitlink.com.sg/PSdetail.aspx?ty=art&Id=39`. Article. Accessed on Dec 07 2013.

[88] EZ-Link Pte Ltd. ez-Link Card Introduction. URL: `http://www.ezlink.com.sg/ez-link-card/`. Article. Accessed on Dec 07 2013.

[89] EMVCo LLC. EMV Integrated Circuit Card Specifications for Payment Systems: Book 1. Application Independent ICC to Terminal Interface Requirements. v4.2., 2008.

[90] Watchdata. CEPAS 2.0 ez-link Transport Card. URL: `http://www.watchdata.com/transportation/10148.html`. Article. Accessed on Jan 01 2014.

[91] Watchdata. CEPAS-compliant card. URL: `http://www.watchdata.com/transportation/10112.html`. Article. Accessed on Feb 25 2014.

[92] Christopher Tarnovsky. Deconstructing a 'Secure' Processor. Presentation at Black Hat DC, 2010.

[93] SPRING Singapore. SS518: Specification for Contactless e-purse application., 2006.

[94] Artem Chakirov and Alexander Erath. *Use of Public Transport Smart Card Fare Payment Data for Travel Behaviour Analysis in Singapore.* Eidgenössische Technische Hochschule Zürich, IVT-Institut für Verkehrsplanung und Transportsysteme, 2011.

[95] Artem Chakirov and Alexander Erath. *Activity Identification and Primary Location Modelling based on Smart Card Payment Data for Public Transport.* Eidgenössische Technische Hochschule Zürich, IVT, Institute for Transport Planning and Systems, 2012.

[96] Land Transport Authority of Singapore. Household Interview Travel Survey 2012: Public Transport Mode Share Rises to 63%. URL: `http://app.lta.gov.sg/apps/news/page.aspx?c=2&id=1b6b1e1e-f727-43bb-8688-f589056ad1c4`. Press Release, October 2013. Accessed on Jan 25 2014.

[97] Land Transport Authority of Singapore. Land Transport Master Plan 2013. URL: `http://www.lta.gov.sg/content/dam/ltaweb/corp/PublicationsResearch/files/ReportNewsletter/LTMP2013Report.pdf`. Report. Accessed on Jan 25 2014.

[98] SgWiki. Main Page. Welcome to sgWiki. URL: `http://sgwiki.com/wiki/Main_Page`. Article. Accessed on Jan 23 2014.

[99] Public Transport Council. Distance Fares (Current). URL: `http://www.ptc.gov.sg/FactsAndFigures/fares.htm`. Article. Accessed on Mar 06 2014.

[100] Transit Link Pte Ltd. Bus Enquiry - 927. URL: `http://www.transitlink.com.sg/eservice/eguide/service_route.php?service=927`. Article. Accessed on Jan 22 2014.

[101] OpenStreetMap. Copyright Notice ODbL and CC BY-SA. URL: `http://www.openstreetmap.org/copyright`. Accessed on Feb 03 2014.

[102] Transit Link Pte Ltd. Bus Enquiry - 179. URL: `http://www.transitlink.com.sg/eservice/eguide/service_route.php?service=179`. Article. Accessed on Jan 22 2014.

[103] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. User Privacy in Transport Systems Based on RFID E-Tickets. *PiLBA08 Privacy in Location-Based Applications*, page 102, 2008.

[104] Der-Horng Lee, Lijun Sun, and Alex Erath. Study of Bus Service Reliability in Singapore Using Fare Card Data. In *12th Asia-Pacific Intelligent Transportation Forum*, 2012.

[105] Marie-Pier Pelletier, Martin Trépanier, and Catherine Morency. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4):557–568, 2011.

[106] Timo Kasper, David Oswald, and Christof Paar. Side-Channel Analysis of Cryptographic RFIDs with Analog Demodulation. In *RFID. Security and Privacy*, pages 61–77. Springer, 2012.

[107] Riscure BV. Inspector 4.6 User Manual. March, 2013.

[108] Thomas Eisenbarth, Sandeep Kumar, Christof Paar, Axel Poschmann, and Leif Uhsadel. A Survey of Lightweight-Cryptography Implementations. *IEEE Design & Test of Computers*, 24(6):0522–533, 2007.