



Norwegian University of
Science and Technology

Securing Near Field Communication

Henning Siitonen Kortvedt

Master of Science in Communication Technology

Submission date: June 2009

Supervisor: Stig Frode Mjølunes, ITEM

Problem Description

Near Field Communication (NFC) is a wireless (magnetic field induction) communication channel with a range around one decimeter. Communication operates on the ISM band of 13.56 MHz with data rates up to 424 kbps. Mobile handsets, smart cards and electronic identification chips are some of the device types proposed to be equipped with NFC and to be used with security sensitive applications, such as electronic ticketing, payments, identification, and access control. The radio frequency signal may be picked up several meters away.

The task will be to present the theory behind NFC and analyze if it's feasible to eavesdrop information sent from a passive NFC-Tag. Physical experiments will define the vulnerabilities of NFC when reading a tag. Possible security mechanisms that can deal with the found weaknesses will then be presented. The final result should be a proposal for the best possible security solution so that NFC can be used for transmission of sensitive data related to banking and payment applications.

Assignment given: 13. January 2009
Supervisor: Stig Frode Mjølunes, ITEM

ABSTRACT

Near Field Communication (NFC) specifies a standard for a wireless communication protocol enabling data transfer by keeping two devices close together, about 10 cm maximum. NFC is designed for integration with mobile phones, which can communicate with other NFC phones (peer-to-peer) or read information on tags and cards (reader). An NFC device can also be put in card emulation mode, to offer compatibility with other contactless smart card standards. This enables NFC devices to replace traditional contactless plastic cards used in public transport ticketing, access control, ATMs and other similar applications.

At the beginning of my work, there seemed to be no available security protocol for NFC. I therefore formed a hypothesis stating that NFC communication can be eavesdropped, with intention to present methods to secure the channel if needed. It should however turn out that ECMA has been working with a security protocol called NFC-SEC-01, which specifies a key agreement and secure channel protocol for NFC in peer-to-peer mode. My work does mainly focus on reader mode and card emulation mode, as I expect these to be the most interesting from a commercial point of view.

This Master's thesis proves that it is possible to eavesdrop on NFC communication, and gives a thorough description of how this can be done using simple equipment and methods. The performed experiments show that the communication protocol does not offer any security in itself, and that transferred data can be picked up at a distance of approximately 20-30cm using an improvised antenna without any signal amplification or filtering circuitry. Using a proper radio receiver, this distance should increase significantly. As NFC in active communication mode has a far greater eavesdropping range than the passive mode, the latter should be preferred in typical banking and payment applications.

A layered security model is presented to counterfeits the possible attacks against applications using NFC technology. This solution is meant to be an idea for a public NFC security framework. A publicly available security library would ease implementation of the desired security level when developing applications, and hopefully attract more applications to start using this technology instead of traditional plastic cards. This may save both cost and the environment, as plastic cards will be replaced by software implemented tokens.

By introducing a common security protocol, cryptographic co processors can be implemented while manufacturing the devices. This will be more effective than implementing special software encryption for each application. The cryptographic mechanisms presented in

my solution shall however work efficiently also for software implementations, which opens for a smooth transition period with coexisting applications using traditional contact based and contactless smart cards.

PREFACE

This Master's thesis is written by Henning S. Kortvedt and submitted to The Norwegian University of Science and Technology (NTNU), completing a two year MSc program in communication technology with specialization in information security. The thesis is carried out at the Department of Telematics, NTNU.

As the RF-signals dealt with in detail during this thesis are somewhat on the side of the subjects taught in the information security program, quite some effort has been put into antenna theory study, oscilloscope configuration and signal conditioning.

During the last weekend of report writing I discovered that ECMA International has published two new standards regarding NFC security, ECMA-385 and ECMA-386. They were sent to ISO/IEC as DISs (Draft International Standard) in March, and are currently undergoing a 5 months ballot period. As part of my assignment was to propose a security protocol for NFC if proved needed, this discovery was a personal set-back that late in the work process. It however turned out that the two approaches were complementing each other rather than overlapping, which was a big relief.

I would like to thank everyone who has helped and supported me during this thesis. Professor Stig Frode Mjølunes has been both academic responsible and project supervisor, and his feedback and support has been crucial to the success of this thesis. I would also like to give a special contribute to the following: the Telematics Workshop at the Department of Telematics for building the DUT positioning rack, the Instrument Service at Department of Telematics for lending me measurement instruments and finally Zoe Ko at Advanced Cards Systems Ltd. in Hong Kong for sending java library files for the example code provided in the ACR 122 SDK.

Trondheim, June 4th, 2009.

Henning Siitonen Kortvedt

TABLE OF CONTENTS

ABSTRACT.....	I
PREFACE.....	III
TABLE OF CONTENTS.....	V
FIGURE LIST.....	IX
TABLE LIST.....	XI
ABBREVIATIONS.....	XIII
TERMINOLOGY.....	XV
1. INTRODUCTION.....	1
1.1. BACKGROUND.....	1
1.2. PROBLEM.....	1
1.3. DELIMITATIONS.....	2
1.4. STRUCTURE.....	2
1.5. RELATED WORK.....	2
1.6. CONTRIBUTIONS.....	3
1.7. METHODOLOGY.....	3
2. THEORETICAL BACKGROUND.....	7
2.1. INTRODUCTION TO NFC.....	7
2.1.1. <i>General</i>	7
2.1.2. <i>RFID – historical overview</i>	9
2.2. NFC TECHNOLOGY.....	10
2.2.1. <i>General</i>	10
2.2.2. <i>RF field</i>	10
2.2.3. <i>RF signal interface</i>	10
2.2.4. <i>General protocol flow</i>	14
2.2.5. <i>Connection initiation</i>	14
2.2.6. <i>Initialization and Single device detection</i>	17
2.2.7. <i>Mode selection</i>	26
2.2.8. <i>NFC protocols</i>	27
2.2.9. <i>Antennas</i>	28
2.2.10. <i>Application examples</i>	29
2.3. RFID ANALYSIS.....	30
2.3.1. <i>General</i>	30
2.3.2. <i>Eavesdropping</i>	30
2.3.3. <i>Analyzer tools</i>	32
2.4. SECURITY ASPECTS.....	33
2.5. DEPLOYED SECURITY MECHANISMS.....	34
3. EAVESDROPPING EXPERIMENT.....	37

3.1.	EXPERIMENT OUTLINE	37
3.2.	TEST EQUIPMENT	38
3.2.1.	<i>General</i>	38
3.2.2.	<i>Improvised antenna</i>	38
3.3.	TEST SOFTWARE	42
3.4.	EAVESDROPPING TEST PROCEDURES	44
3.4.1.	<i>General</i>	44
3.4.2.	<i>Test plan</i>	44
3.4.3.	<i>ASK reading</i>	45
3.4.4.	<i>Load modulation reading</i>	45
3.4.5.	<i>Eavesdropping range set-up</i>	46
3.4.6.	<i>Oscilloscope configuration</i>	46
3.5.	TEST RESULTS.....	47
3.5.1.	<i>Signal verification</i>	47
3.5.2.	<i>Command recognition</i>	50
3.5.3.	<i>Communication sequence recognition</i>	54
3.5.4.	<i>Maximum eavesdropping ranges</i>	58
3.6.	TEST RESULT DISCUSSION	63
4.	SECURITY SOLUTION	65
4.1.	THREATS.....	65
4.1.1.	<i>General</i>	65
4.1.2.	<i>ASK</i>	65
4.1.3.	<i>Load modulation</i>	66
4.1.4.	<i>Tag content</i>	66
4.2.	COUNTERMEASURES	66
4.2.1.	<i>General</i>	66
4.2.2.	<i>Authentication solutions</i>	67
4.2.3.	<i>Encryption solutions</i>	68
4.2.4.	<i>Replay protection solutions</i>	68
4.2.5.	<i>Message integrity solutions</i>	69
4.3.	PROPOSED COMMON SECURITY FRAMEWORK FOR NFC	69
4.3.1.	<i>Introduction</i>	69
4.3.2.	<i>General model</i>	70
4.3.3.	<i>Security level 0</i>	71
4.3.4.	<i>Security level 1</i>	72
4.3.5.	<i>Security level 2</i>	73
4.3.6.	<i>Security level 3</i>	74
4.3.7.	<i>Key management</i>	75
4.3.8.	<i>Random number generation</i>	75
4.3.9.	<i>Key hierarchy</i>	76
5.	CONCLUSION AND FUTURE WORK.....	77
5.1.	FUTURE WORK	78

REFERENCES.....	79
APPENDIX A: TEST DOCUMENTATION	83
A1. TEST EQUIPMENT SPECIFICATIONS	83
A1.1. ACR 122U	83
A1.2. NOKIA 6212 Classic	84
A1.3. Tags and cards	85
A2. TEST RESULT DOCUMENTATION	87
A2.1. Command recognition	88
A2.2. Communication sequence recognition	89
A2.3. Eavesdropping reading range test	92
APPENDIX B: TEST-SOFTWARE DOCUMENTATION	97
B1. NFC TEST PROGRAM	97
B1.1. User manual	97
B1.2. Source code StartTest.java	99
B2. CRC CALCULATION PROGRAM.....	107
B2.1. User manual	107
B2.2. Source code for CRCCalculationProgram.c	108
APPENDIX C: PROJECT PLAN	111
C1. INTRODUCTION	111
C1.1. Background	111
C1.1. Project goal.....	111
C1.2. Limitations	111
C2. SCOPE AND DELIMITATIONS	112
C3. PROJECT ORGANIZING	113
C3.1. Project process.....	113
C3.2. Project management.....	113
C3.3. Other roles	113
C4. DECISION STAGES, FOLLOW-OP AND MILESTONES	114
C4.1. Decision stages.....	114
C4.2. Follow-up	114
C4.3. Milestones	114
C5. CARRYING OUT.....	115
C5.1. Main activities.....	115
C5.2. Time and resource schedule.....	115
C5.3. Tools.....	115

FIGURE LIST

FIGURE 1: ASSORTED NFC EQUIPMENT INCLUDING LABELS, CARD AND MOBILE PHONE.	8
FIGURE 2: PULSE SHAPE OF 100% ASK [1].	11
FIGURE 3: BIT REPRESENTATION FOR MANCHESTER ENCODING WITH OBVERSE AMPLITUDE [1].	13
FIGURE 4: WAVEFORM OF 10% ASK MODULATION [1].	13
FIGURE 5: FLOW CHART FOR NFC INITIALIZATION AND SDD [1].	15
FIGURE 6: TIME FRAME FOR INITIAL RF COLLISION AVOIDANCE [1].	16
FIGURE 7: RESPONSE RF COLLISION AVOIDANCE SEQUENCE DURING ACTIVATION [1].	16
FIGURE 8: GENERAL FRAME FORMAT FOR NFC COMMUNICATION [1].	17
FIGURE 9: FORMAT OF THE “SHORT FRAME” [1].	18
FIGURE 10: FORMAT OF THE “STANDARD FRAME” [1].	18
FIGURE 11: FLOWCHART FOR NFC INITIALIZATION AND SDD, AT THE INITIATOR SIDE [1].	24
FIGURE 12: USAGE OF CASCADE LEVELS [1].	25
FIGURE 13: CODING OF SLP_REQ COMMAND.	25
FIGURE 14: MODE SELECTION PROCEDURE FOR AN NFCIP-2 DEVICE.	26
FIGURE 15: COMMUNICATION CHANNELS USED IN HANCKE’S EXPERIMENT.	31
FIGURE 16: THREE-PASS AUTHENTICATION PROCEDURE [24].	35
FIGURE 17: TEST SET-UP ILLUSTRATION.	37
FIGURE 18: READING RANGE TEST SET-UP.	39
FIGURE 19: LABEL CIRCUITRY DESIGN. (FROM THE LEFT: NOKIA, MIFARE 1K, MIFARE ULTRALIGHT).....	40
FIGURE 20: CLOSE-UP PICTURE OF A MIFARE ULTRALIGHT LABEL WITH MARKING OF THE IC.....	40
FIGURE 21: MODIFIED LABEL WITH MARKING OF CONNECTION POINTS	41
FIGURE 22: MODIFIED LABEL WITH MEASUREMENT PROBE CONNECTED.	41
FIGURE 23: PROBE CONNECTED TO THE ANTENNA OF A MIFARE CLASSIC CARD.	42
FIGURE 24: USER INTERFACE OF THE DEVELOPED “NFC TEST PROGRAM”.	43
FIGURE 25: “CRC CALCULATION PROGRAM” USED TO COMPUTE CRC OF “00000000 00000000”.....	44
FIGURE 26: A 100% ASK PULSE THAT OCCURS IN THE BEGINNING OF A BIT DURATION.	45
FIGURE 27: EXAMPLE OF AN ASK MODULATION SEQUENCE.	48
FIGURE 28: BIT INTERPRETATION OF A 100% ASK SEQUENCE.	48
FIGURE 29: EXAMPLE OF A LOAD MODULATION SEQUENCE.	49
FIGURE 30: BIT INTERPRETATION OF A LOAD MODULATION SEQUENCE.	49
FIGURE 31: LOAD MODULATION FROM NOKIA 6212 CLASSIC.	50
FIGURE 32: CAPTURE OF AN ALL_REQ COMMAND.	51
FIGURE 33: INTERPRETED BITS OF AN ALL_REQ COMMAND.	52
FIGURE 34: START OF SENS_RES FROM TARGET.	53
FIGURE 35: BIT INTERPRETATION OF THE FIRST 5 BIT DURATIONS OF SENS_RES.....	53
FIGURE 36: START OF ALL_REQ.....	55
FIGURE 37: START OF SENS_RES.....	55
FIGURE 38: START OF SEL_REQ.....	56

FIGURE 39: START OF SEL_RES	56
FIGURE 40: CRC CALCULATION OF THE SEL_RES COMMAND.....	57
FIGURE 41: ANTENNA ADJUSTMENT RACK.....	58
FIGURE 42: ANTENNA POSITIONS IN MAXIMUM EAVESDROPPING RANGE TEST.....	59
FIGURE 43: EAVESDROPPING DIAGRAM FOR HORIZONTAL ANTENNA IN THE X-Y PLANE WITH Z=0.	60
FIGURE 44: EAVESDROPPING DIAGRAM FOR HORIZONTAL ANTENNA IN THE X-Z PLANE.	60
FIGURE 45: EAVESDROPPING DIAGRAM FOR HORIZONTAL ANTENNA IN THE Y-Z PLANE.	61
FIGURE 46: EAVESDROPPING DIAGRAM FOR PERPENDICULAR ANTENNA IN THE X-Y PLANE WITH Z=0.....	61
FIGURE 47: EAVESDROPPING DIAGRAM FOR PERPENDICULAR ANTENNA IN THE X-Z PLANE.....	62
FIGURE 48: EAVESDROPPING DIAGRAM FOR PERPENDICULAR ANTENNA IN THE Y-Z PLANE.....	62
FIGURE 49: MSC EXAMPLE ILLUSTRATION.....	70
FIGURE 50: SECURITY LEVELS IN THE SECURITY FRAMEWORK.....	70
FIGURE 51: MESSAGE EXCHANGE IN A TAG AUTHENTICATION PROCEDURE.....	71
FIGURE 52: MESSAGE EXCHANGE IN A READER AUTHENTICATION PROCEDURE.....	72
FIGURE 53: MESSAGE EXCHANGE IN A MUTUAL AUTHENTICATION PROCEDURE.....	73
FIGURE 54: MIC COMPUTATION.....	74
FIGURE 55: PAYLOAD ENCRYPTION PROCEDURE.....	74

TABLE LIST

TABLE 1: TYPICAL STEPS IN A SCIENTIFIC WORK PROCEDURE [22]	4
TABLE 2: DEFINITION OF THE DEVISOR D IN THE FORMULA FOR BD-CALCULATION [1].	11
TABLE 3: DEFINITION OF TIME INTERVALS IN FIGURE 2 [1].....	12
TABLE 4: DEFINITION OF TIME INTERVALS IN FIGURE 4 [1].....	13
TABLE 5: DEFINITION OF FRAME RESPONSE TIME FROM FOR THE TARGET WHEN RECEIVING COMMANDS [1].	18
TABLE 6: POSSIBLE STATES OF OPERATION FOR AN NFCIP-1 DEVICE.....	19
TABLE 7: COMMAND SET FOR NFC INITIALIZATION PROCEDURE [1].	20
TABLE 8: CODING OF COMMANDS USED IN SHORT FRAMES [1].	20
TABLE 9: CODING OF SENS_RES [1].	20
TABLE 10: CODING OF “NFCID1 SIZE BIT FRAME” BITS IN SENS_RES [1].	21
TABLE 11: CODING OF “BIT FRAME SDD” BITS IN SENS_REQ [1].	21
TABLE 12: CODING OF SDD_REQ AND SEL_REQ [1].....	21
TABLE 13: CODING OF SEL_CMD [1].	22
TABLE 14: CODING OF THE 4 “UPPER” BITS OF SEL_PAR [1].	22
TABLE 15: CODING OF THE 4 “LOWER” BITS OF SEL_PAR [1]	22
TABLE 16: CODING OF SEL_RES [1].	23
TABLE 17: CASCADE LEVELS OF AN NFCID1.	25
TABLE 18: EAVESDROPPING RESULTS OF HANCKE’S TEST [15].....	32
TABLE 19: READING RANGE TEST RESULTS FOR ALL AVAILABLE TAG TYPES.....	39
TABLE 20: ALL TRANSMITTED BITS OF A SENS_RES FROM TARGET.	54
TABLE 21: SENS_RES COMMAND	54
TABLE 22: RESULTS OF CAPTURING A WHOLE COMMUNICATION SEQUENCE FROM INITIATOR TO TARGET	57
TABLE 23: XOR OF THE LAST FOUR DATA BYTES COMPARED TO THE INTERPRETED BCC IN A SEL_REQ.	57
TABLE 24: COMPARISON OF INTERPRETED AND CALCULATED CRC IN SEL_RES.	58
TABLE 25: EAVESDROPPING RANGES FOR THE AVAILABLE ANTENNAS.	58
TABLE 26: MAXIMUM EAVESDROPPING RANGE TEST RESULTS.	59

ABBREVIATIONS

ACS	Advanced Cards Systems Ltd.
AES	Advanced Encryption Standard
API	Application Programming Interface
ASK	Amplitude Shift Keying
ATM	Automatic Teller Machine
CBC	Cipher Block Chaining
CC	Common Criteria
CCIT	International Telegraph and Telephone Consultative Committee (now ITU-T)
CCMP	Counter Mode with Cipher Block Chaining Message Authentication Code Protocol
CRC	Cyclic Redundancy Check
CTR	Counter
DES	Data Encryption Standard
DIS	Draft International Standard
DUT	Device Under Test
EAL	Evaluation Assurance Level
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie Hellmann
ECMA	European Computer Manufacturers Association
ECRYPT	European Network of Excellence for Cryptology
EEPROM	Electrically Erasable Programmable Read-Only Memory
FRAM	Ferroelectric Random Access Memory
FRT	Frame Response Time
IC	Integrated Circuit
IEC	International Electrotechnical Commission
ISM	Industrial, Scientific and Medical (unlicensed frequency band)
ISO	International Organization for Standardization
LED	Light Emitting Diode
lsb	Least significant bit
MAC	Message Authentication Code
MIC	Message Integrity Code
MIME	Multipurpose Internet Mail Extensions
MMS	Multimedia Messaging Service
msb	Most significant bit
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
NFCID	Near Field Communication Identifier
NFCIP	Near Field Communication Interface and Protocol
NIST	National Institute of Standards and Technology
NTNU	The Norwegian University of Science and Technology
NXP	Next eXPerience Semiconductors
OS	Operating System
OTA	Over The Air programming
PCB	Printed Circuit Board
PCD	Proximity Close-coupling Device
PICC	Proximity Integrated Circuit Card
PRNG	Pseudo Random Number Generator
PSK	Pre Shared Key
RF	Radio Frequency
RFID	Radio Frequency Identification
RNG	Random Number Generator
RSA	Rivest Shamir Adleman encryption algorithm
SDD	Single Device Detection
SDK	System Development Kit
SIM	Subscriber Identity Module
SMS	Short Messaging Service
SRAM	Static Random Access Memory
UICC	Universal Integrated Circuit Card

UID	Unique Item Identifier
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VCD	Vicinity Close-coupling device
W-LAN	Wireless Local Area Network

TERMINOLOGY

Card:	A Card is a passive close coupling device with ID-1 format, i.e. typical credit card size. A Card can be seen as a subgroup of a Target.
Close coupling:	This means communication that is enabled when two devices are kept close to each other.
ID:	A parameter identifying a unique device, typically a number or a bit value with a defined length.
Initiator:	An Initiator is the term used in the ISO-standards to describe a close coupling device that takes initiative to start any close coupling communication initiation sequence.
RAND:	A random number generated for cryptographic purposes.
Reader:	A reader is an active device that powers up and initiates contact with a passive close coupling device. A Raeder can be seen as a subgroup of an Initiator.
RES:	The result of a cryptographic computation.
Tag:	A Tag is a passive close coupling device without any specified physical layout. A tag can be seen as a subgroup of a Target.
Target:	This is the term used in the ISO-standards to indicate the responding device of any close coupling communication initiation sequence.

1. INTRODUCTION

1.1. Background

The task is based on a Master's thesis suggestion from Prof. Stig Frode Mjølunes at NTNU, Department of Telematics. NFC is an upcoming technology, based on already deployed RFID techniques. NFC may be heavily deployed in a relatively short time to come, as NFC-applications are already undergoing large scale tests in Europe, North America, Asia and Oceania. Test examples are public transport payment, credit cards, electronic tickets, advertisement and W-LAN set up. The development of NFC is driven by the NFC Forum, an organization of 150 companies working together to promote NFC.

NFC enabled mobile phones are expected to eliminate consumer's need for numerous cards, badges and tickets, and become a complete electronic wallet. This may also cause great savings for issuers of all types of plastic cards and ID tokens, as they can be replaced by imaginary cards realized in software applications. The number of potential users may grow extremely fast if this technology becomes a standard feature on all mobile phones, as many people tend to switch phone at least every other year. A technology with many users and valuable transactions involved is however very attractive to people with dishonest intentions. This thesis will focus on the radio interface properties of NFC.

1.2. Problem

Because of the relatively short communication range in NFC and RFID in general, little effort has traditionally been put into security analysis of such protocols. It seems that the short signal range leads people into thinking that the channel cannot be tampered with. This is also indicated at the official NFC Forum home page under the tab "About NFC", where they state that "Because the transmission range is so short, NFC-enabled transactions are inherently secure" [3]. Theoretically it's feasible to pick up information sent on the channel, disturb it, alter information and replay previous sent messages, as long as no security protocol is introduced. The work in this thesis will seek to prove the possibility of eavesdropping transferred data in an NFC communication sequence. I will also propose solutions for a security protocol, so that NFC can be used to reduce peoples need for physical VISA/MASTER-cards and tickets in a secure manner. The goal is to offer a better total security than credit cards, payment cards and access control cards do today.

1.3. Delimitations

I have chosen to focus on the communication between an active reader and a passive tag. This is expected to be the most robust communication form in NFC, as the information sent from a passive tag should be the hardest to pick up by eavesdropping. I also believe that the passive tag model will be the most heavily deployed solution in payment and banking systems, as it is compatible with previous standards and can offer a smooth transition period.

A problem for all types of radio communication is radio jamming, for instance intentional interference by sending noise on certain frequencies or frequency bands. This threat will always exist in wireless communication, and will not be further discussed in this thesis.

1.4. Structure

The thesis starts with a theoretical presentation of NFC and related technology, together with relevant security mechanisms. This is followed by a description of performed tests on the passive tag communication channel, and then the test results are presented. It proceeds with a proposal for the best possible security solution, based on the theoretical background and the test results. The thesis is summed up with a discussion of the findings, a conclusion of my work and suggestions for future proceedings. In Appendix A, you will find detailed documentation of the test results and technical information about employed equipment that is omitted in the report. Appendix B provides user manuals for the test software together with source code for programs developed during this thesis. The programs I made are also attached as executable files (.jar and .exe). Appendix C is a project plan for this thesis.

1.5. Related work

At the starting point of this thesis, not much work regarding security of NFC was available. RFID has however gone through a lot of studies, which to some extent also applies to NFC, as the radio interfaces of the different standards are similar. Chapter 2.3, 2.4 and 2.5 describe relevant work related to RFID which is also may be applicable to NFC.

ECMA International has recently published two standards that deals with the security of NFC, namely “NFCIP-1 Security services and Protocol” (ECMA-385) [30] and “NFC-SEC Cryptography Standard using ECDH and AES” (ECMA-386) [31]. They are

currently undergoing an approval procedure by ISO/IEC, and the voting should be finished in late august 2009 [32] [33] [34]. The drafts have been numbered ISO/IEC DIS 13157 and ISO/IEC DIS 13158. These standards present a security protocol and appurtenant security mechanisms for NFC peer to peer communication. They do however not consider reader and card emulation mode, which are dealt with in this thesis.

1.6. Contributions

Chapter 3 presents an experiment planned and performed by me while working with this thesis. In this section I show how to eavesdrop and demodulate an NFC communication initiation sequence, and at which ranges this can be done using an oscilloscope and an improvised antenna. The work included developing software for reader configuration, so that it can perform a stable test sequence, and also a small program for CRC calculation.

As there was no existing common security framework for NFC at the time this thesis was started, Chapter 4 presents a layered security solution proposal. This section describes my suggestion on how a public security framework for NFC should look like. The goal was to help this wireless technology becoming widely deployable and highly compatible, and still offer sufficient security for multiple applications using the same platform.

The work in this task will also result in an article, which is going to be sent to The Norwegian Information Security Conference (NISK 09) for acceptance evaluation.

1.7. Methodology

The experimental work in this thesis is performed in a scientific manner [22], while the security solution is formed using a theoretical approach. A scientific method is recognized by formulating and testing a hypothesis, and collection of data through observation and experimentation. The data shall be empirical, observable and measurable in order to fulfill the requirements of a scientific approach. The data collection is often performed using scientific instruments such as oscilloscopes, logic analyzers or spectrum analyzers. An example of a step by step scientific approach is shown in Table 1 [22].

1	Define the question
2	Gather information and resources (observe)
3	Form hypothesis
4	Perform experiment and collect data
5	Analyze data
6	Interpret data and draw conclusions that serve as a starting point for new hypothesis
7	Publish results
8	Retest (frequently done by other scientists)

Table 1: Typical steps in a scientific work procedure [22]

A scientific process is an iterative process which means that the findings can make you go back and repeat earlier tasks at any stage of the work. This can lead into rethinking both the hypothesis and/or the experimental method in use.

It is important that the results are reproducible, as the confirmation by other researchers is crucial to get the scientific community's approval of the results. If the work is done systematically any other scientist can start his own work at any stage in the process. Another important element to consider when performing experiments is the uncertainty. It is common to include an estimate of the accuracy related to the measurement method, equipment and test environment.

The experimental work of my thesis has been planned, carried out and documented in such a manner that it fulfills the requirements of a scientific approach. After forming my hypothesis I started with information gathering, resulting in a chapter presenting all theoretical background needed in order to understand and carry out the experiments. Then I started working with the oscilloscope and available NFC equipment to reach the goal of the measurements. In the beginning I spent some time of trying and failing to get familiar with the test equipment, in addition to writing some necessary software to make the test equipment work as intended. All tests were completed once using a MIFARE Label as antenna, but were redone as I managed to make a better antenna of a MIFARE Classic card. The second round of testing was far more effective and easier to document as the approach was well known. As all the planned tests were successful, there was no need for rethinking the hypothesis. To make sure that my work is reproducible and comparable, I

have described all approaches and presented all test results together with documentation of all equipment which has been in use.

As the experimental work of this thesis is not very closely related to the subjects taught in the “Communication Technology”-program at NTNU, a comprehensive theoretical study was necessary in the information gathering phase.

2. THEORETICAL BACKGROUND

This chapter will present a theoretical background for the NFC technology, some possible applications, NFC's development status of today and examples of available equipment. Then the security aspects of this technology will be discussed followed by a description of relevant security mechanisms used in the proposed security solution presented in Chapter 4.

2.1. Introduction to NFC

2.1.1. General

NFC is a short range communication protocol with moderate bit rates developed mainly for use with mobile phones. The technical solutions are derived from RFID technology. This initiative is a result of cooperation between mobile operators, mobile manufacturers and the electronics industry. Companies like Nokia, Sony, NXP Semiconductors, VISA, MasterCard, Microsoft, Telenor and Vodafone are present. NFC is standardized in ISO/IEC 18092 [1] and ISO/IEC 21481 [6], corresponding to ECMA-340 [4] and ECMA-352 [7]. For easier integration, NFC has been derived from the same platform as ISO/IEC 14443 [2], or "proximity cards". This contactless card protocol is widely used in access control and public transport payment cards. NFC-tags shall be compliant with ISO/IEC 14443. NFC is also compatible with FeliCa (Japanese Industrial Standard X 6319-4), and to some extent ISO/IEC 15693 (vicinity cards). This means that a NFC enabled phone should be able to transmit a unique ID to for instance an access control system using vicinity technology. NFC has published 4 different "Tag Operation" specifications, which has different level of compatibility with existing standards. These features enable application developers to make solutions for both customers with NFC phones and customers with traditional contactless cards.

The basic communication model is an NFC-reader which is reading a tag, but is also possible to use NFC for mutual data exchange. NFC enabled phones can act as both reader and tag, and can operate in both active and passive communication mode. In active mode, two active units communicate with each other. In passive mode a tag is put into operation by a reader by using the reader's RF-field to generate power and to respond. In the active mode, two active NFC devices are communicating by alternately generating an RF-field.

The communication technology is based on magnetic field induction from an active device. This means that the passive device is powered up when the magnetic field is strong enough to induce the needed voltage in the receiver's antenna so that its internal circuitry can operate. When activated, the tag responds to commands sent by the reader. Possible actions can be to authenticate, or to transmit the content stored in its memory. The data stored on the tag can be pictures, device commands, text, videos, URLs or phone numbers. The data on the tag can be the content of interest, or it can be used to redirect a reader device to the actual wanted service. An example can be a tag outside a grocery store containing an URL that is redirecting the reader to a webpage showing the special offers of today.

NFC enabled equipment is manufactured in many different configurations, in addition to mobile phone implementation. Some are shown in Figure 1.



Figure 1: Assorted NFC equipment including labels, card and mobile phone.

The three smallest devices in Figure 1 are implemented as flexible stickers that can be put almost anywhere. The thickness is below half a millimeter. In addition a traditional credit card size contactless card and an NFC enabled phone are shown.

2.1.2. RFID – historical overview

RFID is produced in countless variants. To classify them they are divided into Full Duplex (FDX), Half Duplex (HDX) and Sequential (SEQ) systems [11]. In HDX and FDX the transponders response is broadcasted whenever a reader's magnetic field is present. To differentiate the strong reader signals from the much weaker transponder signals, load modulation from the transponder is used. In sequential systems, the reader's magnetic field is periodically switched off in order for the transponder to send its data. To accomplish this, the transponder uses capacitors or batteries to even out the power loss within the periods without magnetic field induction.

The data storage in RFID transponders varies between a few bytes and several kilobytes, dependent of the user application. A much used version is also the 1-bit transponder, which only can indicate whether a transponder is present in the field or not. This can for instance be used in anti theft systems to protect clothes and other goods in shops.

Transponders may be preprogrammed from the manufacturer, or the content can be changed by writing data from a reader. Writeable transponders contain EEPROMs, FRAMs or SRAMs. To control the read and write operations, state machines or microprocessors are used. RFID systems can also be divided by reading ranges, close-coupling (0-1cm), remote-coupling (0-1m) and long-range (>1m). Transponders use backscatter modulation, load modulation or sub harmonic frequencies to send its response to the reader.

To optimize the performance of RFID systems, a lot of functionalities have been provided in addition to the original serial number identification. EEPROM and SRAM make it possible to read and write larger amounts of data to the transponder (16 bytes to 8 Kbytes and 256 bytes to 64 Kbytes respectively). To prevent units from interfering with each other, anti-collision procedures are developed. To provide more secure communication, authentication and encryption systems have been launched. The most advanced versions have microprocessors with a smart card operating system to perform more complex operations, such as heavier encryption algorithms. Today dual interface smart cards are the top of the line, with a cryptographic coprocessor to reduce the computing time significantly. These types of cards are mostly manufactured to be compliant with the ISO/IEC 14443 standard, using an operating frequency of 13,56MHz.

2.2. NFC Technology

Sections 2.2.2 through 2.2.6 are derivatives of ISO/IEC 18092 [1], limited to what is needed to understand the work presented in this thesis. In order not to disrupt the original meaning of the standardization text, some of the parts in these sections are direct transfers from the standard. As this task focuses on the passive communication mode, detailed description of the active communication mode is omitted. ECMA-340 [4] is complementary to ISO/IEC 18092, and gives the same knowledge.

2.2.1. General

The NFC standard defines two modes of operation, active and passive. In passive mode the initiator generates a RF field to energize the target. In turn the target responds using a load modulation scheme on the field generated by the initiator. In active mode both devices generate their own RF field and modulation. The initiator sends information or commands on its field, and the responder answers on another field. All generated fields have to stay within the field strength limits defined in Chapter 2.2.2. The communication mode cannot be changed during a communication sequence.

2.2.2. RF field

NFC operates in the 13,56MHz ISM band. ISM bands (Industrial, Scientific and Medical) are unlicensed frequency bands open for all manufacturers of radio equipment as long as they stay within certain limits such as emitted power. This means that interference from other equipment always must be taken into consideration, as you cannot control the environment. The carrier frequency (f_c) is 13,56MHz and the bandwidth of the channel is around 1MHz. The modulation and coding techniques decide the data transfer speed. An active device is continuously generating a field to be exploited by a tag. The unmodulated field has to stay between 1,5A/m rms (H_{\min}) and 7,5A/m rms (H_{\max}), and will be modulated during communication. All NFCIP-1 devices shall also detect any external fields higher than 0,1875A/m ($H_{\text{Threshold}}$) while performing an external RF field detection.

2.2.3. RF signal interface

The bit duration (bd) in NFC is dependent on the communication mode and the data rate, and can be calculated by the following formula [1]:

$$bd = 128/(D \times f_c)$$

D is defined in Table 2.

Communication Mode	kbps	Divisor D
active or passive	106	1
active or passive	212	2
active or passive	424	4
Active	847	8
Active	1 695	16
Active	3 390	32
Active	6 780	64

Table 2: Definition of the divisor D in the formula for bd-calculation [1].

In active communication mode, the specification shall always be the same for both initiator to target and target to initiator communication. At the lowest data transfer speed supported by NFC, the initial bit rate shall be 106 kbps ($f_c/128$), and is set by the initiator. For this bit rate, the initiator shall use 100% ASK modulation to generate pulses as shown in Figure 2.

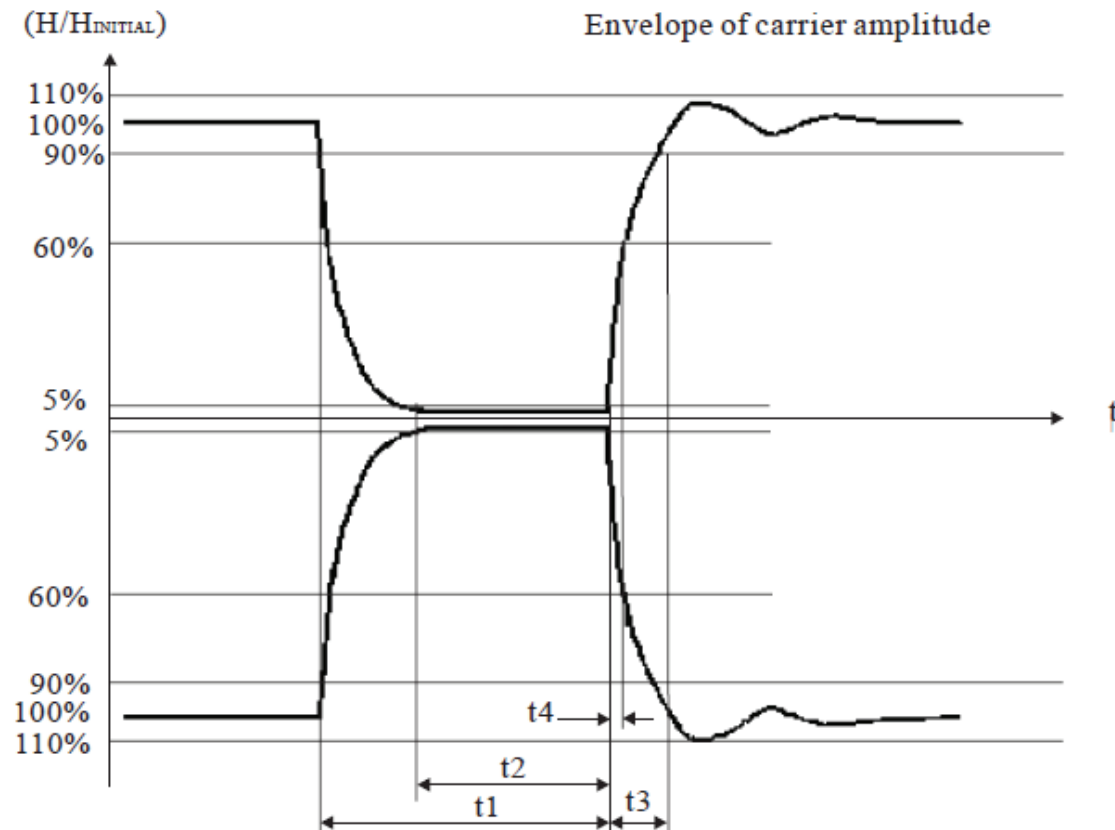


Figure 2: Pulse shape of 100% ASK [1].

Pulses length (Condition)	t1 [μs]	t2 [μs]		t3 [μs]	t4 [μs]
		(t1 ≤ 2,5)	(t1 > 2,5)		
Maximum	3,0	t1		1,5	0,4
Minimum	2,0	0,7	0,5	0,0	0,0

Table 3: Definition of time intervals in Figure 2 [1].

The time intervals shown in Figure 2 are defined in Table 3. These values set the limits for the envelope of the signal, and t₄ is used to detect the “end of pulse”. The byte encoding shall be least significant bit (lsb) first. To represent bits, the following coding shall be used [1]:

- Start of communication: at the beginning of the bit duration a “Pulse” shall occur.
- ONE: after a time of half the bit duration a “Pulse” shall occur.
- ZERO: For the full bit duration no modulation shall occur with the following to exceptions:
 - If there are two or more contiguous ZEROs, from the second ZERO on a “Pulse” shall occur at the beginning of the bit duration.
 - If the first bit after a “start of communication” is ZERO, a “Pulse” shall occur at the beginning of the bit duration.
- End of Communication: ZERO followed by one bit duration without modulation.
- No information: shall be coded with at least two full bit durations without modulation

When the bit rate in passive communication is 106 kbps, the target shall respond via load modulation generating a subcarrier with frequency $f_s = f_c / 16$, where the amplitude has to exceed a minimum value relative to the strength of the present magnetic field. Bytes shall be encoded lsb first and the bit representation shall be performed by Manchester Coding with obverse amplitude as shown in Figure 3.

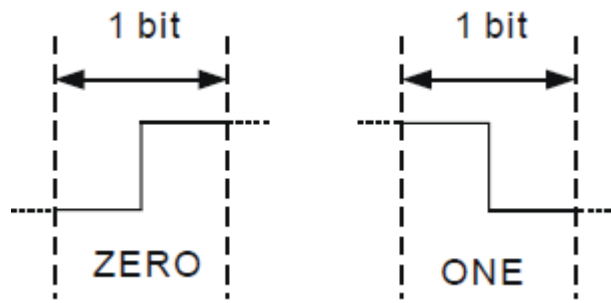


Figure 3: Bit representation for Manchester encoding with obverse amplitude [1].

If a higher bit rate is selected by the Initiator, another scheme is used. The bit rate for communication and single device detection shall be $f_c / 64$ (212 kbps) or $f_c / 32$ (424 kbps) respectively. ASK is still used, but with a modulation index of 8% to 30% (called 10% ASK) as described in Figure 4 and Table 4.

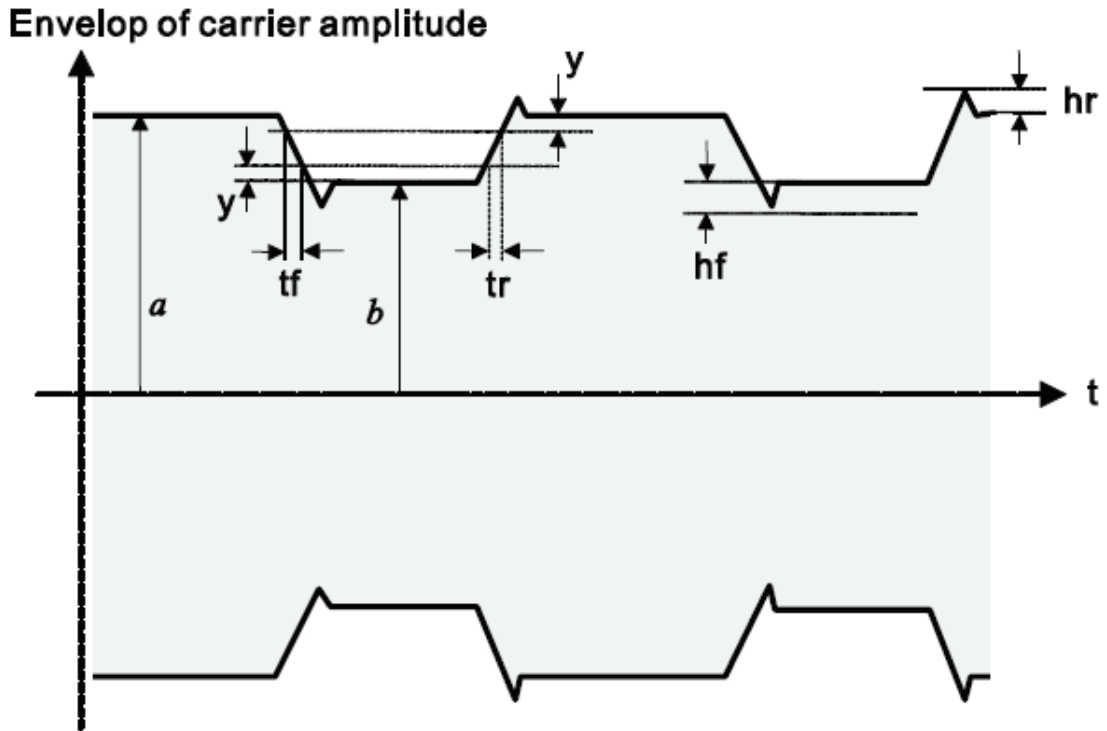


Figure 4: Waveform of 10% ASK modulation [1].

	212 kbps	424 kbps
tf	2,0 μ s max	1,0 μ s max
tr	2,0 μ s max	1,0 μ s max
y	0,1 (a - b)	0,1 (a - b)
hf, hr	0,1 (a - b) max	0,1 (a - b) max

Table 4: Definition of time intervals in Figure 4 [1].

The byte encoding shall be most significant bit (msb) first and the bit representation shall be Manchester Coding with obverse amplitude as shown in Figure 3. The target shall respond with the same load modulating scheme, but the bit duration of the Manchester coding must be changed so that it matches the bd related to the actual bit rate. The byte encoding shall be msb in this direction too.

2.2.4. General protocol flow

To ensure proper operation, an NFCIP-1 device has to follow a distinct set of sequential operations [1]:

- Any NFCIP-1 device shall per default be in target mode.
- When in target mode, it shall not generate an RF field, and shall wait silently for a command from the initiator.
- The NFCIP-1 device may switch to Initiator mode only if required by the application.
- The application shall determine either Active or Passive communication mode, and transfer speed.
- Initiator shall test for external RF field present and shall not activate its RF field if an external field is detected.
- If an external field is not detected, the Initiator shall activate its RF field.
- The target shall be activated by the RF field of the Initiator.
- Transmission of a command by the Initiator either in the Active communication mode or in the Passive communication mode at a selected transfer speed.
- Transmission of a response by the Target either in the Active communication mode or the Passive communication mode. The communication mode and the transfer speed shall be the same as the Initiator communication mode and the transfer speed.

2.2.5. Connection initiation

NFC has an initiation and collision detection protocol for both active and passive communication mode. An initiator shall detect occurring collisions when two or more targets are transmitting bit patterns with complementary bit position values. A sequence diagram for initialization and Single Device Detection is shown in Figure 5.

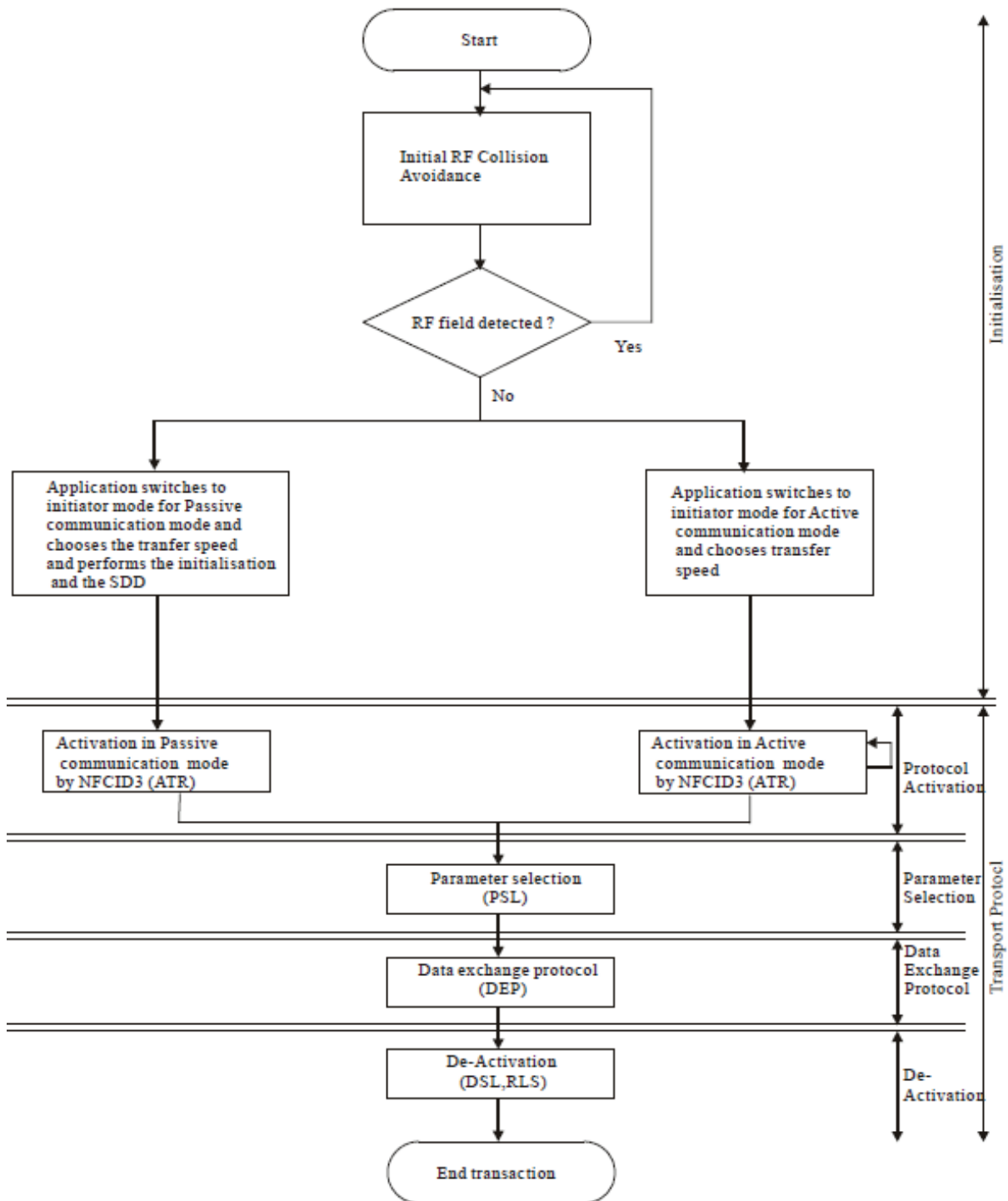


Figure 5: Flow chart for NFC initialization and SDD [1].

To prevent interference with ongoing NFC communication or other equipment using the same carrier frequency, an RF field shall not be generated as long as another RF field is detected. This means that an initiator continuously shall sense for presence of any external RF field. If no field is detected within a given timeframe, the initiator can switch its own field on. Figure 6 defines this timeframe ($T_{IDT} + n * T_{RFW}$).

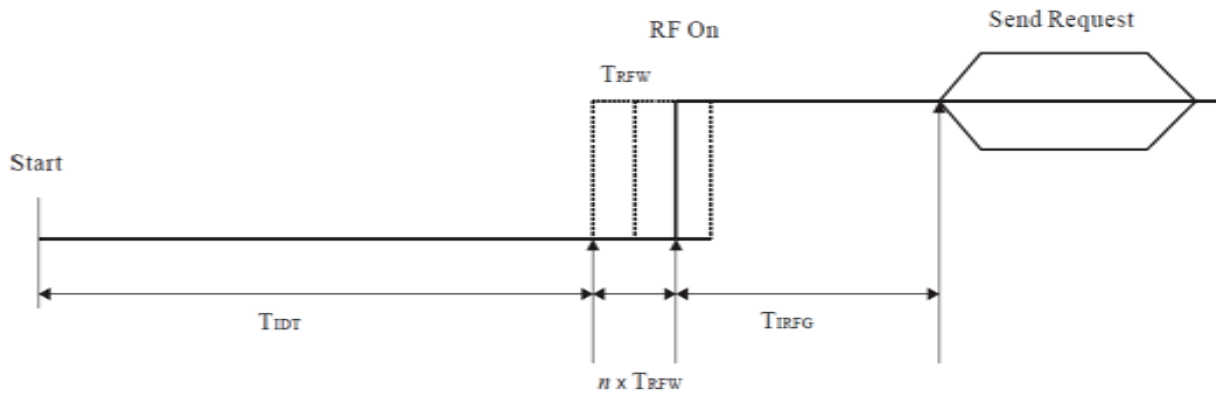


Figure 6: Time frame for initial RF collision avoidance [1].

T_{IDT} : Initial delay time

T_{RFW} : RF waiting time

n : Randomly generated number of time periods for T_{RFW}

$$0 \leq n \leq 3$$

T_{IRFG} : Initial guard time between switching on RF field and start to send command or data frame

$$T_{IRFG} > 5 \text{ ms}$$

When operating in passive communication mode, the RF field generated by the initiator cannot be switched off during a transaction. In active communication mode on the contrary, the Initiator turns of the RF field when it's waiting for response from the Target as it shall generate its own RF field for transmission. In active communication there is also a process called "Response RF Collision Avoidance", to avoid collision of data if more than one target is simultaneously responding. The sequence is shown in Figure 7.

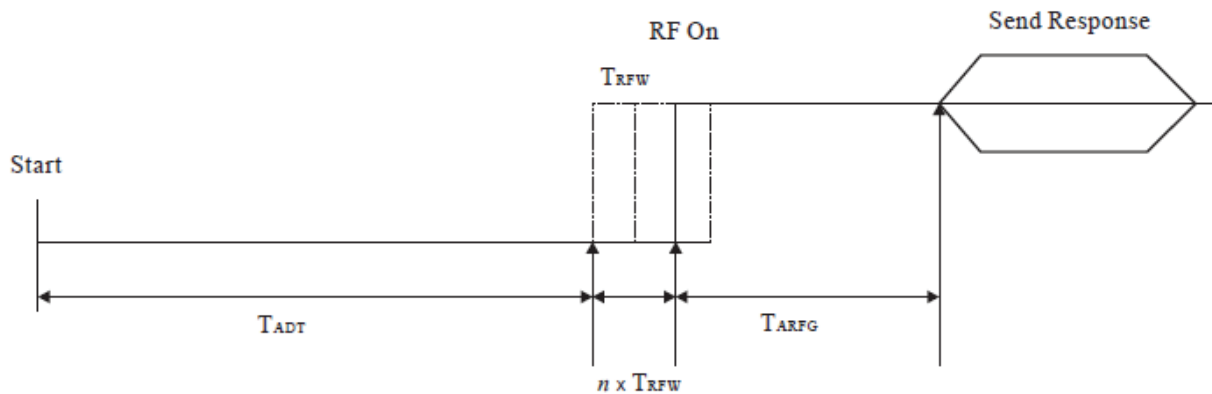


Figure 7: Response RF Collision Avoidance Sequence during activation [1].

T_{ADT} : Active delay time, sense time between RF off Initiator/Target and Target/Initiator

T_{RFW} : RF waiting time

n : Randomly generated number of time periods for T_{RFW}

$$0 \leq n \leq 3$$

T_{ARFG} : Active guard time between switching on RF field and start to send command

$$T_{ARFG} > 1024/f_c$$

2.2.6. Initialization and Single device detection

This procedure is defined separately for the two communication modes, and for passive mode it is also divided into one specification for 106 kbps and one specification for 212/424kbps. As passive communication mode at 106 kbps is the only one used in the experiment, this section is limited to the relevant part of ISO/IEC 18092 [1]. If further knowledge of the other modes is desired, information can be found in Chapter 11.2.2 and 11.3 of the standard.

The initialization and SDD procedure is defined by a set of frame formats and timing intervals, and is performed after a successful collision avoidance procedure. The data is always transmitted in pairs initiated by a request from the Initiator, followed by a target response. The frame format is shown in Figure 8.

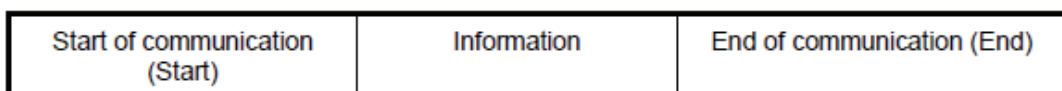


Figure 8: General frame format for NFC communication [1].

The Frame Response time (FRT) is a definition of the time a target has to wait from the last received pulse until it can transmit start of communication, and is defined in Table 5. Because the last part of an “End” from the Initiator to the target is defined by one unmodulated bit duration after the last “0”, the FRT and “End” will overlap. The length of the overlap depends on whether the last data bit is a “0” or a “1”, which defines the position of the last pulse.

Command type	n (integer value)	FRT	
		last pulsed bit = ONE	last pulsed bit = ZERO
SENS_REQ ALL_REQ SDD_REQ SEL_REQ	9	$(n \times 128 + 84) / fc$	$(n \times 128 + 20) / fc$
All other commands	≥ 9		

Table 5: Definition of frame response time from for the Target when receiving commands [1].

For the commands used in the SDD, “n” always equals 9. This is done to ensure that all devices respond synchronously during the SDD.

During communication, two types of frames are used as shown in Figure 9 and Figure 10.

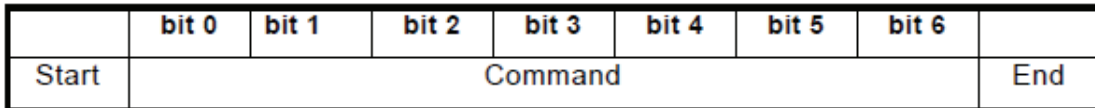


Figure 9: Format of the “short frame” [1].

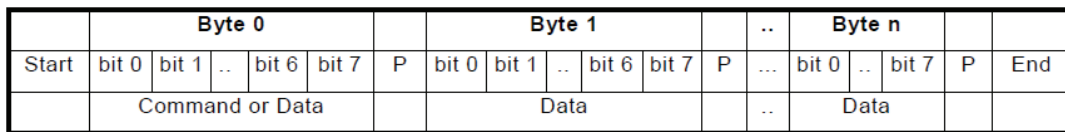


Figure 10: Format of the “standard frame” [1].

Both frames shall transmit lsb first. In the standard frame, the lsb transmission is done “per byte” and every byte shall be followed by an odd parity bit. A collision shall be detected when at least two targets transmit different bit patterns to a reader. This is recognized by a whole bit duration with the carrier modulated by the subcarrier because of the properties of Manchester Coding.

A target in operation can be put in a number of states which describe how it shall react to the certain actions, as describe in Table 6 [1].

POWER-OFF State	The tag is inactive because there is no field to energize it. If I enters a field above H_{min} , the tag shall enter its SENSE State
SENSE State	The tag is powered and listens for SENS_REQ and ALL_REQ. If one of the above is received, the tag shall transmit SENS_RES and enter Resolution State. If any other command is received, the tag shall stay in SENSE State
RESOLUTION State	Bit frame SDD may be applied. Cascade levels are handled to get the complete NFCID1. When receiving a valid SEL_REQ with its complete NFCID1, it enters the SELECTED State and sends SEL_RES. If any other commands are received, the tag shall fall back to SENSE State.
SELECTED State	Dependent on the coding of the SEL_RES, the tag shall listen to an ATR_REQ or a valid proprietary command. If a valid SLP_REQ is received it shall enter SLEEP State. DSL commands in the transport protocol are specified to return the tag to its SLEEP State. If any other commands are received it shall fall back to SENSE State
SLEEP State	In this state a tag shall only respond to an ALL_REQ. If a valid ALL_REQ is received, the tag sends SENS_RES and enters RESOLUTION* State. If any other command is received, the tag shall stay in SLEEP State.
RESOLUTION* State	This state is similar to the RESOLUTION State, except that the tag shall enter SELECTED* State when it's selected with its complete NFCID1.
SELECTED* State	This state is similar to SELECTED State

Table 6: Possible states of operation for an NFCIP-1 device.

The command set used in Table 6 is defined in Table 7.

Mnemonic	Definition
SENS_REQ	Sense Request (sent by Initiator)
SENS_RES	Sense Response (sent by Target)
ALL_REQ	Wakeup ALL Request (sent by the Initiator)
SDD_REQ	Single device detection Request (sent by Initiator)
SEL_REQ	Select Request (sent by Initiator)
SEL_RES	Select Response (sent by Target)
SLP_REQ	Sleep Request (sent by Initiator)

Table 7: Command set for NFC initialization procedure [1].

SENS_REQ and ALL_REQ are used by a reader to activate all targets in the field and shall be transmitted in a Short Frame. ALL_REQ is used to put all targets in SLEEP State back into RESOLUTION State so that they can participate in further SDD procedures. The coding of these commands is shown in Table 8 [1].

Short Frame Commands	b6	b5	b4	b3	b2	b1	b0
SENS_REQ	0	1	0	0	1	1	0
ALL_REQ	1	0	1	0	0	1	0
Proprietary	1	0	0	x	x	x	x
Proprietary	1	1	1	1	x	x	x
RFU	All other values						

Table 8: Coding of commands used in short frames [1].

When receiving a SENS_REQ, all tags in SENSE State shall respond. When receiving an ALL_REQ, all tags in both SENSE State and SLEEP State shall respond. The Initiator shall detect any collision that occurs if more than one target responds. The coding of SENS_RES, or bit frame SDD, is described by the Table 9, Table 10 and Table 11[1].

bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ZER 0	ZER 0	ZER 0	ZER 0	Proprietary coding				NFCID1 size bit frame	ZER 0	Bit frame single device detection					

Table 9: Coding of SENS_RES [1].

bit 7	bit 6	Explanation
0	0	NFCID1 size: single
0	1	NFCID1 size: double
1	0	NFCID1 size: triple
1	1	RFU

Table 10: Coding of “NFCID1 size bit frame” bits in SENS_RES [1].

b4	b3	b2	b1	b0	Meaning
1	0	0	0	0	bit frame SDD
0	1	0	0	0	bit frame SDD
0	0	1	0	0	bit frame SDD
0	0	0	1	0	bit frame SDD
0	0	0	0	1	bit frame SDD

Table 11: Coding of “bit frame SDD” bits in SENS_REQ [1].

The next commands used in the SDD procedure are SDD_REQ and SEL_REQ, and shall be coded according to Table 12.

Size	1 byte	1 byte	0-40 bits
Content	SEL_CMD	SEL_PAR	NFCID1 CLn according to SEL_PAR

Table 12: Coding of SDD_REQ and SEL_REQ [1].

The SDD_REQ command shall be transmitted within a bit oriented SDD Frame, while the SEL_REQ command shall be transmitted within a Standard Frame. If SEL_PAR specifies 40 valid data bits (SEL_PAR = 70 hex), a CRC shall be appended and the command shall be called SEL_REQ. Else the command shall be called SDD_REQ and the target stays in the RESOLUTION or RESOLUTION* State. If a target has sent its complete NFCID1, it shall transfer from RESOLUTION State to SELECTED State (or RESOLUTION* to SELECTED*) and indicate in the SENS_RES that the NFCID1 is complete. Else the target stays in RESOLUTION State (or RESOLUTION*) participating in a new SDD with increased cascade level. The SEL_CMD shall be coded as shown in Table 13.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Hex notation	Meaning
1	0	0	1	0	0	1	1	(93)	Select cascade level 1
1	0	0	1	0	1	0	1	(95)	Select cascade level 2
1	0	0	1	0	1	1	1	(97)	Select cascade level 3
Other settings are RFU									

Table 13: Coding of SEL_CMD [1].

The coding of SEL_PAR is defined in Table 14 and Table 15.

bit 7	bit 6	bit 5	bit 4	Meaning
0	0	1	0	Byte count = 2
0	0	1	1	Byte count = 3
0	1	0	0	Byte count = 4
0	1	0	1	Byte count = 5
0	1	1	0	Byte count = 6
0	1	1	1	Byte count = 7

Table 14: Coding of the 4 “upper” bits of SEL_PAR [1].

bit 7	bit 6	bit 5	bit 4	Meaning
0	0	1	0	Byte count = 2
0	0	1	1	Byte count = 3
0	1	0	0	Byte count = 4
0	1	0	1	Byte count = 5
0	1	1	0	Byte count = 6
0	1	1	1	Byte count = 7

Table 15: Coding of the 4 “lower” bits of SEL_PAR [1]

Bit 7 to 4 (“byte count”) shall indicate the number of bytes sent from the initiator, and is computed by dividing all valid data bits (including SEL_CMD and SEL_PAR) by 8. Bit 3 to 0 is called “bit count” and is computed by applying modulo 8 to all the valid data bits (including SEL_CMD and SEL_PAR).

If the 40 data bits of UID sent from the reader in the SEL_REQ matches the actual UID of the tag, it shall respond with SEL_RES followed by two CRC bytes used to

check for errors in the transmission. The SEL_RES shall be coded according to Table 16.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Meaning
x	x	x	x	x	1	x	x	Cascade bit set: NFCID1 not complete.
x	1	x	x	x	0	x	x	NFCID1 complete, Target compliant with NFC transport protocol. Request for Attributes supported.
x	0	x	x	x	0	x	x	NFCID1 complete, Target not compliant with transport protocol, Request for Attributes not supported.

Table 16: Coding of SEL_RES [1].

The CRC shall be computed by CCIT CRC-16, which is defined by the following polynomial:

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

The initial value for the 16 stage shift register shall be 6363h.

A flowchart for a complete initialization and SDD sequence is illustrated in Figure 11.

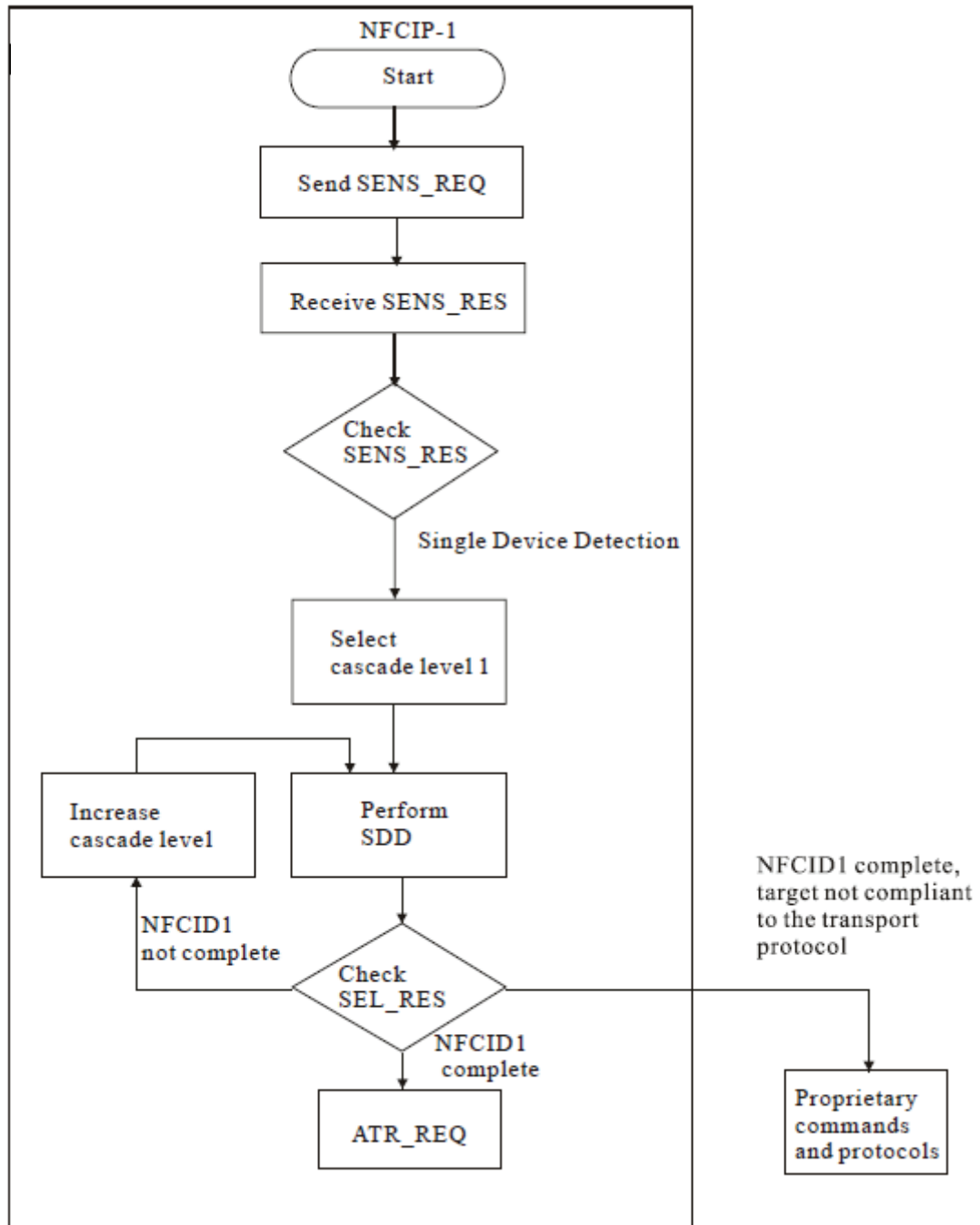


Figure 11: Flowchart for NFC initialization and SDD, at the Initiator side [1].

If a collision is detected, an SDD loop shall be performed for each cascade level in order to select one single device for communication. The cascade levels are defined in Table 17.

NFCID1 size	Cascade level	Number of NFCID1 bytes
single	1	4
double	2	7
triple	3	10

Table 17: Cascade levels of an NFCID1.

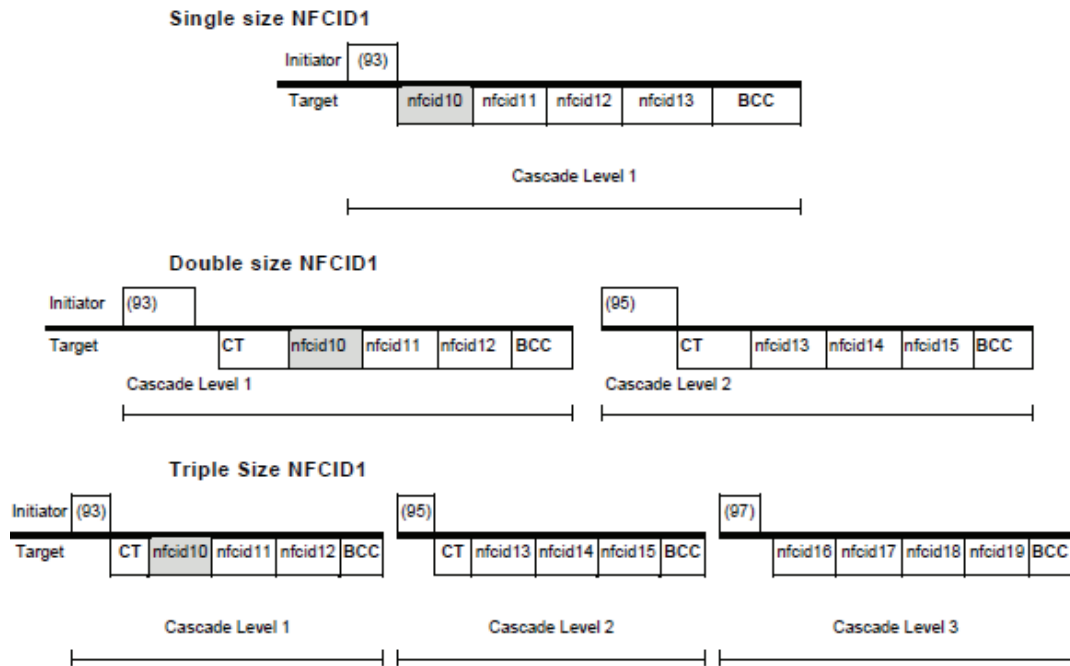


Figure 12: Usage of cascade levels [1].

The SDD command sequence for each cascade level is shown in Figure 12.

The BCC is a check byte for the cascade level length and is computed by performing exclusive-or over the four bytes prior to the BCC.

If an Initiator wants to set a Target in SLEEP State, it shall send a SLP_REQ command coded as shown in Figure 13. The target acknowledges the command by keeping inactive for a period of 1ms after the end of the frame transferring the SLP_REQ.

Byte 0	Byte 1	Byte 2	Byte 3
(50)	(00)	CRC	

Figure 13: Coding of SLP_REQ command

2.2.7. Mode selection

There are three different contactless close-coupling communication standards operating at 13,56MHz, namely ISO/IEC 18092 (NFCIP-1), ISO/IEC 14443 (PCD) and ISO/IEC 15693 (VCD). Although they use the same frequency, they all specify distinct communication modes. To make NFC devices able to choose between these modes, ISO/IEC 21481 (NFCIP-2) [6] has been developed (also standardized in ECMA 352 [7]). It specifies a selection procedure of the three communication modes and at the same time ensures that no disturbance is caused to any ongoing communication at 13,56MHz. The mode selection procedure is illustrated in Figure 14.

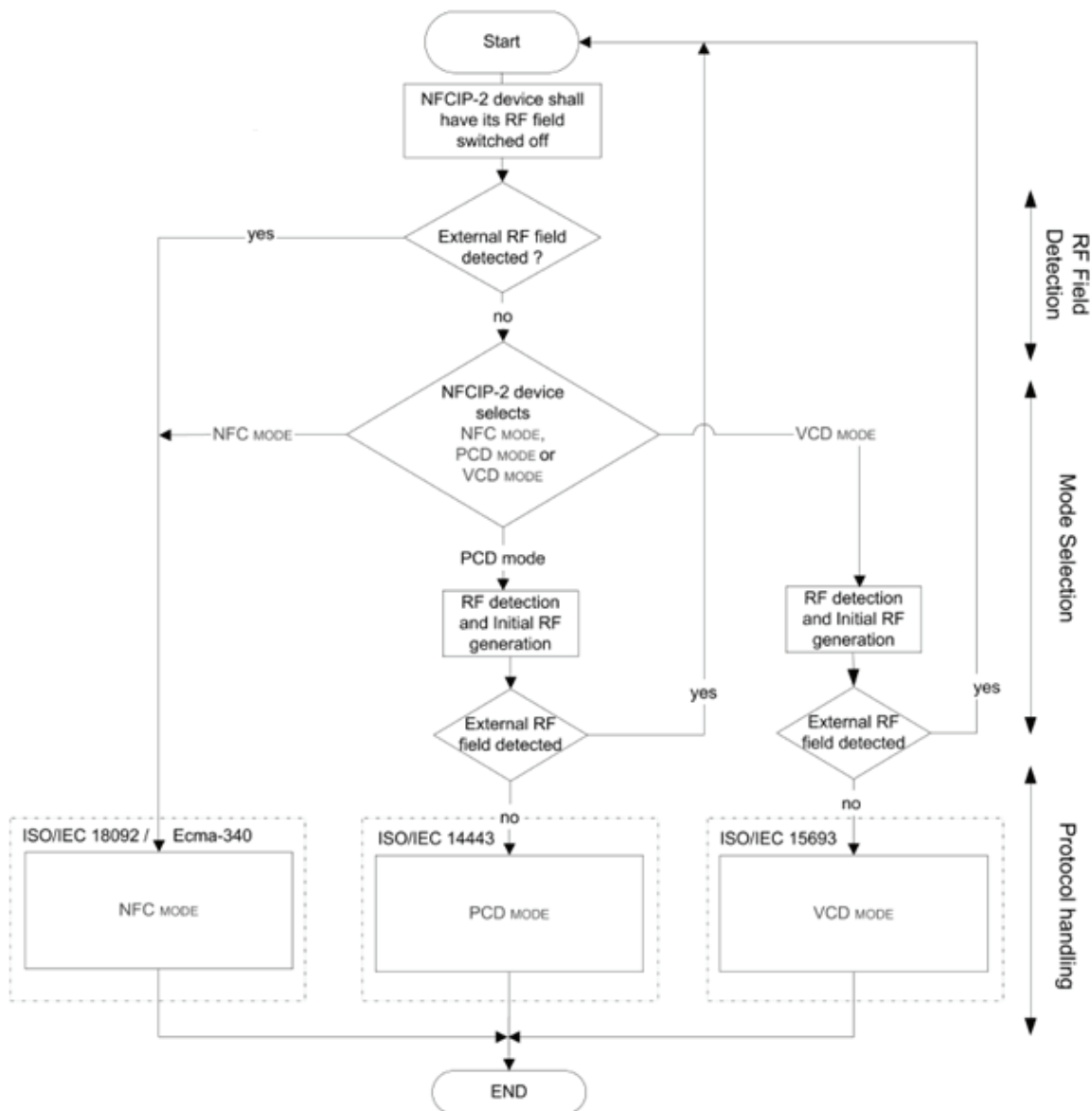


Figure 14: Mode selection procedure for an NFCIP-2 device.

The external RF field detection, RF detection and Initial RF generation is the same as for ISO/IEC 18092 with one exception, the T_{IRFG} (Figure 6). This value defines the initial guard time between switching on the RF field and starting the modulation. An NFCIP-2 device shall adjust this value according to the standard of the chosen communication mode.

2.2.8. NFC protocols

NFC Data Exchange Format (NDEF) is an NFC Forum specified message encapsulation format for information exchange between NFC Forum Devices or between an NFC Forum Reader and an NFC Forum Tag [8]. It is a binary message format for exchange of application payloads of any type and size within a single message. A payload is described by a type, a length and an optional identifier. Possible types are URIs, MIME media types and NFC-specific types. The length is an unsigned integer indicating the number of bytes in the payload, with a compact short-record layout for very small payloads. The optional identifier may be used to handle multiple payloads, and cross-reference between them. Payloads may include nested messages or chains of linked chunks with unknown length at the time the data is generated. NDEF is only a message format and keeps no knowledge of connections or logical circuits.

The NDEF specification defines the NDEF data structure format and rules to construct a valid NDEF message as an ordered and unbroken collection of NDEF records. NDEF defines the structure for application and service specific data, but detailed record type definitions are defined in separate specifications. NDEF assumes a reliable underlying protocol and does not specify how the data are to be transferred between NFC entities. The message size is $2^{32} - 1$ bytes.

Based on the NDEF, NFC Forum has developed 4 different Record Type Definitions for NFC [27], namely Text, URI, Smart Poster and Generic Control. The purpose is to enable users to effectively create their own applications using a well designed and free available set of formats and rules.

There are also four different Tag Type operations named NFC Type 1-4 Tag operation that must be supported by an NFC Forum device. The tag type used in this task is the “NFC Forum Type 1 Tag Operation Specification”, which is based on ISO/IEC 14443A. A tag is by default re-writeable, but can be configured to read-only mode. The standard memory is 96 bytes, but can be expanded to 2kbytes. The

communication speed is 106kbps. Type 2 is based on ISO/IEC 14443B and Type 3 is related to FeliCa. Type 4 is fully compatible with the whole ISO/IEC 14443 standard series, with re-write or read-only pre configuration, memory up to 32kbytes, Type A or Type B compliant communication interface and data transfers speeds up to 424kbps.

2.2.9. Antennas

The typical antenna used within contactless cards, RFID and NFC is a loop antenna. One of the reasons for this is that the effective area of the antenna is squared for every new loop, making it very effective relative to the size. As these protocols are based on induction in the antenna to power the internal circuits, they have to be as effective as possible within the pretty limited physical space available. The effectiveness of an antenna is measured by a Q-factor, or quality factor. This is based on the resonance frequency, the number of turns and the effective area. The resonance frequency (f_{RES}) should be near the carrier frequency, but not below. Although giving better Q-factor, the number of turns cannot be increased unlimited. There has to be a negotiation between reading range and transfer speed, as the quality factor (Q) is inversely proportional to the bandwidth of the transponder resonant circuit if f_{RES} is constant [11].

The typical NFC-tags have a square loop with approximately 3 cm sides, rounded corners and 9 turns. Making the tag antennas is not so difficult, as they shall only support passive communication mode (i.e. they do not have to generate their own magnetic field). The radio-environmental surroundings are also less complex, usually limited to normal amount of radio noise and maybe the reflection effect of being placed on a metal surface. Integrating an antenna in a cell phone is however much more complex. The integration has to be done individually on each phone, to make both the GSM antenna and the NFC antenna as effective as possible. As for phones with manufacturer integrated NFC-capabilities, the antenna is usually integrated on the PCB (Printed Circuit Board). The effectiveness of the antenna depends on the rest of the metal chassis and other components that may interfere with the radio communication. The antenna must also be able to generate a magnetic field within the limits defined by ISO/IEC 18092. To accomplish this none of the two functions can be made with optimal performance. Tests performed in this thesis will show the reading

distance differences between plain tag antennas and the integrated antenna in NOKIA 6212 Classic.

To make an antenna work as intended the impedance of the antenna should be as close to 50 ohms as possible. This has to be done by a separate impedance-matching circuit, or can be integrated in the antenna design by introducing metal surfaces, overlapping turns and insulating material to create the proper resistance, capacitance and inductance.

If the internal circuits had its own power-supply, the antenna could have had another design. With the possibility to amplify the signals in the receiver, much lower signals may be picked up. This fact is to be exploited by this thesis in order to prove that the communication channel can be eavesdropped. Except the typical loop antenna, other antennas could be effective in order to pick up the signal from larger distances than approximately 10cm. Log-periodic and parabolic antennas are far more directive than loop, dipole and whip-antennas. This give more gain in the antenna and the limit for actual received power can be reduced, i.e. the reading distance is increased.

2.2.10. Application examples

NFC has numerous potential areas of use. The main idea is to reduce the size of people's wallets, and ease the process of retrieving the customer's part of any application. Mobile phones and SIM-cards of the future (UICC), with memory and processor capacity to download, store and run various software applications makes a powerful platform to be exploited together with NFC. The three maybe most promising areas are public transport payment, credit card replacement and advertising. Other possibilities are membership cards, identity cards, electronic keys for access control and initiation, configuration and security parameter negotiation for other communication protocols such as Bluetooth and Wi-Fi.

Today many public transport companies offer a solution with value cards that are presented to a reader when entering a bus or a train. The cards may be magnet strip, chip or contactless. This can be replaced by the card emulation mode of NFC, so that the number of cards a consumer has to be in possession of can be reduced. This might also reduce costs related to card management for the service provider.

Advertising, or smart posters, is also an interesting feature. For this service the phone acts as an NFC reader, and collects information on strategically placed tags. This can be in store advertisements such as offer of the day, provisional poster info such as

movie trailers, or payment information from vending machines that is relayed via SMS or MMS. The tag can contain all the information needed, or the tag can give an URI combined with a phone command so that the user is redirected to a phone number or a website to complete the service.

The bank industry is an enormous consumer of various plastic cards or smart cards, until now mostly magnet strip and chip cards. The cards have limited lifetime and are vulnerable to demagnetizing and breakage. By replacing the cards with software applications, great expenses can be eliminated. The fact that many cards can be replaced by one phone will also ease the detection of a lost card, as the user has only one physical item to keep control of. The obvious disadvantage is of course based on the same fact. If you lose the phone, you may lose access to a lot of necessities at the same time. However, the loss of a wallet will in many cases have the same impact.

2.3. RFID analysis

2.3.1. General

The number of deployed RFID applications has increased significantly the last decades. As more and more sophisticated services have been introduced, the interest amongst groups with malicious intentions has been growing. This has led to development of various security solutions to protect the applications against attacks, such as authentication and encryption. MIFARE, which is manufactured by NXP Semiconductors, has introduced ISO/IEC 14443 compliant solutions with support for 3DES, AES, RSA. Their first attempt to implement security was Crypto-1, an NXP proprietary stream cipher with 48 bit key made for MIFARE Classic. Because of low cost and good reliability, this version has been heavily deployed in electronic wallet, public transportation and ticketing applications. Several publications during 2007 and 2008 describe how to break this cipher, one of them with a secret key recovery time down to 40 ms on a laptop [35].

2.3.2. Eavesdropping

The fact that RFID applications uses radio communication to transmit data, gives attackers the opportunity to pick up information from a distance without physical access to any of the communicating units. Gerhard Hancke has published some papers

regarding the radio interface of RFID. He is a research assistant with the ISG Smart Card Centre at Royal Holloway, University of London.

In “*A Practical Relay Attack on ISO 14443 Proximity Cards*” [14], he shows that it is possible to perform a relay attack on proximity card communication. By introducing a fake reader (mole) and a fake tag (proxy) between a genuine reader and a user tag, he showed that data could be relayed at distances up to 50 meters. The proxy is placed near the real reader, and has RF communication with the mole. If a card is located within the reading range of the mole, the card is powered and believes that it’s communicating with a genuine reader. Unless another card is presented to the genuine reader at the same time as the proxy is in relay operation, the reader is also unaware of the attack. The relayed data can be stored for further analysis, and possibly used for cloning of the tag.

In “*Eavesdropping Attacks on High-Frequency RFID Tokens*” [15], he describes how to eavesdrop RFID communication from different RFID tokens, both ISO/IEC 14443 and ISO/IEC 15693. As NFC has compatibility modes for all of these tags, these test results are of great interest. Because a reader and a tag use different modulation techniques, the communication is split into two channels as shown in Figure 15.

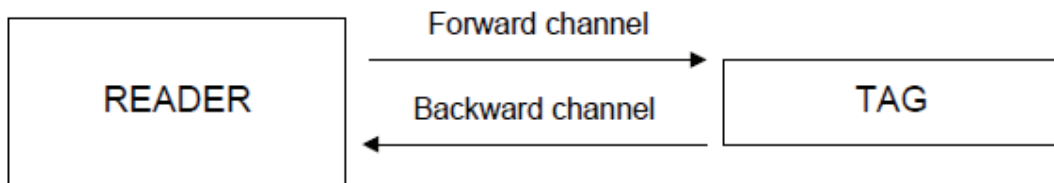


Figure 15: Communication channels used in Hancke’s experiment.

The test results presented in Table 18 [15] show that the forward channel in ISO/IEC 14443A can be picked up at a much greater distance than the backward channel, and that the reading ranges differ according to both tag type and the test environment.

	ISO 14443A	ISO 14443B	ISO 15693
Entrance hall			
1 m	FB	FB	FB
3 m	Fx	xB	Fx
5 m	Fx	xx	Fx
10 m	Fx	xx	Fx
Lab corridor			
3 m	FB	FB	Fx
4 m	Fx	xB	Fx

Table 18: Eavesdropping results of Hancke’s test [15].

The F indicates that the forward channel is recovered, while the B indicates that the backward channel is recovered. With a proper radio receiver like the one used in this test, the backward channel of ISO/IEC 14443A can be eavesdropped up to 1 meter away. Since NFC is compliant with this standard, it should imply that NFC should have similar properties. The fact that the forward channel can be read at distances up to 10 meters is also very interesting, as NFC in active communication mode uses this modulation technique for both communication directions. It indicates that NFC in active communication mode is more vulnerable to eavesdropping than NFC in passive communication mode.

2.3.3. Analyzer tools

Many organizations, researchers and other people with personal interest of the subject are putting a great effort into proving and exploiting the potential weaknesses of RFID. Jonathan Westhues has designed a device called PROXMARK, which can read and clone RFID tags and also sniff RFID communication [16]. A manual [17] on the designer’s web page provides information on how to build the device, and how to make antennas out of enamel wire or by stripping and modifying a standard USB cable. It also explains all the features of the device and how to make it sniff an ISO 14443A communication sequence. The documentation of the claimed functions seems valid according to the relevant standards, so there should be little doubt in the device’s actual performance. This work shows that it’s possible to make equipment that efficiently retrieves data from short range radio communication at relatively low cost. The author also links to another web page where it’s possible to buy a fully assembled

and tested PROXMARK device at 499\$ [18]. This is significantly decreasing the threshold for those who want to have a go at experimenting with RFID technologies. There is no longer need for advanced skills in radio communication and PCB assembly, nor high tech soldering equipment. Both hardware and software is open and available.

The device described here is a multifunctional unit supporting two frequency bands, several modulation techniques and different types of antennas. This implies that the size of the circuitry can be reduced and the effectiveness of the radio receiver increased by optimizing for one specific standard, for example NFC. If the transmitter part is also skipped, these effects should increase further. By introducing larger memory and an internal power supply (batteries), you have a skimmer able to store a certain amount of NFC transactions. This is the same concept as a magnet strip ATM skimmer used in nowadays VISA and credit card frauds.

2.4. Security aspects

All types of wireless communication are vulnerable to a number of threats. It's easy to listen into the channel and the intended signal can easily be disturbed by other signals. The latter is beyond the scope of this task. The problem with radio transmission is that malicious users can access the communication channel undetected. Without any protection, it's possible to pick up messages, alter information in live communication and store messages with the intention to replay them with the same or altered content at a later time. To overcome the threats, the system needs to implement authentication, integrity check, confidentiality and replay protection. The level of security needed is defined by the application itself. Whenever money is involved, an application will attract potential malicious users. Private content sharing on the other hand may not require the same level of security. As NFC is made to be compliant with ISO/IEC 14443, it's reasonable to expect that the two standards will coexist for a quite some time. This implies that the security solutions will be developed as derivatives of those made for ISO/IEC 14443. MasterCard and VISA are big actors in this area. The equipment vendors themselves also implement different security solutions. Some of the work of these companies is public, but specific technical specifications are kept within the companies and their trusted partners.

The passive communication mode seems like a good solution for transfer of sensitive data, as this mode seems harder to eavesdrop compared to the active communication

mode. The short set-up time is also crucial for applications such as public transport payment and access control. This feature does however apply to both modes.

2.5. Deployed security mechanisms

Together with the recent ECMA NFC security protocols under evaluation, there are some mechanisms published for the ancestors, i.e. RFID technologies. MIFARE did early on deploy their first security algorithm for contactless cards, called Crypto-1. This was however broken in the early 2000's. The latest solutions from both MIFARE and FeliCa are approved at minimum Common Criteria (CC) assurance level EAL4, and the hardware of MIFARE SmartMX has passed EAL5+ evaluation. CC is standardized in ISO/IEC 15408, which is a certification standard for IT security. EAL4 is the highest EAL-level which is expected to be economically feasible while implementing the security in an existing product line [23]. EAL5 is applicable for systems requiring high level of independently assured security in a planned development. The design process is rigorous, but should keep the cost related to security specialists at a reasonable level [23].

In their RC-S860 chip, FeliCa has implemented mutual authentication using 3DES and data encryption by a transaction key generated in the authentication procedure. FeliCa claims in the product description that "These features make forgery and card fraud nearly impossible" [37].

MIFARE has a range of solutions supporting different levels of security. In 1997 they introduced MIFARE Pro with a 3DES cryptographic coprocessor. This was followed by ProX in 1999, which had a PKI coprocessor [35]. As of today, they have solutions employing AES, 3DES, RSA, ECC, one way counters and PRNGs [36]. Both MIFARE ProX and SmartX are highly flexible microprocessor based cards with cryptographic coprocessors supporting all the mentioned ciphers together with a Fast RNG [35]. The hardware is however passive and has to be programmed by an operating system (OS). Both proprietary OSs and open source OSs like JavaCard are supported. As authentication MIFARE has introduced a three-pass mutual authentication compliant to the ISO/IEC 9798-2 [24] standard [36], which describes different authentication techniques using symmetric ciphers. A three-pass authentication procedure as defined in this standard is illustrated in Figure 16.

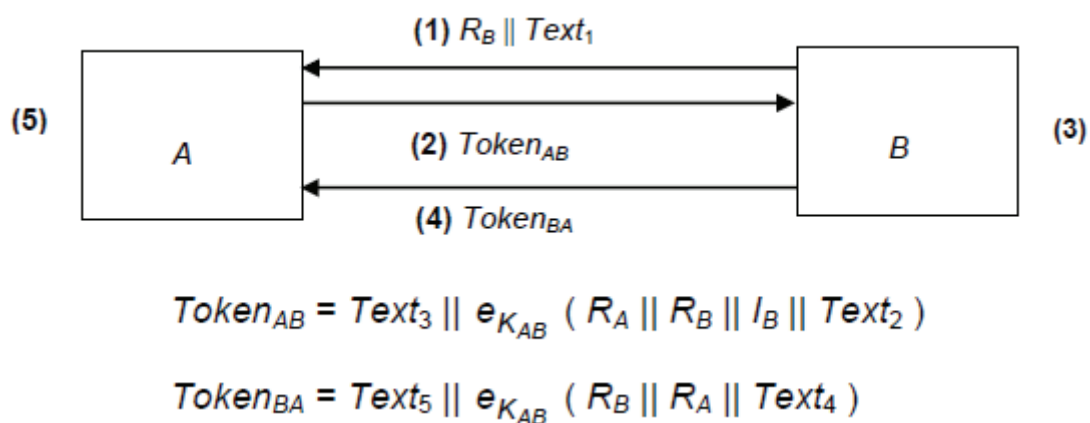


Figure 16: Three-pass authentication procedure [24].

The authentication procedure is performed in 5 steps:

- (1) B generates a random number R_B which is sent to A.
- (2) A generates a random number R_A , and then a $Token_{AB}$ which is sent to B.
- (3) B decrypts the token and checks the R_B and the I_B .
- (4) B generates and sends $Token_{BA}$ to A.
- (5) A decrypts the token and checks the two random numbers.

Using this protocol, uniqueness is controlled by generating and checking random numbers. According to the standard, three passes is needed for mutual authentication while there is no trusted party involved and random numbers are used for challenge and response generation.

An entity corroborates its identity by proving knowledge of a secret key, by encipherment of the random numbers (and other optional content) using symmetric ciphers. Introducing a random sequence from both parties prevents one party from performing a complete chosen plaintext or reuse of numbers from earlier sections. If a reflection attack by reusing tokens is feasible, the distinguishing number I_B is used to assure that a token is accepted only once. The text fields are optional and may contain additional timeliness information such as timestamps and sequence counters, or key index if multiple key exists between the two parties. The text fields can also be used by additional protocols related to the authentication procedure, such as key distribution or key agreement protocols [25]. The security of this protocol is based on the quality of the random numbers, which should not be repeated during the lifetime of an authentication key. This lifetime should be set so that it prevents chosen plaintext or known plaintext attacks, and the key should only be known by trusted entities. The tokens must be impossible to forge, even if an attacker has gained knowledge of old tokens. This is solved

by using an authenticated encryption technique that is providing both data confidentiality and data integrity.

As the RFID readers and tags use similar technology as NFC, this section shows what level of complexity that is possible to implement on tags and cards with today's memory and processor capacity.

3. EAVESDROPPING EXPERIMENT

This chapter will describe an experiment which seeks to prove that NFC-communication can easily be eavesdropped using simple methods and basic “off the shelf” equipment. Detailed documentation and test results can be found in Appendix A. Documentation, user manuals and source code for the developed test software is presented in Appendix B. Understanding this chapter requires good knowledge about the technical features of NFC, as presented in Chapter 2.2.

3.1. Experiment outline

The purpose of this experiment is to prove that it is possible to pick up information sent between NFC devices by the use of simple equipment, in order to show that sophisticated security mechanisms are needed when developing NFC applications dealing with sensitive information. To perform this I will try to read out sent data between an NFC reader and an NFC tag by the use of an oscilloscope and an improvised antenna, as shown in Figure 17.

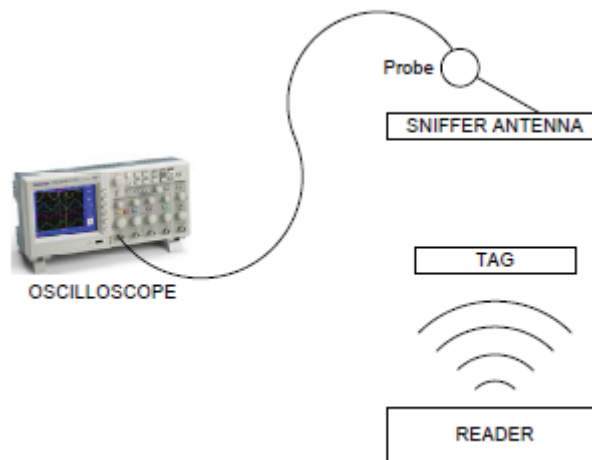


Figure 17: Test set-up illustration.

If the improvised antenna is good enough, it will be possible to visually demodulate the modulated RF-carrier based on the modulation techniques described in Chapter 2.2.3.

3.2. Test equipment

3.2.1. General

As shown in Figure 17, the basic test assembly consists of a reader, a tag and a passive antenna connected to an oscilloscope. To perform accurate reading distance measurements, an adjustable rack has been built. To make sure that the rack was transparent to RF-signals all the parts were made in plastic. The NFC reader used in this experiment is an ACR122U, a product of Advanced Card Solutions (ACS) in Hong Kong. This reader supports NFC, ISO/IEC-14443 Type A, ISO/IEC-14443 Type B, Mifare and FeliCa. It was bought as a part of “ACR 122 Starter Kit”, an SDK-solution for contactless application development. The NFC-enabled cell phone used the test are a NOKIA 6212 Classic. Three NFC tags from NOKIA were also enclosed in the box. Detailed information about the test equipment can be found in Appendix A. To analyze the RF-signals, a Tektronix TDS 2004B digital storage oscilloscope has been used. By the use of manufacturer provided software this scope can be connected to PCs to capture screenshots and signal sample sequences.

3.2.2. Improvised antenna

To be able to capture the radio signals, you need an antenna matched to the frequency of the carrier signal, i.e. 13,56MHZ in this particular case. My first idea was to make a loop antenna on a Printed Circuit Board (PCB), or a coiled loop antenna with thin insulated wire. To make effective antennas of these types, the coil radius and number of coils have to be optimized for the RF signal. The antenna also has to be impedance matched to 50Ω in order to work as intended. To find the optimal antenna design, I performed reading range test of the tags and cards provided with the reader and the mobile phone. This test was performed using the ACR122U Tool-software that was enclosed in the SDK. By enabling automatic PICC polling and a polling interval of 250ms it will automatically detect any card or tag within range. Whenever a tag is within communication range, the LED on the reader will turn green. When the LED stays continuously green for at least 10 seconds (40 polls), the reading is considered successful. The test set-up configuration is shown in Figure 18.



Figure 18: Reading range test set-up.

Tag type	Reading range (in cm)
Mifare UltraLight Label	6,0
Mifare 1K Label	5,6
Mifare 1K Card	7,0
Nokia Tag	6,4
Nokia 6212 Classic in Card Emulation Mode	3,0

Table 19: Reading range test results for all available tag types.

As shown in Table 19, the results vary quite a lot. As the MIFARE 1K card had the best radio performances, I tried to open one to see the layout of the coils. It was however not possible to separate the upper and lower parts of the card just using a knife, without destroying the metal wire. I could have counted the number of turns and measured their thickness, to get an idea of the design. The placement and size of the antenna within a typical plastic card is explained in detail in ISO/IEC 14443-1 [2]. However, I decided to have a look at the tags first, since some of them were pretty close to the cards reading range. As shown in Figure 19, the two MIFARE labels have the same physical design, while the Nokia tag was a little bit bigger and had a bit more complex circuitry.

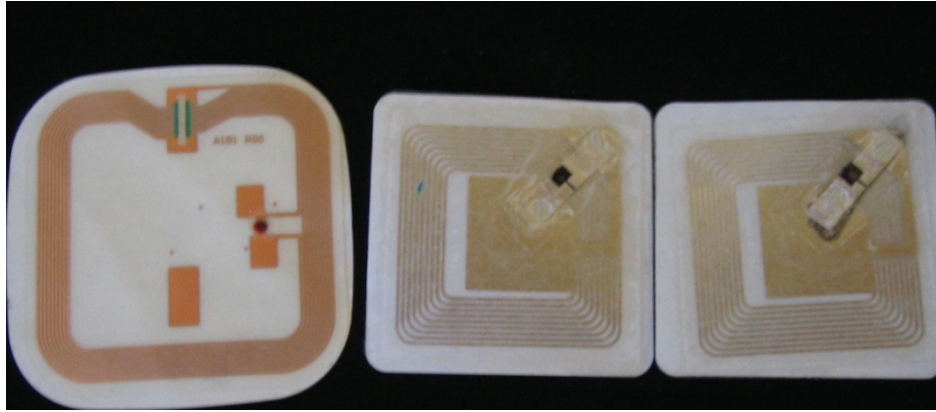


Figure 19: Label circuitry design. (From the left: NOKIA, MIFARE 1K, MIFARE UltraLight)

When further inspecting the MIFARE tags, I found that it could be possible to modify a tag to make a passive antenna. The tag consists of a matched antenna connected to an IC. The IC type is dependent of the tag type, and contains transceiver, modem and memory. The whole circuitry is printed on glued plastic, and coated with another layer of plastic. By removing the top layer it's possible to access the printed circuit. The antenna has two connection points, one in the centre and one in a corner. The IC is located in the middle of these two points, connected with a thin metal film as illustrated in Figure 20.

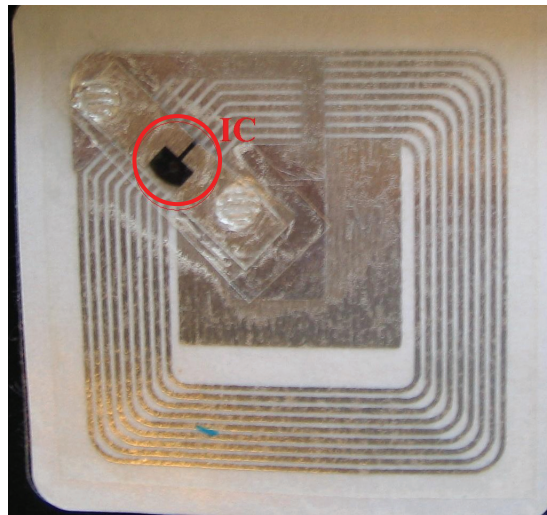


Figure 20: Close-up picture of a MIFARE UltraLight label with marking of the IC.

I was able to remove the IC with a pincer and a scalpel, and then I had a completely passive and impedance matched antenna with two connection points. This was all I needed to be able to connect a measurement probe from the oscilloscope to the antenna as shown in Figure 21 and Figure 22.

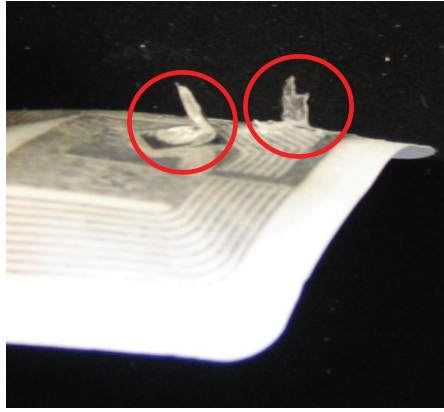


Figure 21: Modified label with marking of connection points

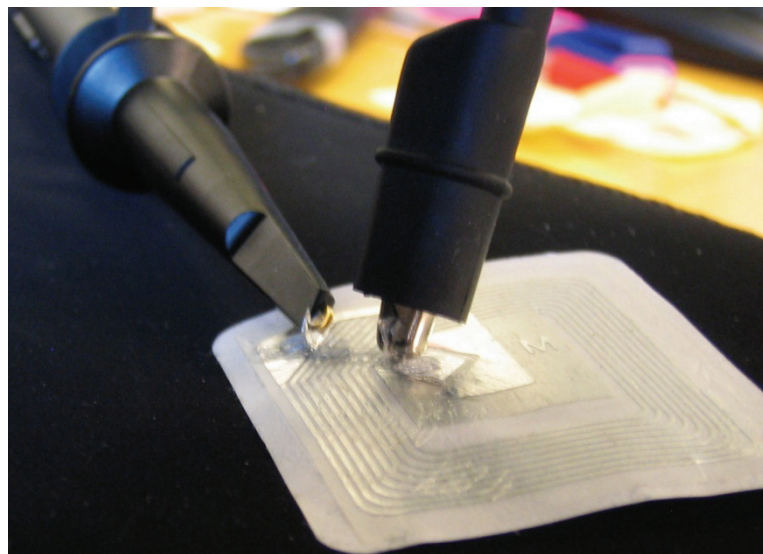


Figure 22: Modified label with measurement probe connected.

Some quick tests showed that this antenna gave good readings of the magnetic field, so I settled with this antenna “design” for three reasons. First, the goal of this thesis was to prove the possibility of eavesdropping on the channel, not maximizing the reading distance. Second, the size of the tag implies that it should be possible to make a passive receiver device with batteries and enlarged memory in approximately the same size. Lately, batteries in the shape of μm film have been developed [19], which might power such tags without great influence on the total size. Third, as this tag is commercially available and pretty cheap (2£ per piece [20]), it’s possible to make an antenna without having to know anything about antenna design. Easy design and methods is a point in this thesis, as the number of potential attackers grows fast when an attack gets easier.

Late in the experiment work I found that there was time to perform some additional tries with the MIFARE classic card. The antenna coil is connected to the IC via two thin copper wires. When holding the card up against a bright light, it's possible to see the placement of the IC. Then I tried to scrape off the covering plastic to reveal the IC and enough wire to connect the test-probe. After some attempts I was able to deface some millimeters of wire without breaking it so that I could connect the probe, as shown in Figure 23. According to the reading range results in Table 19, this antenna should give greater eavesdropping ranges than the tag-antenna.

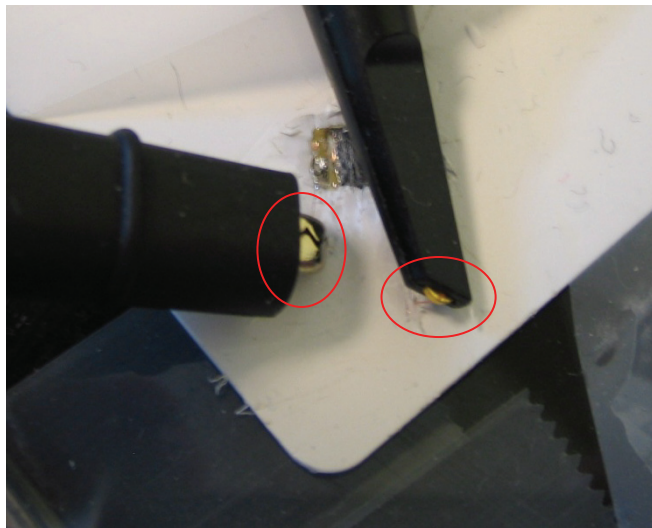


Figure 23: Probe connected to the antenna of a MIFARE Classic card.

3.3. Test software

To be able to make accurate measurements of the transmitted signal, a test program was needed. To be able to interpret the information on the scope, I had to be sure that it triggered on the same sequence every time. The SDK contained a lot of example code written in multiple programming languages, I chose Java as it's the one I'm most familiar with. The sample code didn't have the exact features I needed, so it had to be modified. The desired function was to disable the RF field, and restart it by command when the measurement instruments were ready. The optimal solution was to send a SENS_REQ to a tag in unpowered mode. The provided sample code couldn't connect to the reader unless a tag was present. The reader also has a function for automatic PICC polling, which when enabled also activates the RF-field. To be able to turn off the radio power, I first had to disable automatic PICC polling. When the RF-field is turned off in this case the present tag is put into sleep mode instead, and the reader remembers which tag it was communicating with. To wake up units in sleep mode, a reader sends an ALL_REQ and

any tag in the field shall answer with SENS_RES. If no collision is detected, the reader sends a SEL_REQ or a SDD_REQ to the tag. If the tag has 40 bits UID, SEL_REQ is used. Else the command is called SDD_REQ. If the command is SEL_REQ and tag is successfully selected, it answers with a SEL_RES followed by two CRC bytes. This sequence is stable and repetitive for each test cycle when reading the same tag, and fulfills the requirements for a test sequence on which the oscilloscope can trig. The interface of the test program is shown in Figure 24.

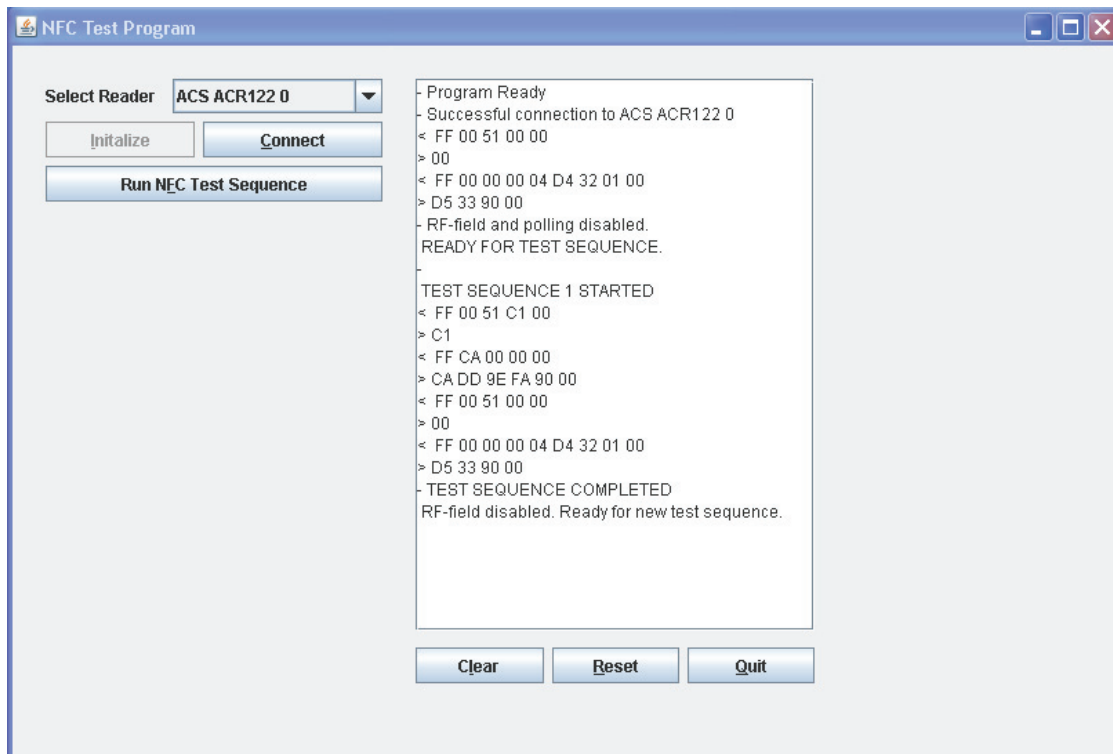


Figure 24: User interface of the developed “NFC Test Program”.

The text window is showing connection information, which commands that are sent and received, and the status of the test cycle. A counter shows the number of test cycles performed to make it easy for the user to see that a test is actually performed. This is done because the text field apparently just repeats the content under normal and successful test conditions. If any errors occur, error messages will be displayed.

To be able to check the CRC bytes, I made a small calculation program based on source code found on the Internet [22]. The program was tested by checking the examples values describe in “Annex A” of ISO/IEC 18092 [1] to verify correct calculations. By entering the data bits lsb first you get the two CRC bytes presented with lsb first, as illustrated in Figure 25.

```
C:\Documents and Settings\bruker>CRCCalculationProgram.exe
Enter bitstring to calculate (lsb first):0000000000000000
CRC of [0000000000000000] is [0000010101111000] with P=[10001000000100001]
```

Figure 25: “CRC Calculation Program” used to compute CRC of “00000000 00000000”.

This program is made for the CRC computation at 106kbps in passive communication mode, with initial shift register value of 6363h. If it shall be used for other data speeds or active communication mode, the initial value has to be changed in the source code.

3.4. Eavesdropping test procedures

3.4.1. General

For each reading in this test, a step-by-step procedure has been followed. As the oscilloscope has a pretty small display window, only a few bits of information can be visually interpreted at one time. This makes the work very tedious, but though pretty easy once you get into it. It’s important to know what you are looking after, i.e. what type of modulation you expect to see and where in the signals time domain the modulation should appear. Another challenge is to configure the oscilloscope to display the signal in the way you want and to trig on the desired events.

3.4.2. Test plan

To prove that it is possible to eavesdrop on NFC communication a number of tests have been planned, increasing the complexity by steps. In order to start the testing, a proper antenna has to be designed. The first test goal is to be able to find the modulated signal and decide whether specific bit durations indicates a “0” or a “1”. The next step is to recognize complete commands, and check whether they match with the communication sequences describe in the standards. If single commands are able to be captured, the next challenge will be to capture a whole communication sequence. If all the previous tests are successful, the final goal will be to perform maximum reading range tests. As these tests are performed with completely passive equipment, they will only give an indication of the eavesdropping possibility. One can expect that the ranges can be increased significantly by introducing a radio receiver with signal amplifiers.

3.4.3. ASK reading

The ASK signal is the easiest to find and convert to byte code. This is the communication from the reader to the tag, and is thus the first signal will appear in the time domain. The signal properties described in Chapter 2.2.3 is used to decide how to read out bits from the signal displayed on the oscilloscope. A bit duration is $9,44\mu\text{s}$, and a pulse with duration around $2\text{-}3\mu\text{s}$ may occur within the bd. Figure 26 shows how one single bd looks on the oscilloscope. The screen covers $10\mu\text{s}$ of the signal, which is a little bit more than one bd.

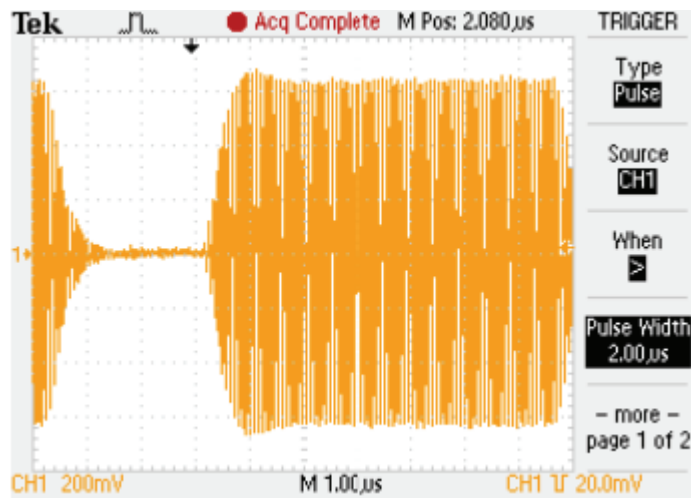


Figure 26: A 100% ASK pulse that occurs in the beginning of a bit duration.

The pulse's position is deciding what value a bit duration has (Chapter 2.2.3). By showing a number of bit durations, you can start interpreting the actual data given that you know the position of the “start of communication” pulse.

3.4.4. Load modulation reading

The load modulated signal is more difficult to display, mainly because it's weak compared to the carrier signal. It's however possible to minimize the effect of this by just looking at one side lobe of the signal. By adjusting the vertical position of the oscilloscope so that the average peak value appears around the middle of the screen and adjusting the vertical resolution to the finest level (20mV on the scope I used), you are creating the best conditions for reading out '0's and '1's. This is used to test the reading distance for load modulation. When the difference between high and low amplitude gets below approximately 15mV , the signal gets hard to read.

3.4.5. Eavesdropping range set-up

The goal of this test is to draw a rough radiation diagram showing how the possible eavesdropping distance is changing according to the antennas position relative to the reader. This is done by placing the antenna in 17 different positions and adjusting the distance so that the load modulated signal is visually readable as indicated in the end of Section 3.4.4. The different positions as explained in detail in Section 3.5.4.

3.4.6. Oscilloscope configuration

To perform the tests as described in this thesis, you need a digital storage oscilloscope with pulse triggering opportunities. For each test with specific distances between the reader, tag and test antenna, the average signal level has to be measured. This is done by setting the scope to trigger on an event that is always happening, for instance positive edge triggering with low amplitude trigger level. In the measurement menu, set the oscilloscope to measure the average peak-to-peak value. To catch the first ASK pulse, set the oscilloscope to trig on pulses with negative polarity, trig level about $\frac{1}{4}$ of the measured peak-to-peak value and pulse width larger than $2\mu\text{s}$. While using “Single-Sequence-Triggering”, this set-up will always trig on the first ASK pulse. As the load modulated signal doesn't have any properties that don't match an ASK pulse, there is no possibility to trig on the load modulated signal. To overcome this, the storage capabilities of the oscilloscope are exploited. By adjusting the horizontal position knob, you can choose which part of the signal you want to look at. The two first frames, ALL_REQ and SENS_RES, are easy to find. The start of ALL_REQ is off course the pulse that the oscilloscope triggers on. The frame response time for a target after receiving both SENS_REQ and ALL_REQ is defined so that the targets “Start” always is transmitted $162,8\mu\text{s}$ after the first pause in the “Short frame” sent from the reader. This can easily be controlled by reading out the end of the last pulse in the ALL_REQ signal and adding $91,15\mu\text{s} ((9*128 + 84) / f_c)$ because the last pulsed bit in the ALL_REQ is a “1” (Chapter 2.2.6). The following commands have no exact starting point, except minimum values. The only way to find them with the oscilloscope is to adjust the horizontal position some bits at a time until you find another modulated sequence.

When working with interpreting the signals, there are two ways to capture a complete command. The quickest one is to set the horizontal scale to a value so large that the oscilloscope screen displays all the bits from the start delimiter to the end

delimiter. For instance, SENS_RES has a total width of just under 200 μ s which should give a scale of 20 μ s per div. To be able to read out the bits, you then have to use the window function in the horizontal menu to zoom in on smaller parts of the signal. 10 μ s per div is a good scale when trying to interpret the modulated signal, as every square on the screen (div) has almost the same width as one bit duration (9,44 μ s). By using this setting you can display 10 bits, or bit durations per screenshot. Using the window/zoom-function, does however give a less accurate reading of the signal, than by just capturing 100 μ s at a time. For the ASK signal this doesn't matter much, but when trying to maximize the reading distance of a load modulated signal the difference is substantial. The second and most accurate way of reading out the signal is to run a new test sequence for every 10 bits, and capture the signal piece by piece. As the test sequence is repetitive, the bits appear in the same position every time. To minimize the possibility of errors, I chose to interpret 5 bits at a time. By adding 5 bit durations to the horizontal position for each capture, the left half of the screen shows the last 5 bits that I read and the right half shows 5 new bits. This method is time consuming, though very reliable.

3.5. Test results

The test results will be presented step by step according to the test plan. One screenshot from the oscilloscope is presented for each test. In cases where the complete result consists of a number of screenshots, you can find the full sequence in Appendix A. All test result descriptions, except the first, assumes that the procedure and findings of the previous test(s) are known.

3.5.1. Signal verification

The purpose of this test was to be able to display a signal and convert the modulated carrier to bit values. By enabling pulse triggering on the oscilloscope as described in Section 3.4.6, the oscilloscope reacts on every pause in the signal. Figure 27 shows how an arbitrary ASK-sequence looks like on the oscilloscope.

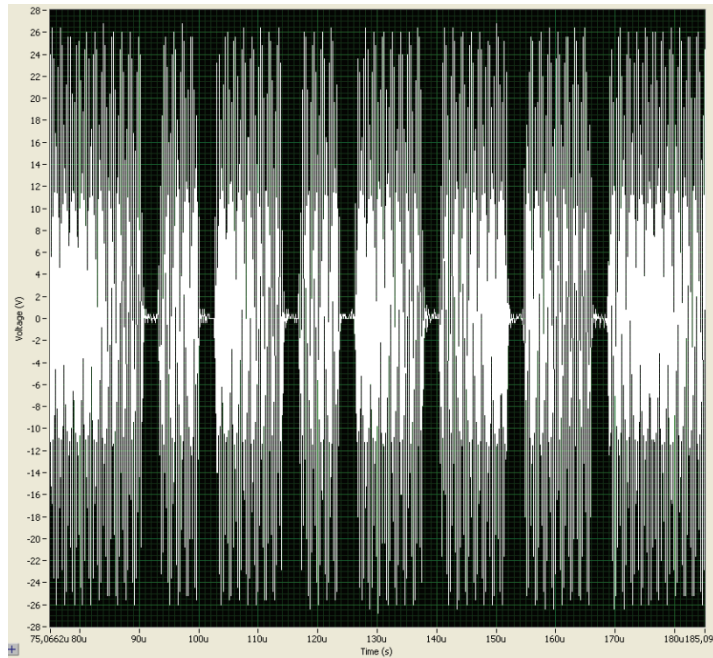


Figure 27: Example of an ASK modulation sequence.

The interpretation of the signal is performed by employing the pulse position definitions in Section 2.2.3, which is illustrated in Figure 28.

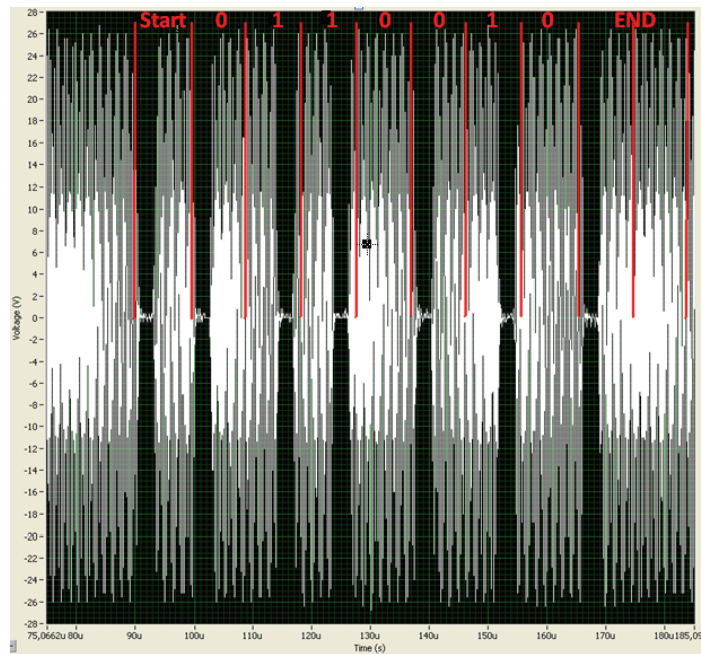


Figure 28: Bit interpretation of a 100% ASK sequence.

To get a reading of the load modulated signal, some more advanced triggering has to be performed as described in Section 3.4.6. A load modulation sequence will look like the example in Figure 29.

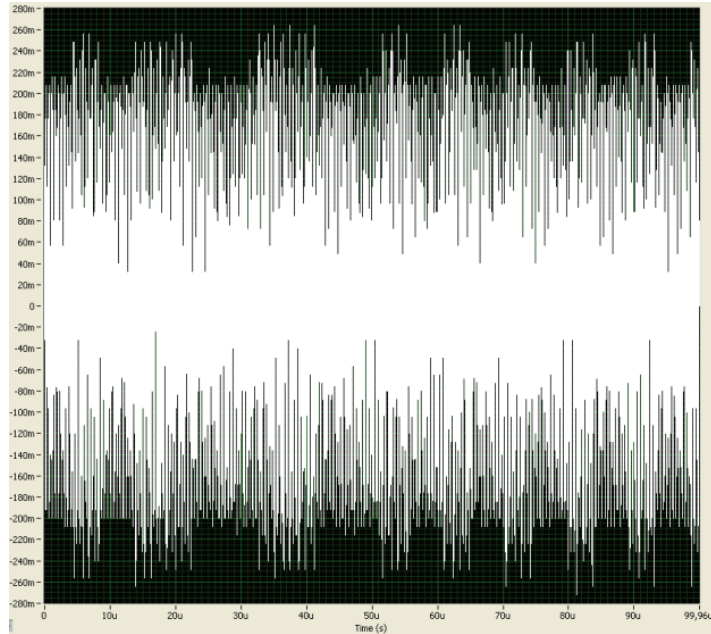


Figure 29: Example of a load modulation sequence.

When modifying this signal by employing the Manchester encoding described in 2.2.3, you get the result in Figure 30.

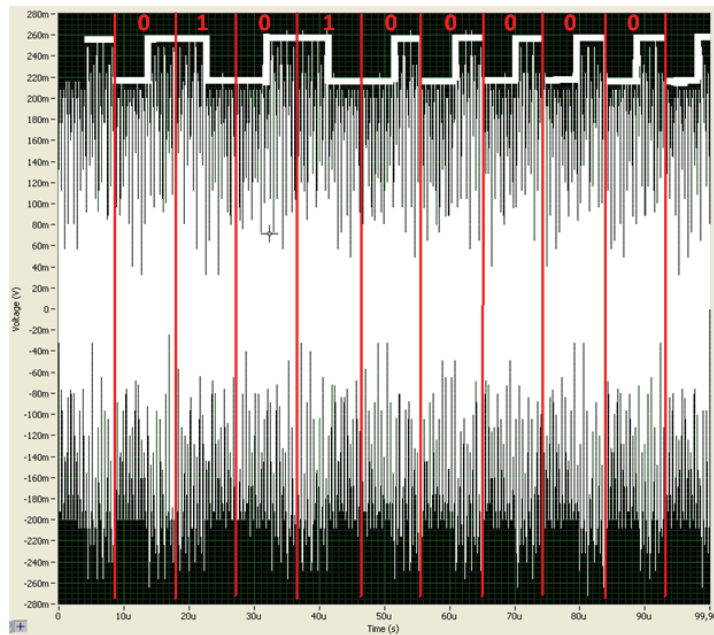


Figure 30: Bit interpretation of a load modulation sequence.

The load modulation of the NOKIA 6212 classic was a bit harder to display than the other tags, although it was put in card emulation mode. The reason for this was that it kept on generating its own RF-field regardless of the communication mode. It only switched off the field when detecting another field. Because of this, it was nearly impossible to make the oscilloscope to trigger on the right events. I did however manage to capture a signal after some really tedious experimenting. I had to activate the oscilloscope in a small time interval while the phone was preparing for power saving mode. In a short period of time, the emulated card is then available after the phone's RF field has been switched off. As the reader is NFC enabled, the phone answers with 10% ASK modulation employing Manchester bit encoding with reverse amplitude as shown in Figure 31. This means that the bit interpretation is inverted compared to Figure 30.

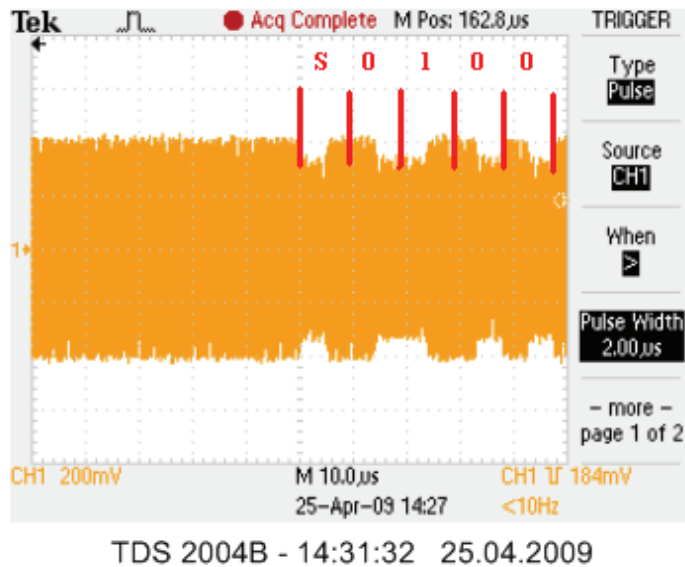


Figure 31: Load modulation from NOKIA 6212 classic.

Because of the trigger difficulties with the NOKIA phone in card emulation mode, the rest of the tests are performed with tags.

3.5.2. Command recognition

When using the “NFC Test Program”, the first command is always an ALL_REQ. As this is transferred in a short frame, it has a total duration of 10 bit durations including start and stop delimiters. Figure 32 shows how the ALL_REQ signal looks like when adjusting the horizontal position so that the whole signal is displayed on the screen.

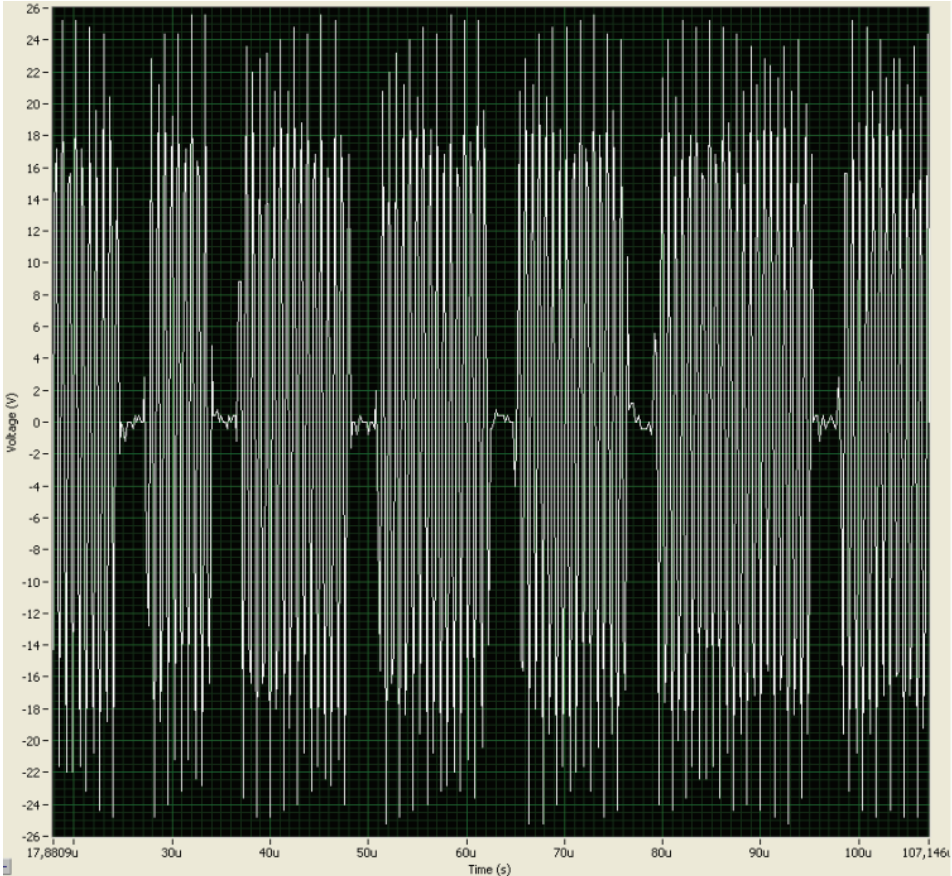


Figure 32: Capture of an ALL_REQ command.

In Figure 33, the screenshot has been modified to show the bit durations and each bit value.

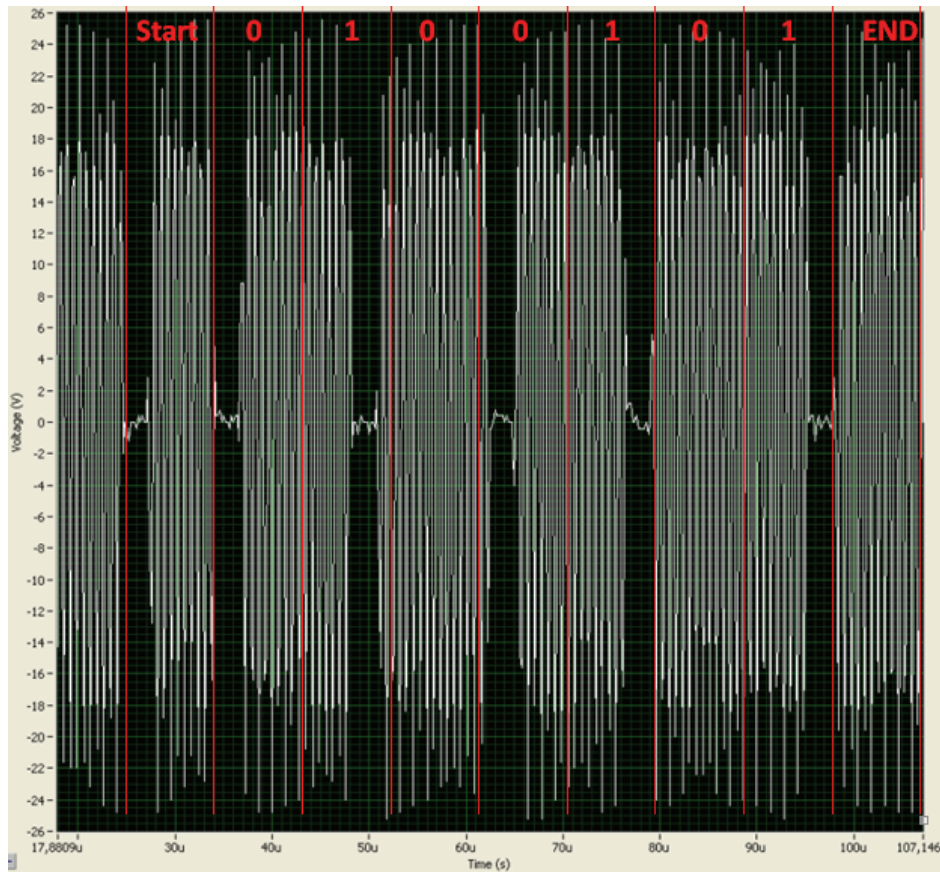


Figure 33: Interpreted bits of an ALL_REQ command.

As the command is transmitted LSB-first, the leftmost bit is bit 0 and so on. Rearranging the bits, you get the command “1010010”, which matches with the specification of ALL_REQ described in Section 2.2.6.

To find the first response from the target, the horizontal position has to be set to 162,8 μ s. As this command is longer it has to be captured in steps, where the first part is shown in Figure 34.

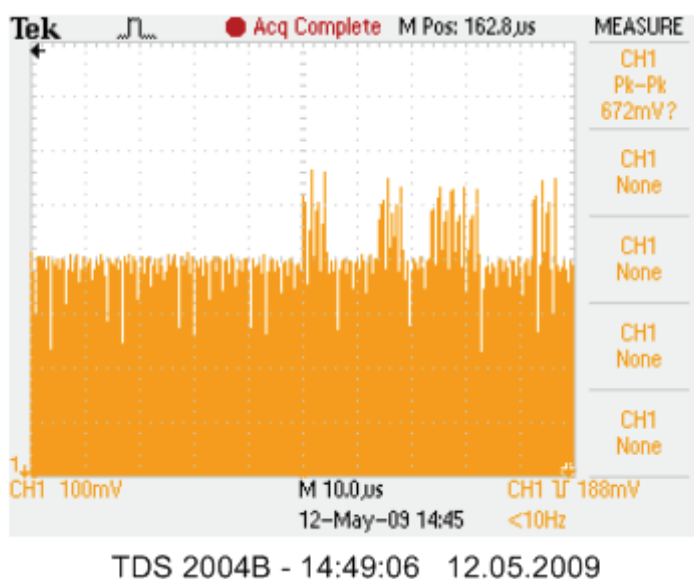


Figure 34: Start of SENS_RES from Target.

The bit positions and bit values for the same sequence is shown in Figure 35.

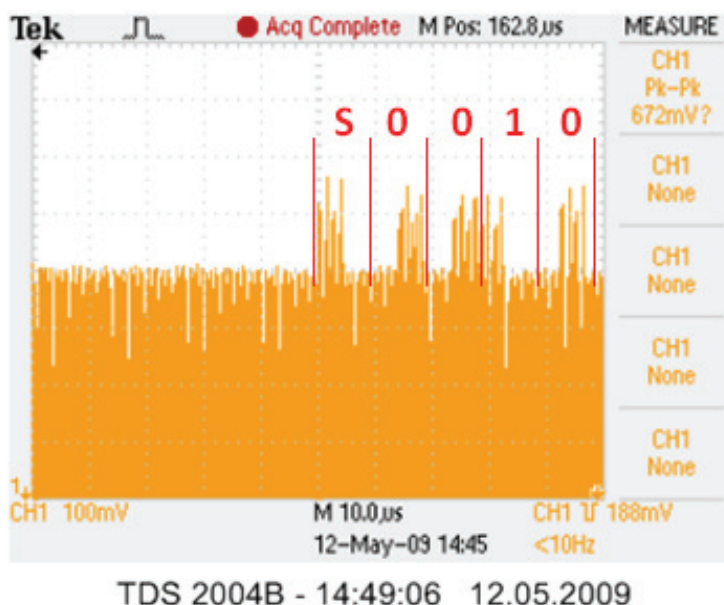


Figure 35: Bit interpretation of the first 5 bit durations of SENS_RES.

The whole interpreted sequence is shown in Table 20, start and end delimiters are omitted. This command is transmitted in a standard frame where the information is sent byte by byte, with LSB first within each byte, and one parity bit after every byte. Applying this to the captured sequence, you get the SENS_RES as shown Table 21 which also matches with the specifications described in Section 2.2.6. The bits 7 and 6

indicate that this device has single NFCID1 size, which means that the UID is 40 bits long. The device has also set bit b2 to '1' for bit frame SDD purposes.

Bd	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Byte	1									2								
Bit	0	1	2	3	4	5	6	7	P	0	1	2	3	4	5	6	7	P
Value	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 20: All transmitted bits of a SENS_RES from Target.

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Table 21: SENS_RES command

3.5.3. Communication sequence recognition

This is the most tedious task of the experiment. When executing a test sequence in the Test Program, the reader starts a communication initiation sequence with the tag. The number of commands needed is dependent of the tag type. For NFC and ISO/IEC 14443 Type A tags with single length UID (40 bits), this sequence consists of 4 commands:

- 1) Reader → Tag: ALL_REQ
- 2) Tag → Reader: SENS_RES
- 3) Reader → Tag: SEL_REQ
- 4) Tag → Reader: SEL_RES + CRC

The two first commands are easy to find in the time domain, as explained in Section 3.4.6. The FRT after the SENS-RES is however not that evident. Recall from Section 2.2.6, that the FRT for the reader is only specified by a minimum value. To find the third command, it is therefore necessary to start with the minimum value since the last pulsed bit, and then capture screen by screen until you find the signal. Using the Test Program I found that every time the test sequence was run, command 3 appeared 1318μs after the very first pulse. The FRT between command 3 and 4 is possible to calculate by the same formula used to find the start of SENS_RES. It should be 86,43μs or 91,15μs dependent on the last pulsed bit (Section 2.2.6). For the tags I used, this command appeared 2182μs after the very first pulse. The start of each command is shown in Figure 36 to Figure 39.

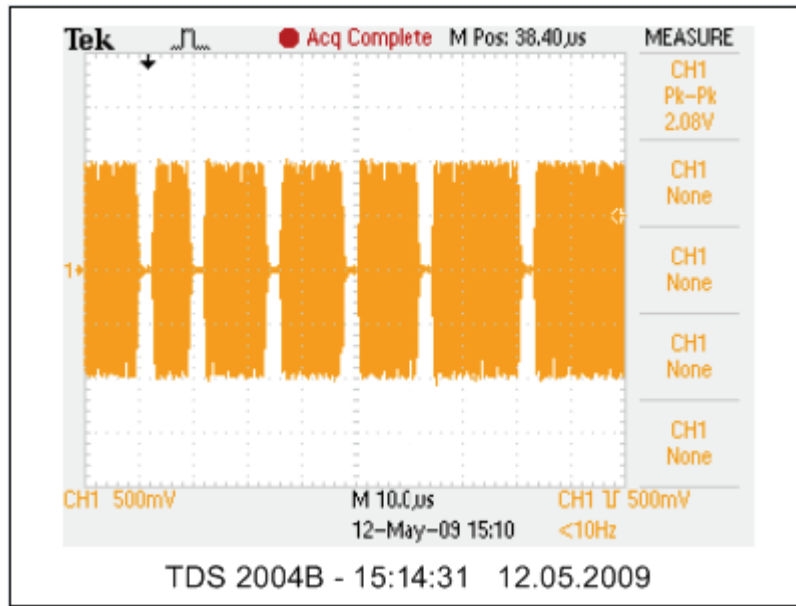


Figure 36: Start of ALL_REQ

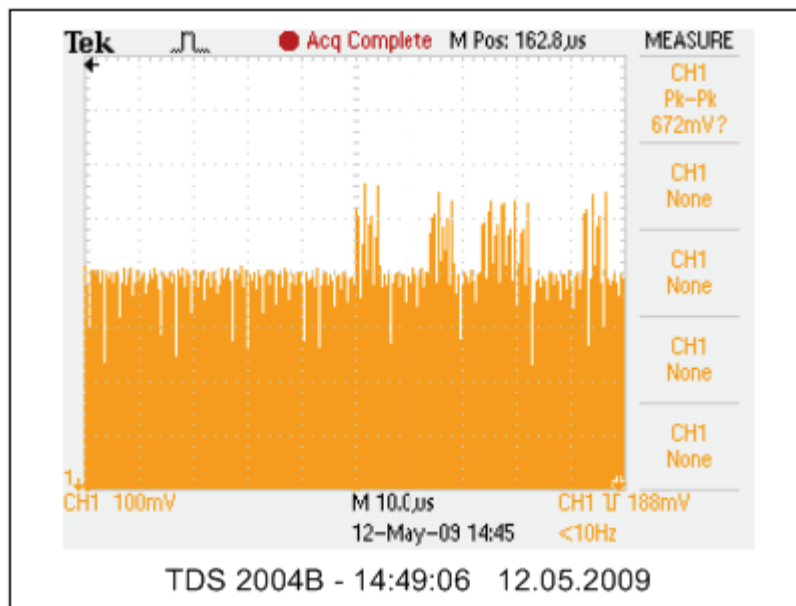


Figure 37: Start of SENS_RES

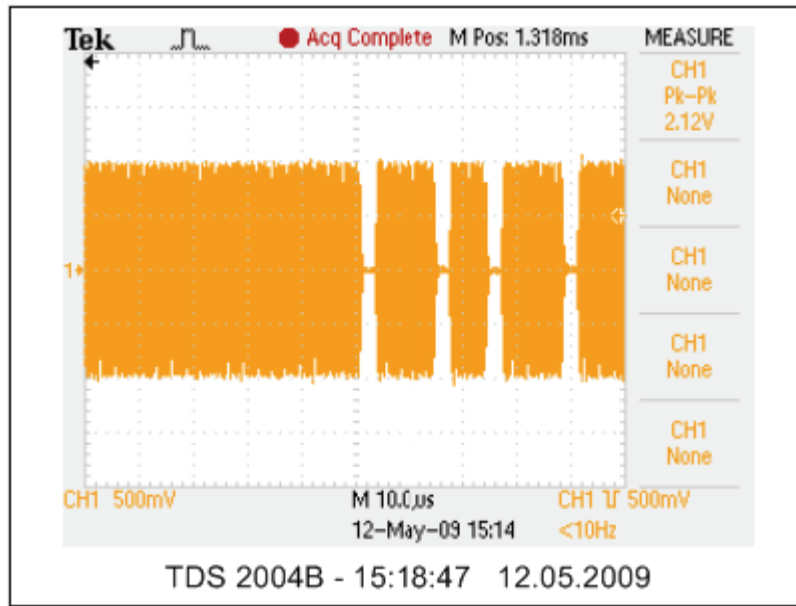


Figure 38: Start of SEL_REQ

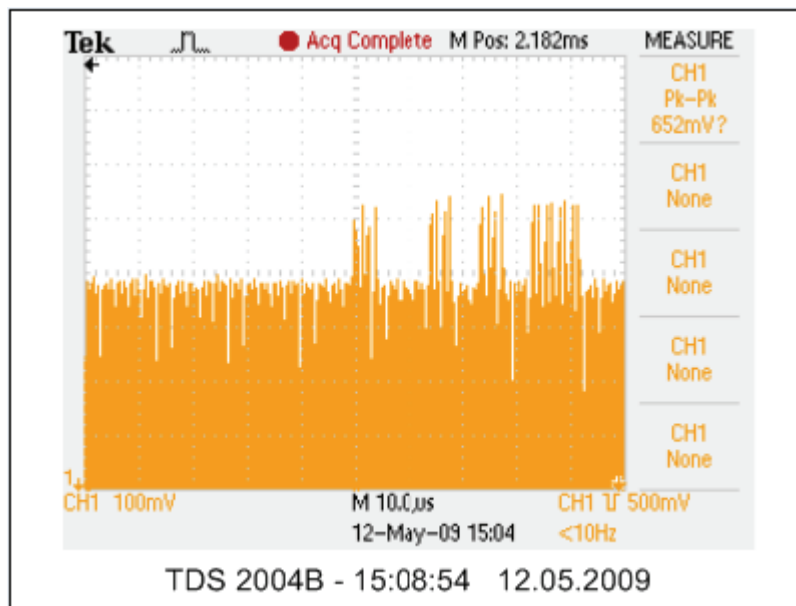


Figure 39: Start of SEL_RES

After demodulating each command by turning the bit order and removing parity bits, start delimiters and end delimiters, I got the results in Table 22.

ALL_REQ	1010010				
SENS_RES	0000 0000 00 0 00100				
SEL_REQ	93	70	0C CF BD 40	3E	2A CC
	SEL_CMD	SEL_PAR	NFCID	BCC	CRC
SEL_RES	08		B6	DD	
	SEL_RES		CRC		

Table 22: Results of capturing a whole communication sequence from Initiator to Target

To verify the UID of the tag, I used the ACR 122U Tool-software to read out the UID and the two values matches. The BCC byte can be verified by performing and XOR on the 4 previous bytes, in this case the whole UID of the tag. As you can see in table, XOR'ing the four UID bytes gives the same result as the interpreted value of the BCC Byte (Byte 7).

Byte	Value							
0C	0	0	0	0	1	1	0	0
CF	1	1	0	0	1	1	1	1
BD	1	0	1	1	1	1	0	1
40	0	1	0	0	0	0	0	0
XOR'ed	0	0	1	1	1	1	1	0
BCC	0	0	1	1	1	1	1	0

Table 23: XOR of the last four data bytes compared to the interpreted BCC in a SEL_REQ.

To verify the two CRC bytes of SEL_RES, I used the CRC Calculation Program as shown in figure. The input data must be lsb first without parity bits, and the CRC is also lsb first. The program takes maximum two bytes as input, so the CRC for SEL_REQ is not computed.

```
C:\Documents and Settings\bruker>CRCCalculationProgram.exe
Enter bitstring to calculate (lsb first):00010000
CRC of [00010000] is [0110110110111011] with P=[10001000000100001]
```

Figure 40: CRC calculation of the SEL_RES command.

Data input lsb first	00010000
Calculated CRC lsb first	0110110110111011
“B6 DD” in reversed order	0110110110111011

Table 24: Comparison of interpreted and calculated CRC in SEL_RES.

Table 24 shows that the CRC value calculated over all data bits in the command are the same as the value of the interpreted CRC bytes. Using a tag with longer UID will increase this sequence by one or two more command→response pairs.

3.5.4. Maximum eavesdropping ranges

As it is now proven that the signal can be picked up and visually interpreted using an improvised antenna and an oscilloscope, the final task was to measure the ranges in which this is possible using pure passive methods. The first test was to find the more effective of the two improvised antennas. This test is performed with the antenna placed in horizontal position directly above the NFC-mark on the reader, as illustrated in Figure 41. The test results are shown in Table 25.

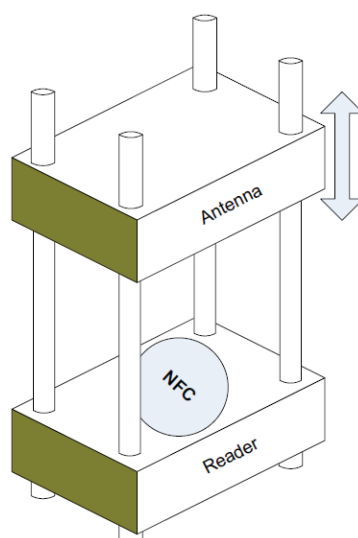


Figure 41: Antenna adjustment rack

Antenna	Eavesdropping distance
Mifare UltraLight Label	17,4 cm
Mifare Classic Card	29,2 cm

Table 25: Eavesdropping ranges for the available antennas.

As the MIFARE Classic card gave the best readings, I used this to perform the rest of the test. The goal was to draw a rough eavesdropping diagram of the communicating devices' RF-field. This is done in two steps, one with the antenna always in horizontal position and one with the antenna placed perpendicular to the magnetic field. Both tests are performed in three levels, right above the reader, horizontally aligned with the reader and at 45 degrees as illustrated in Figure 42.

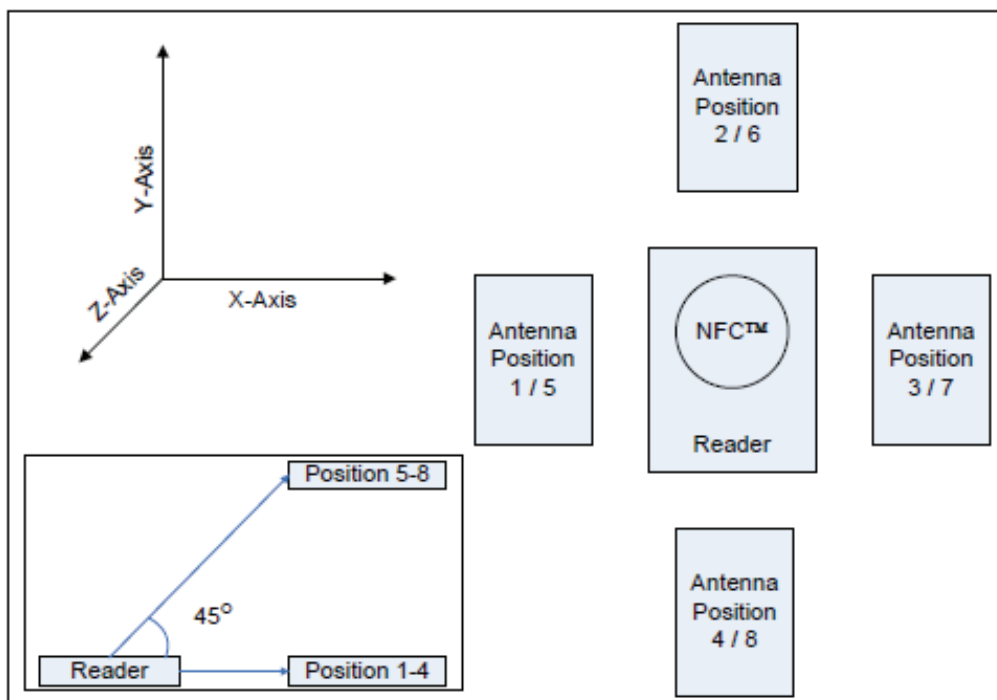


Figure 42: Antenna positions in maximum eavesdropping range test.

The distance is increased until the signal is no longer visible, and then slowly decreased until the signal becomes visible again. In the limit area, there seems to be no graded decrease. Either the modulation is visually interpretable, or it's too weak to be seen at all. The limit seems to be around 15mV above the carrier signal. The results are shown in Table 26.

		Antenna position							
		1	2	3	4	5	6	7	8
Range in cm	Antenna angle								
	Leveled	22,3	20,6	21,7	23,2	19,1	16,7	19,6	20,3
	Perpendicular	19,1	20,1	19,6	22,9	23,5	24,4	23,2	25,2

Table 26: Maximum eavesdropping range test results.

To give a visual interpretation of the test results, eavesdropping distance diagrams are presented in Figure 43 to Figure 48.

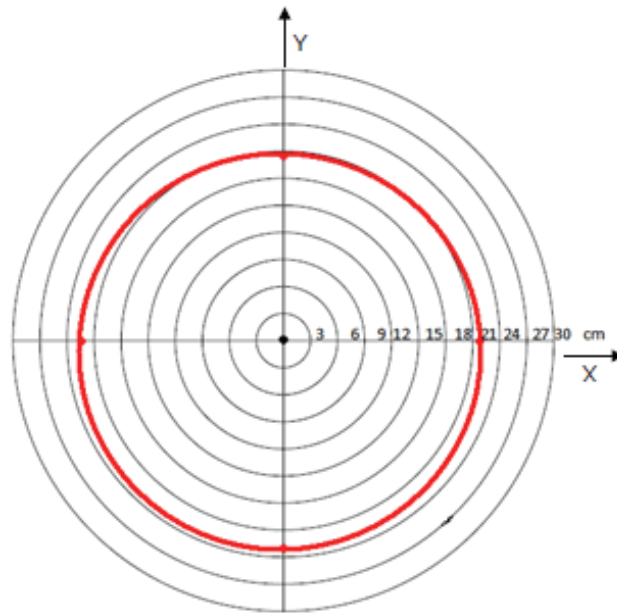


Figure 43: Eavesdropping diagram for horizontal antenna in the X-Y plane with $Z=0$.

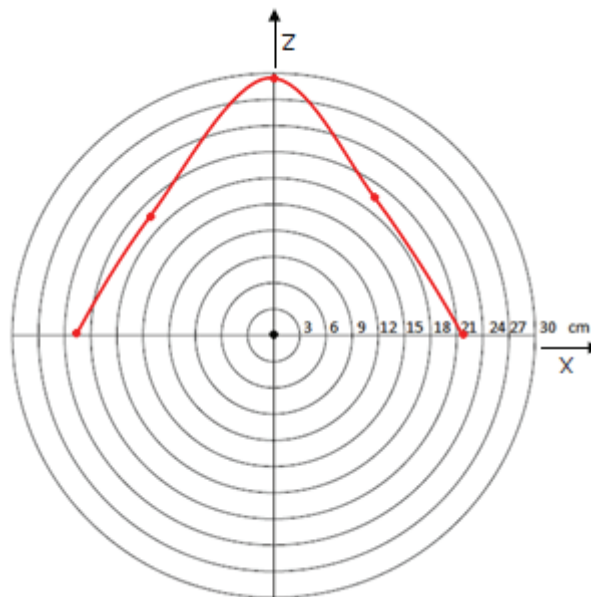


Figure 44: Eavesdropping diagram for horizontal antenna in the X-Z plane.

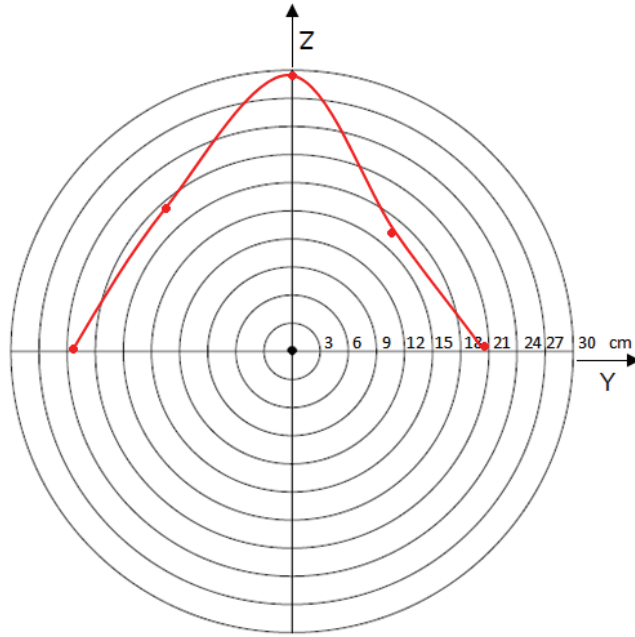


Figure 45: Eavesdropping diagram for horizontal antenna in the Y-Z plane.

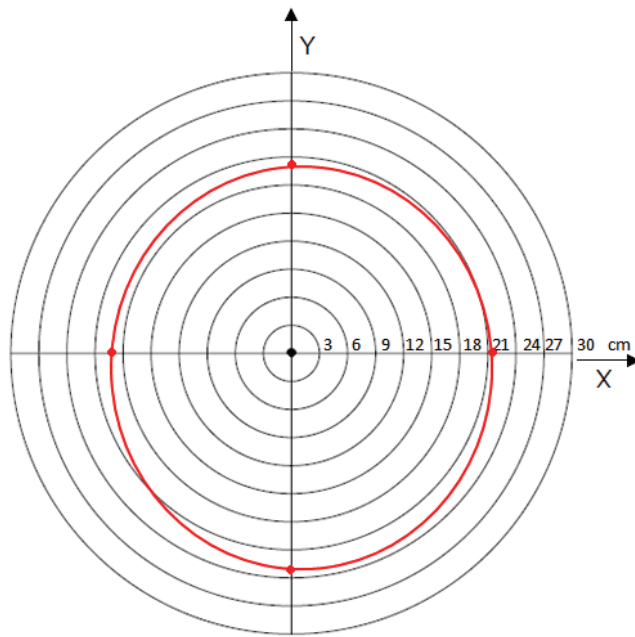


Figure 46: Eavesdropping diagram for perpendicular antenna in the X-Y plane with $Z=0$.

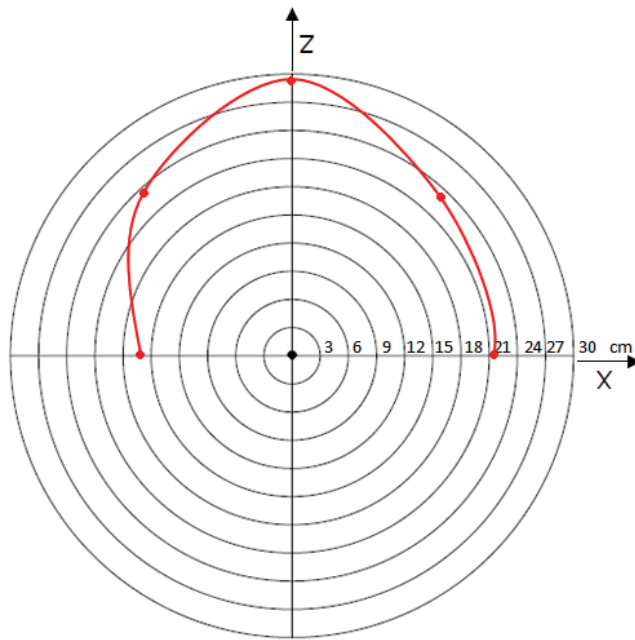


Figure 47: Eavesdropping diagram for perpendicular antenna in the X-Z plane.

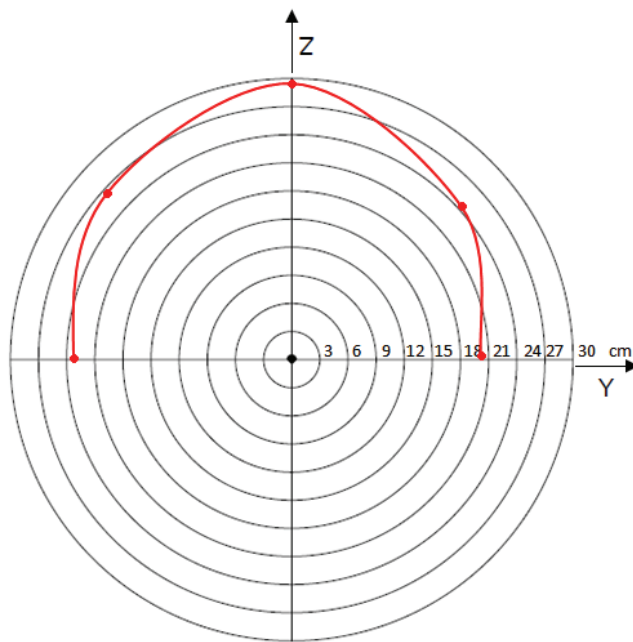


Figure 48: Eavesdropping diagram for perpendicular antenna in the Y-Z plane.

As the antennas are implemented as rectangular loops, they have a complex radiation diagram. The tests in this section are performed in a limited number of test positions, to prove the concept and give an indication of the emitted field. A complete eavesdropping diagram should be done in a three dimensional manner with a lot more

reference points. To perform this with a certain level of accuracy, one would need more sophisticated equipment for stable DUT positioning and distance measurement.

3.6. Test result discussion

The test results prove the hypothesis made early in the thesis, namely that it is easy to eavesdrop NFC-communication. This means that any transmitted bit sequence can be picked up by an antenna placed within the possible reading range. As the procedure can be done visually using passive equipment only, it should be easy to develop radio receivers performing the same task in a much more effective way. It should be possible to capture a whole communication sequence in one piece, and the eavesdropping ranges should increase significantly using the proper circuitry.

Although I had some problem with reading the communication sent from the NOKIA 6212 Classic, this was related to limitations of the oscilloscope triggering only. A radio receiver made for this type of communication would not have this problem, as it doesn't need all RF-fields to be switched off before making ready for signal reception.

It is important to notice that the tests performed here have a certain fault margin, as all tests are based on my opinion of what is possible to read visually. Although this might give an uncertainty on 1-2 cm on each measured value, it does however not influence the fact that the channel can be eavesdropped in ranges up to 30 cm with passive equipment. The environmental variables have neither been considered. All tests have been performed in a computer lab housing a lot of different electronic equipment, such as wireless access points, desktops, laptops and mobile phones. This means that the radio environment may be quite complex. To get optimal conditions, these tests could have been performed inside a protected environment, such as a Faraday cage. Alternatively, the background noise could have been measured to get an idea of the interference from the surroundings. Because of frequent changes in the total amount of equipment in use in the room, such a measurement would probably not be accurate for any longer period. For the concept of this task it was far more important to prove the possibility than making 100% accurate measurements.

4. SECURITY SOLUTION

In this chapter, I will present a security framework based on theoretical knowledge about NFC, information security in general and the experimental results presented in Chapter 3.5. This thesis focuses on applications dealing with sensitive information, such as payment solutions and ATMs.

4.1. Threats

4.1.1. General

As indicated in Chapter 2 and proved in Chapter 3, it is possible to eavesdrop on information sent between NFC devices. Both my test results and the experiment by Gerhard Hancke referred to in Chapter 2.3.2 shows that both modulation technique and device design influence the reading range of eavesdropping. The type of application is however the most significant thing to consider when implementing a security system. It will always be a compromise between cost and protection. For single applications this is relatively easy, as the decisions don't affect other systems. NFC is however meant to be a platform supporting multiple applications, which introduces a complete different set of considerations. The weakness of one application might be exploited to open a backdoor into another application. The success of NFC as an electronic wallet is strictly dependent on trust amongst the users.

4.1.2. ASK

The fact that ASK has significantly greater reading range than the load modulated signal makes it extra vulnerable to malicious activity. This means that applications using active communication mode or transmitting sensitive data from an active device to a passive device has to take this into account when evaluating the security of the application. When the range is several meters it's possible to perform attacks with far more sophisticated equipment, as the actual size can be heavily increased without seeming suspicious. It might be hidden in briefcases, vehicles, mobile booths and so on. It is also harder to prove who performed an attack, unless you catch them on site with the used equipment. When a skimmer is placed within decimeters of an ATM, all activity is usually picked up by surveillance cameras and the incidents can be documented.

4.1.3. Load modulation

Although the reading range of a load modulated signal is pretty short, it's nevertheless possible to eavesdrop. If it's possible to implement a skimmer as a sticker with the size of the MIFARE-labels or the MIFARE-classic card, a malicious device might be very hard to detect. Detecting who is placing or removing such a device might also be very difficult.

4.1.4. Tag content

Though this thesis focuses mainly on applications where valuable information is involved, I also want to point out some issues regarding typical "low security"-applications such as public advertisement. When an NFC device is put into reading mode, it will react to tag content dependent of the user settings. On the Nokia 6212 Classic used in my experiment, the default settings is that tag content shall be put in an inbox, but not executed without confirmation. The confirmation check may however be disabled. When reading a tag that is supposed to redirect the phone to a web-page, most people will probably neither check the actual URL very carefully. By experimenting with fake tags or tampering with tag content, it possible to make phone download malicious or fake content. Exploiting these properties, attackers may be enabled to distribute code which is interfering with more sophisticated applications. If such attacks are successful it might be possible to retrieve information like card details or key information from the phone.

4.2. Countermeasures

While working with the security model, literature regarding UMTS [38] and RSN [40] has been used together with general cryptography and network security literature [5].

4.2.1. General

To make NFC a successful application dealing with both high and low security applications, a leveled security system with a set of minimum-requirements should be implemented. A strong mutual authentication protocol could solve the problem of fake tags and tag content. This requires however that the two parties have some pre shared information, or that they can communicate with an authentication server. Possible solutions to this are authentication by symmetric ciphers or digital certificates. If an

authentication protocol is to be obligatory for all NFC implementations, it has to be cheap, fast and secure. It also must be able to operate with strict resource limitations.

With authentication implemented, the basic security might be covered. The possibility of altering bits in live a communication sequence is however not dealt with in this thesis, but should nevertheless be given some thought. If it is possible to alter bits in such a way that it will appear to a valid command at the receiving end, a MIC (message integrity code) should be introduced. This must then be implemented in such a way that it can't be changed so that it matches altered bits in the actual data.

The main device of the NFC technology is the NFC enabled mobile phone. Although such phones are going through constant improvement when it comes to storage capacity and processor speed, the size of applications will always be an issue. By keeping the needed data download per application at a minimum, the user will get faster and easier access to new applications. If a common security framework for all applications is developed, each application developer can focus on the functionality and just add application specific key parameters in the download.

4.2.2. Authentication solutions

The first level of the authentication solution should protect against fake tags and writing of malicious content to tags placed in public areas. This solution should be performed by unilateral authentication dependent on the function. When writing content to a tag, the writing device should be authenticated by the tag to open the write protection of the internal memory. This will complicate the process of tampering with the tag content. When a tag has been deployed, it should be possible to authenticate the tag to ensure that the original tag hasn't been swapped with a fake one. The biggest challenge for this low level solution is to make it as fast, easy and cheap as possible and still offer sufficient security. The definition of sufficient security is another question. The effort put into breaking into a system depends on the possible gain. If no valuable information is involved, the typical attacker will be little knowledge hackers wanting to cause disruption of services or seeking approval. If however it proves possible to gain valuable information from an NFC phone indirectly by first exploiting the low security of public advertisement tags, more sophisticated attackers with far greater resources will get attracted. Evaluating this threat must be a continuous work to adapt to the technological development. To make adjustments as easy as possible, highly acknowledged algorithms with flexibility in key length should be used.

4.2.3. Encryption solutions

To offer easy implementation of applications transferring sensitive data, NFC should offer an encryption framework. If a common standard is developed, proper hardware can be deployed while manufacturing NFC-phones. Hardware encryption is faster than software encryption and the applications will be easier to develop.

To offer sufficient security for a reasonable amount of time, symmetric encryption like AES with at least 128 bit security should be used. Extension to 192 and 256 bit key should also be possible. According to NIST and ECRYPT, 128 bit key for symmetric ciphers should offer good security for approximately 30 years if today's processing speed development carries on [26]. With asymmetric encryption as RSA the security should be above 3000 bit to offer similar protection. This level of security should be a goal for applications transferring static sensitive data such as account numbers, personal identification numbers or UIDs. On the other hand, there might be applications needing short time protection which can settle with shorter keys.

Both 3DES and AES128 are candidates in such a solution. As there are cryptographic processors supporting both algorithms, it might be a solution to implement both, maybe with one of them as default for applications developers that won't or can't make a choice of their own.

4.2.4. Replay protection solutions

Both plain text and encrypted packets may be resent by an attacker. If the content is static, such as an ID from a bus fare payment card, it would be possible for an attacker to pick up a message and replay it at a later time to travel "free of charge". Although authentication might seem as a solution to this problem, there is also a possibility that an attacker can capture the authentication sequence and work around this too. To avoid that an earlier message is reused, a replay protection scheme should be used.

Sequence numbers, time stamps or transaction keys for the encryption algorithm may be used for this purpose. A scheme that is fast and easy to implement is AES in combination with counter mode operation as illustrated in figure, which has good efficiency in both hardware and software implementations [5]. If the communicating devices agree on a 128 bit random number during the authentication procedure, this can be used as the initial counter value for a specific transaction. By doing this, none of the devices need to remember the state of any pair wise sequence number counter in between transactions. The counter value can be generated during a mutual

authentication procedure, or it can be transmitted to the user device as a data packet. As the counter is a random number and encrypted by the key before it is used for encryption, it can be sent in plain text without decreasing the security.

4.2.5. Message integrity solutions

When a symmetric cipher counter mode encryption solution is used, it is possible to extend the solution with a CBC-MAC to provide message integrity similar to CCMP used in RSN [40]. A MIC is computed over all data and header bytes, and then encrypted together with the data.

4.3. Proposed common security framework for NFC

Based on the previous chapters, I will now present a general security framework for NFC, supporting different levels of security requirements. A common standard may lead to a high level of compatibility, easy integration and easier application development. The goal is to attract both service providers and consumers in order to make the number of NFC-users increase rapidly.

4.3.1. Introduction

All message sequence diagrams shows a successful session, and assumes that any error during a procedure will cause an authentication failure and abort the session. The MSCs are in accordance with the ITU-T specification for such diagrams [42]. Every device taking part in the message exchange has a time axis, where message are sent and actions are performed. A message is indicated by an arrow which indicates sender and receiver, and has a descriptive name. If the message contains any data, the data parameter is added in the brackets. If more than one parameter is sent, they are separated by a comma. An action is indicated by a rectangle, and the text describes the action performed. An example is illustrated in Figure 49.

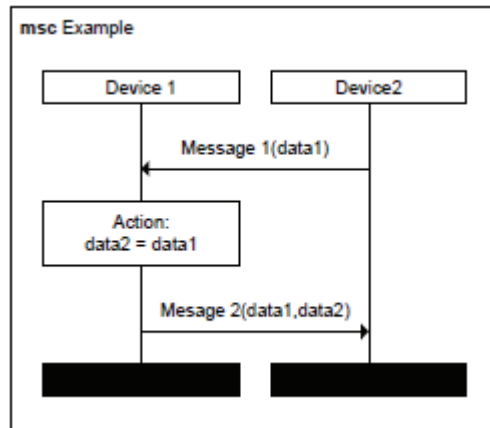


Figure 49: MSC example illustration.

4.3.2. General model

In order to support many different types of applications, the security regime must be flexible yet offering sufficient protection against expected threats. I suggest making a protocol divided in levels with increasing security, as illustrated in Figure 50.

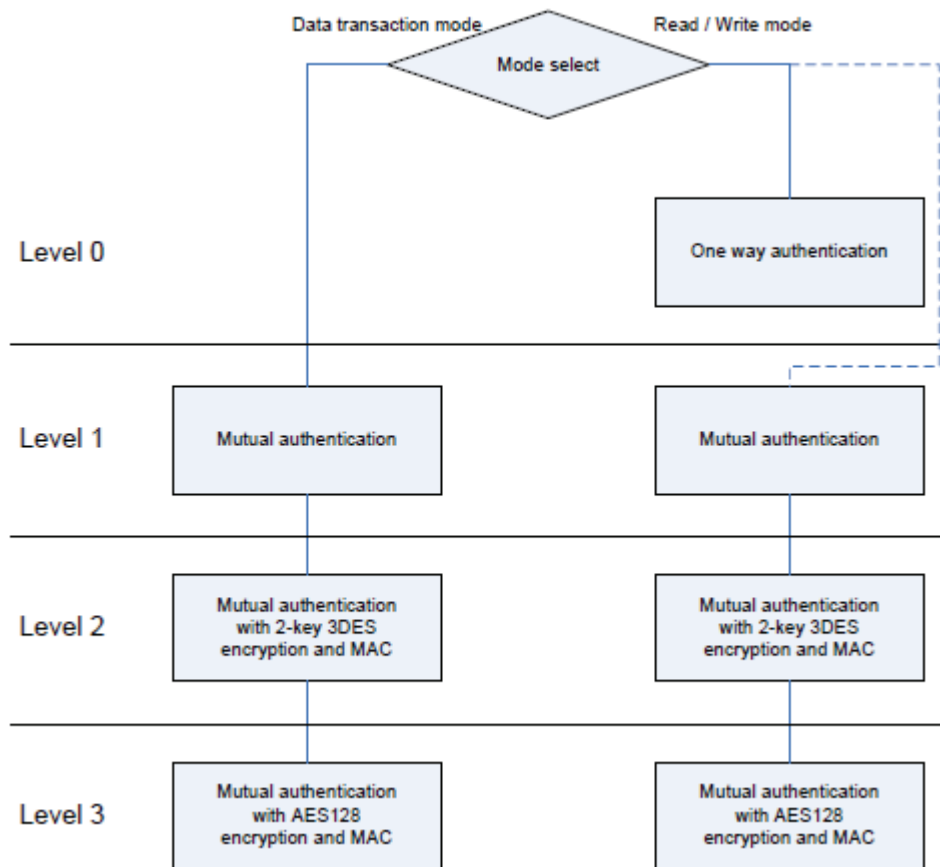


Figure 50: Security levels in the security framework.

4.3.3. Security level 0

The purpose of this level is to authenticate the tag providing information for a reader, or to authenticate the writing device if tag content is going to be changed. This level is typically applicable for advertisement tags or tags containing redirect information for simple vending machines. The unilateral two pass authentication is based on encryption by 3DES with a pre shared key (PSK). This implies that a distribution channel must be in place. This matter is dealt with in Section 4.3.7. Message sequence charts for the authentication procedures are shown in Figure 51 and Figure 52. The challenge should be generated using a PRNG, or at least a one way counter. The ID-parameter is a sequence number preventing the reuse of one particular challenge. This is counterfeit masquerade attempts by capturing, storing and replaying authentication information at a later time.

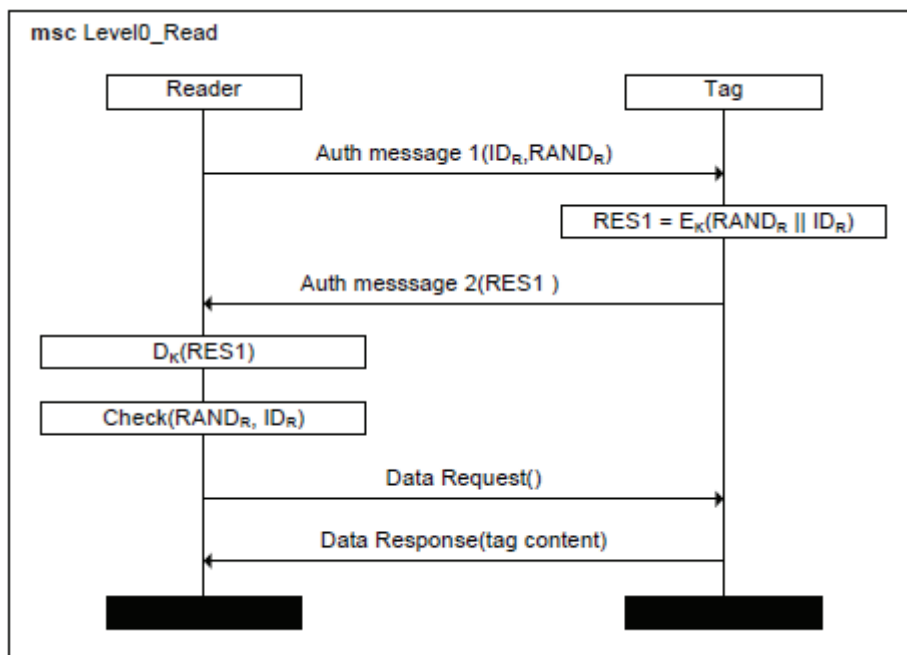


Figure 51: Message exchange in a tag authentication procedure.

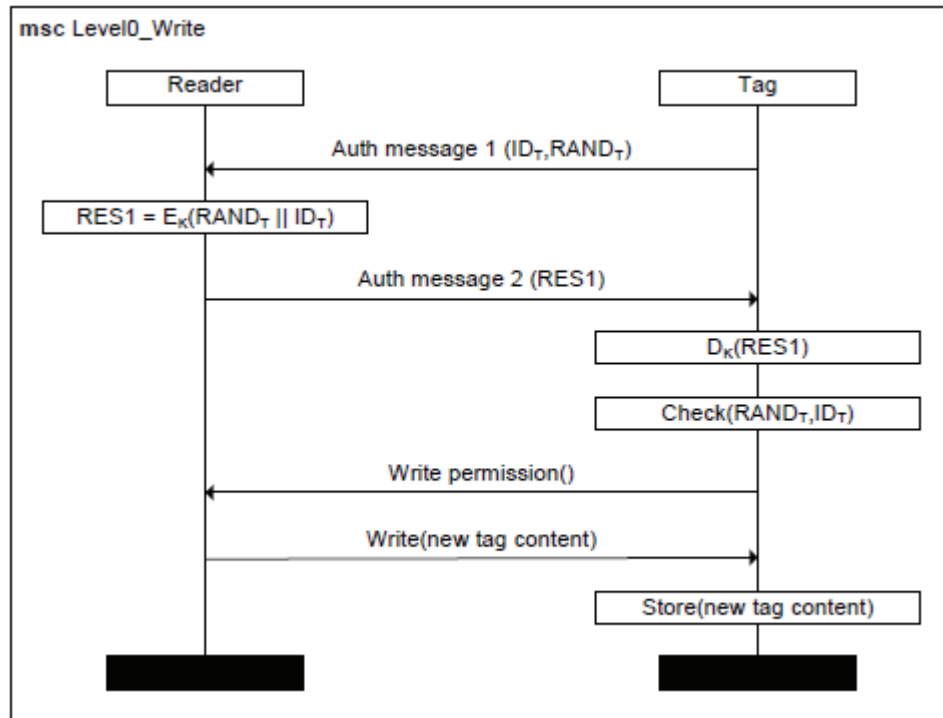


Figure 52: Message exchange in a reader authentication procedure

4.3.4. Security level 1

This level forms the basis for all applications performing duplex data transactions to ensure that none of the parties participate without proving its identity. This is solved by a three pass mutual authentication procedure, using generation and checking of random numbers to prove uniqueness. The ID-parameters are still sequence numbers related to the challenge to prove timeliness and prevent authentication token replay attacks. The procedure is illustrated in Figure 53.

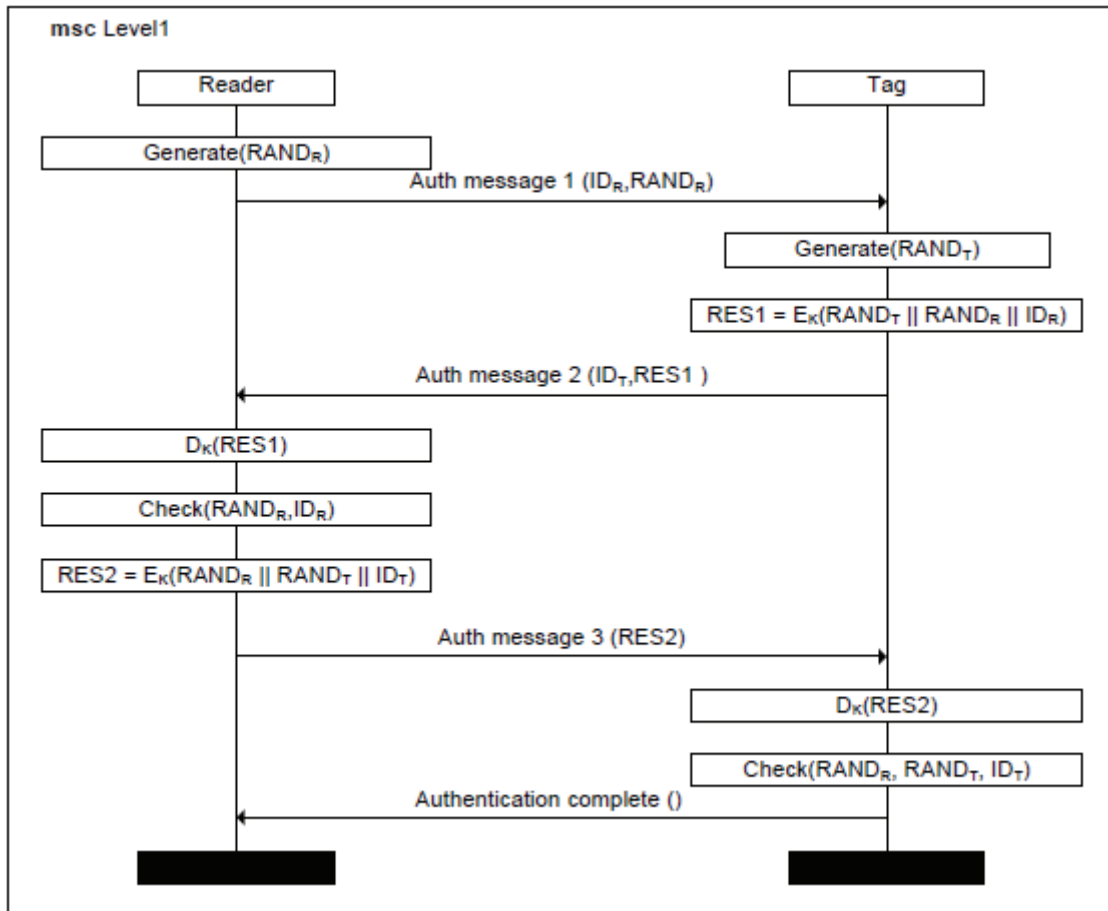


Figure 53: Message exchange in a mutual authentication procedure

4.3.5. Security level 2

This level is constructed for applications transmitting data that loses its value within a short time period after the actual transaction. This could be temporary ID, key or login information negotiation, and other parameter set-up applications.

The encryption technique is 2-key 3DES in CTR mode, concatenating 32 bits from each of the two challenges as counter initiation. The 3DES cipher is also used for the MIC computation, in CBC mode. By using random values from both parties, the possibility that one party can insert complete fake random values is eliminated. The complete confidentiality and integrity solution with replay protection is illustrated in Figure 54 and Figure 55. The block (B) size for 3DES is 64 bits, and this implementation uses two keys of 56 bits each. Key derivation is explained in Section 4.3.9.

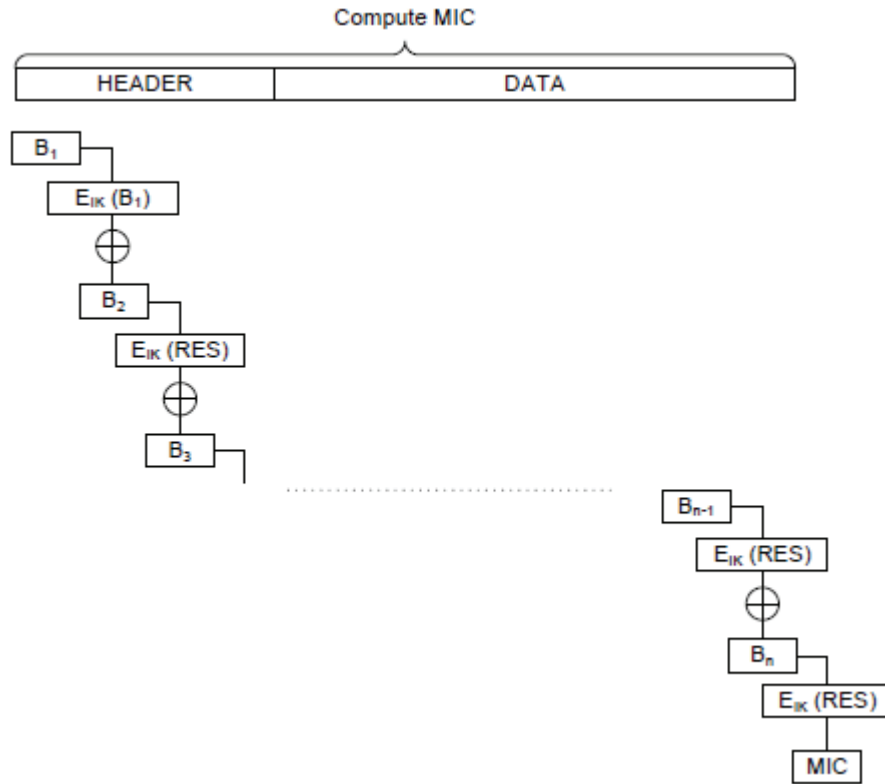


Figure 54: MIC computation.

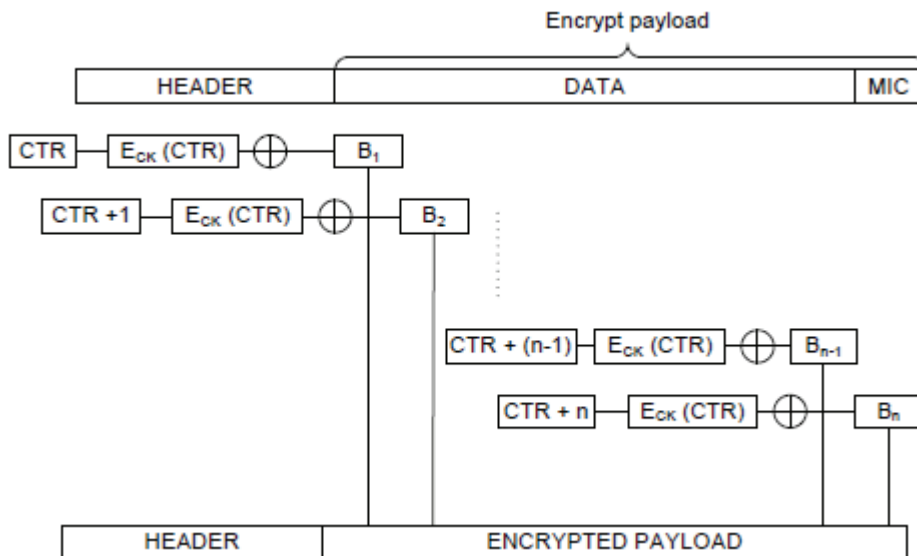


Figure 55: Payload encryption procedure.

4.3.6. Security level 3

The top level is constructed for long term protection of transmitted data, supporting applications dealing with personal account information, personal identification information and all other information with slow or no value degradation. The

encryption algorithm is AES128 in CTR mode, where the counter is initiated by a one way function of the two random numbers producing a 128 bit counter IV. Except from the cipher used, this procedure is the same as the one described for Level 2. The block (B) and key size is 128 bit, but the key can be expanded to 192 and 256 bit.

4.3.7. Key management

Key distribution is an important feature if this scheme is going to succeed. As at least one authentication key is required regardless of application, some action has to be taken in advance to make a service operative. Passive tags should be loaded with keys before deployment. The biggest challenge is however to distribute keys to the users. As NFC is intended to be used in mobile phones, there is already a distribution channel in place, used for operator purposes to send instructions to the USIM (or UICC in the future). As of today the mobile industry is working with a transfer protocol called “Over-The-Air programming” (OTA) [41], to perform high speed secure programming of users devices over the radio interface. This is ideal for downloading security parameters to applications installed on the phone. This solution is depended on that the user initially follows some simple instructions to download the necessary parameters for a specific service, for instance by sending an SMS to a distribution service with access to perform OTA transactions.

For applications using readers as their equipment a key management centre must be implemented, distributing keys to all devices in operation. There must be a service for distributing all current pair-wise PSKs and a method for key period transfer. This can be solved by introducing a “next key”-storage, or two indexed key storages with pair wise index negation in the communication initiation.

There should also be implemented a recommendation for key period evaluation. Dealing with long life data, the key period regulates how much information that is compromised if one single key is broken. When dealing with session dependent data, the key period regulates how much time an attacker has to succeed in breaking the key before he has to start over again.

4.3.8. Random number generation

To generate random numbers for the challenges in the authentication and key agreement procedure, a pseudo random number generator (PRNG) seeded by a time value and the key should be used [38]. An interesting recent finding is however that an

SRAM generates both a unique constant sequence and a completely random sequence when initialized, due to inevitable errors during production [39]. This might be a valuable feature when the price of SRAMS becomes feasible for universal NFC tag deployment.

4.3.9. Key hierarchy

In order to make this PSK scheme practical, there can be only one pair wise key between each application-user pair. Else the user equipment would have to store one key per terminal of one single application. This situation is not desirable, and it has two solutions. Transaction keys have to be derived for each communication sequence. If all application terminals are trusted, then the PSK can be distributed, with “in terminal” transaction key generation. Else the terminal has to fetch a pre generated key package from a central server for each communication set-up procedure.

To use PSK for a combination of authentication, confidentiality and integrity, separate keys is necessary. This can be solved either by distributing key triplets, or using key derivation functions with one PSK combined with additional session specific input. The functions should be one-way and non-linear. The three needed functions with suggested input could be like this:

- $AK = F_1(RAND_A || RAND_B)$
- $IK = F_2(RAND_A || RAND_B)$
- $CK = F_3(RAND_A || RAND_B)$

If the key derivation functions prove to be too complex for tags with limited resources, the pre shared triplet might be the only solution.

5. CONCLUSION AND FUTURE WORK

The experiments performed in this thesis were planned according to the assumption that it should be possible to catch and demodulate data sent between two NFC devices communicating in passive communication mode. The test results do indeed state that this is feasible, and that the needed knowledge to succeed is open and available. As the eavesdropping experiment was successful, there is neither any doubt that a security protocol is needed to protect any sensitive information sent between two NFC devices.

The test results show that NFC is vulnerable to eavesdropping if no security mechanisms are deployed. The signal that is load modulated on the RF carrier can be analyzed and demodulated visually at ranges up to 29 centimeters using completely passive equipment. The eavesdropping range is dependent on the implementation of the reader device, the tag device and the eavesdropping device. If a tag has a long reading range, it has an effective antenna which generates more effect when transmitting the load modulated signal. In other words devices with reading ranges in the area of 10 cm will be easier to eavesdrop than devices with ranges in the area of 2-3 cm. When it comes to the attacking antenna, it can be made more sophisticated than the ones used in this experiment implementing active circuitry in order to increase the signal reading range. For the passive communication mode it should be possible to pick up the signal up to a distance of one meter. For the active communication mode this distance is much higher.

As it is proved that the signal is possible to pick up in both communication directions, the need for security mechanisms becomes obvious. Because this communication platform shall support many types of applications, there should be a common security protocol ensuring that the low security applications and high security applications can coexist. To accomplish this, NFC needs a public security protocol with API documentation to be used by all application developers. During the design process each application should be able to choose a sufficient level of security, with a mandatory minimum requirement. This can be done by a layered model, with a set of levels offering increasing security. The lowest level should be the mandatory minimum and consist of unilateral authentication of a reader writing new data to a tag, or authentication of the tag when a reader device wants to read the tag content. For data transactions, mutual authentication should be the minimum requirement, with possibilities to increase security with encryption providing short term or long term protection. My solution suggests use of symmetric block ciphers in counter mode together with CBC MAC as this is

proven to be effective in both software and hardware implementations and is also effective within limited resource environments.

The two ECMA standards that were noticed just before submission of this thesis, is a confirmation of my stated need for a security protocol related to NFC. The specification of the data encryption is based on AES in counter mode, the very same encryption technique that used in my proposal. There are however two fundamental differences. First, my solution is based on pre shared keys while ECMA specifies a Diffie-Hellman Key Exchange protocol where the communicating devices don't share any common secret prior to the communication initiation [31]. Second, the ECMA standards describe the peer-to-peer mode while my solution is developed for reader mode and card emulation mode. So in a way the two solutions complement each other. As the peer-to-peer mode is a typical application for arbitrary content sharing between NFC phones, the key exchange approach seems like the best solution. Pre shared keys would not be a practical solution in this case, as there is no common server or central point of service.

5.1. Future work

Further work based on the results of this thesis may take several different directions. One possibility is to continue the analysis of the radio interface. A goal would be to develop a small battery driven radio receiver able to eavesdrop and store NFC communication sequences, i.e. a passive wireless NFC skimmer.

Another approach might be to continue with and improve the security framework. The solution presented in this thesis could be rewritten in a typical standardization document and merged with or made as an amendment to the ECMA standards to offer a complete common security standard for all types of NFC communication. A goal should be to make security implementation easy for the application developers. An API and a programming language library could also be developed. It might also be possible to analyze this solution in comparison with a PKI approach, to find the most effective solution. Terms to evaluate would be offered security, transaction time, key storage size compared to the available memory and processor capacity.

The properties of SRAMs are also an interesting theme for further enquiries. An interesting task would be to investigate how the "flaws" of SRAMs can be exploited in order to improve the security of NFC when it comes to keys, temporary key generation, IDs and random number generation.

REFERENCES

- [1] ISO/IEC, “*Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1)*”, 1.Edition. ISO/IEC 18092; 2004-04-01.
- [2] ISO/IEC, “*Identification cards – Contactless integrated circuit(s) cards – Proximity cards- Part 1: Physical characteristics*”, 2.Edition. ISO/IEC 1443-1; 2008-06-15.

ISO/IEC, “*Identification cards – Contactless integrated circuit(s) cards – Proximity cards- Part 2: Radio Frequency power and signal interface*”, 1.Edition. ISO/IEC 1443-2; 2001-07-01.
- [3] NFC Forum, “*NFC Forum Home Page* “. <http://www.nfc-forum.org/home>; 2009.
- [4] ECMA, “*Near Field Communication Interface and Protocol (NFCIP-1)*”, 2.Edition. ECMA-340; 2004-12.
- [5] W. Stallings, “*Cryptography and Network Security – Principles and Practices*”, 4.Edition. New Jersey: Pearson Education Inc.; 2006.
- [6] ISO/IEC, “*Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-2)*”, 1.Edition. ISO/IEC 21481; 2004-04-01
- [7] ECMA, “*Near Field Communication Interface and Protocol -2 (NFCIP-2)*”, 1.Edition. ECMA-352; 2003-12.
- [8] NFC FORUM, “*NFC Data Exchange Format (NDEF)*”. NFCForum-TS-NDEF_1.0; 2006-07-24.
- [9] NFC Forum, “*NFC Record Type Definition (RTD)*”. NFCForum-TS-RTD_1.0; 2006-07-24.
- [10] ECMA, “*NFCIP-1 - RF Interface Test Methods*”, 1.Edition. ECMA-356; 2004-6.
- [11] Klaus Finkenzeller, “*RFID Handbook*”, 2.Edition. West Sussex: John Wiley and Sons Ltd.; 2003.
- [12] John D. Kraus and Ronald J. Marhefka, “*Antennas – For All Applications*”, 3.Edition. New York: McGraw-Hill; 2002.
- [13] Wolfgang Rankl and Wolfgang Effing, “*Smart Card Handbook*”, 3.Edition. West Sussex: John Wiley and Sons Ltd.; 2003.
- [14] Gerhard Hancke, “*A Practical Relay Attack on ISO 14443 Proximity Cards*”. Paper from MSc project report submitted to University of Cambridge, Computer Laboratory; 2005-02. <http://www.rfidblog.org.uk/hancke-rfidrelay.pdf>

- [15] Gerhard Hancke, “*Eavesdropping Attacks on High-Frequency RFID Tokens*”. Proceedings of the 4th Workshop on RFID Security (RFID’sec08), pp 100-113; 2008-07.
<http://www.rfidblog.org.uk/Hancke-RFIDsec08-Eavesdropping.pdf>
- [16] Roel Verdult and Gerhard de Koning Gans, “*PROXMARK.org – A Radio Frequency Identification tool*”. <http://www.proxmark.org/proxmark>; 2009-05.
- [17] Edouard Lafargue, “*The "official" Proxmark 3 user's and developer's manual*”. <https://www.lafargue.name/rubrique63.html>; 2009-05.
- [18] proxmark3.com, “*proxmark³ distribution information*”. <http://proxmark3.com/>; 2009-05.
- [19] Nancy Dudney, “*Thin-Film Rechargeable Lithium, Lithium-Ion, and Li-free Batteries*”. Oak Ridge National Laboratory,
<http://www.ms.ornl.gov/researchgroups/Functional/BatteryWeb/index.htm>;
2009-05.
- [20] Smartcard Focus, “*ACR 122U Starter kit product information*”.
<http://www.smartcardfocus.com/shop/ilp/id~344/p/index.shtml>; 2009-05.
- [21] Kay Gorontzi, “*Online CRC Calculation*”.
<https://www.ghsi.de/CRC/index.php?Polynom=10001000000100001&Message=0000>; 2009-05.
- [22] Wikipedia - The Free Encyclopedia, “*Scientific method*”.
http://en.wikipedia.org/w/index.php?title=Scientific_method&oldid=288315093;
2009-05-07.
- [23] ISO/IEC, “*Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 1: Introduction and general model*”, 2.Edition. ISO/IEC 15408-1; 2005-09-22.
- ISO/IEC, “*Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 2: Security functional components*”, 3.Edition. ISO/IEC 15408-2; 2008-08-19.
- ISO/IEC, “*Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 3: Security assurance components*”, 3.Edition. ISO/IEC 15408-3; 2008-08-19.
- [24] ISO/IEC, “*Information technology – Security techniques – Entity authentication – Mechanisms using symmetric encryption algorithms*”, 3.Edition. ISO/IEC 9798-2; 2008-12-15.
- [25] ISO/IEC, “*Information technology – Security techniques – Entity authentication – General*”, 3.Edition. ISO/IEC 9798-1; 1997-08-01.

- [26] BlueCrypt, “*Cryptographic Key Length Recommendation*”. <http://www.keylength.com/>; 2009-05.
- [27] NFC Forum, “*Type 1 Tag Operation Specification*”. NFCForum-TS-Type-1-Tag_1.0; 2007-07-09.

NFC Forum, “*Type 2 Tag Operation Specification*”. NFCForum-TS-Type-2-Tag_1.0; 2007-07-09.

NFC Forum, “*Type 3 Tag Operation Specification*”. NFCForum-TS-Type-3-Tag_1.0; 2007-08-16.

NFC Forum, “*Type 4 Tag Operation Specification*”. NFCForum-TS-Type-4-Tag_1.0; 2007-03-13.
- [28] ACS, “*ACR122U NFC Reader API*”, Version 1.2. <http://www.acs.com.hk/drivers-manual.php?driver=ACR122>; 2008-08.
- [29] ACS, “*ACR122U NFC Reader SDK User Manual*”, Version 1.2. ACR122U Starter Kit CD-ROM; 2008-08.
- [30] ECMA, “*NFC-SEC: NFCIP-1 Security Services and Protocol*”. ECMA-385; 2008-12.
- [31] ECMA, “*NFC-SEC-01: NFC-SEC Cryptography Standard using ECDH and AES*”. ECMA-386; 2008-12.
- [32] ISO, “*International harmonized stage codes*”. http://www.iso.org/iso/standards_development/processes_and_procedures/stages_description/stages_table.htm; 2009-05
- [33] ISO, “*ISO/IEC DIS 13157 Information*”. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53430; 2009-05.
- [34] ISO, “*ISO/IEC DIS 13158 Information*”. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53431; 2009-05
- [35] Wikipedia - The Free Encyclopedia, “*MIFARE*”. <http://en.wikipedia.org/w/index.php?title=MIFARE&oldid=291066811>; 2009-05-20.
- [36] NXP Semiconductors, “*MIFARE Product Information*”. <http://www.mifare.net/>; 2009-05.
- [37] SONY Global, “*FeliCa Product Information*”. <http://www.sony.net/Products/felica/index.html>; 2009-05

- [38] Valtteri Niemi and Kaisa Nyberg, *“UMTS SECURITY”*. West Sussex: John Wiley and Sons Ltd.; 2003.
- [39] Mark Anderson, *“Quirks of RFID Memory Make for Cheap Security Scheme”*. IEEE Spectrum Online: <http://www.spectrum.ieee.org/print/8251>; 2009-03-18.
- [40] Jon Edney and William A. Arbaugh, *“Real 802.11 Security”*. Boston: Pearson Education, Inc.; 2004.
- [41] Wikipedia - The Free Encyclopedia, *“Over-the-air programming”*. http://en.wikipedia.org/w/index.php?title=Over-the-air_programming&oldid=283545740; 2009-04-13.
- [42] ITU-T, *“Formal description techniques (FDT) – Message Sequence Chart (MSC)”*. ITU-T Recommendation Z.120; 2004-04.

APPENDIX A: TEST DOCUMENTATION

This appendix presents detailed documentation and test results which are omitted in the actual report. The screenshots documented here shows all sequences that are used in retrieving the results presented in Chapter 3.5 of the report.

A1. TEST EQUIPMENT SPECIFICATIONS

This section will provide detailed technical information on the NFC related equipment used in this thesis. It is for information purposes only, and is not necessary for understanding any of the performed experiments.

A1.1. ACR 122U

The ACR 122U is a contactless smart card reader supporting NFC and other contactless smart card standards, and has been the basis for the experiment performed in this thesis. Detailed information is presented in the data sheet below [20]:

ACR122 reader standard features:	
<ul style="list-style-type: none"> ● USB Full Speed (12 Mbps) ● USB PnP ● Bi-Color LED ● Built-in Antenna ● NFC Reader <ul style="list-style-type: none"> ○ ISO/IEC18092 (NFC) compliant ○ NFC Tags Access Speed = 424 kbps ● Contactless Smart Card Reader <ul style="list-style-type: none"> ○ Support FeliCa card ○ Support ISO 14443 Type A & B cards <ul style="list-style-type: none"> - MIFARE® cards (Classics, DESFire) ● PC/SC and CCID Compliant ● CE and FCC Certificated ● RoHS Compliant ● User Controllable Buzzer (optional) ● SAM Socket (optional) 	
Typical applications:	
<ul style="list-style-type: none"> ● Home Banking and Home Shopping ● E-commerce ● Checking the balance of electronic purses ● Network access control ● Customer Loyalty Program ● Identification and Authentication ● Ticketing ● Online gaming ● Parking and toll collection ● Automatic Fare Collection ● Public Transportation Terminals ● Physical Access Control ● Time attendance ● Vending machines ● Contactless public phones ● Logistics and supply chain management 	
Technical Specifications:	
Dimensions	98 mm (L) x 65 mm (W) x 12.8 mm (H)
Weight	70 grams
Interface	USB Full Speed
Operating Distance	<= 50mm
Supply Voltage	Regulated 5V DC
Supply Current	200mA (operating); 50mA (standby); 100mA (normal)
Operating Temperature	0-50°C
Operating Frequency	13.56 MHz
Compliance/Certifications	ISO14443 1-4, CE, FCC, RoHS Compliant
Operating System Support	Windows 2000, 2003, XP 32, XP64, Vista 32 and Vista 64

Technical specifications of the ACR 122U contactless reader.

A1.2. NOKIA 6212 Classic

The NOKIA 6212 Classic mobile phone was the only NFC enabled phone available at the Norwegian market at the time equipment was ordered for this thesis. Using the NFC interface, the phone has reading, writing and sharing capabilities. No further technical information is available for this device.

A1.3. Tags and cards

During the experiments performed in this thesis, a number of tags and cards have been exploited. This includes NOKIA NFC tags, MIFARE Classic 1K Card, MIFARE 1K Labels, MIFARE UltraLight Labels and NOKIA 6212 Classic in card emulation mode.

The MIFARE Classic 1K card complies with the ISO/IEC 14443A standard, shall have an operating distance of up to 10 cm and supports authentication. The card is NFC Forum enabled [20]. Further details are shown in the next figure.

The screenshot displays the 'Smartcard Focus' application interface. At the top left is the logo, and at the top right is a 'Close' button. The main content area is titled 'Mifare' and contains several technical specifications:


MiFare, RF Interface (ISO/IEC 14443 A)	
Operating distance	Up to 100mm
Operating frequency	13.56 MHz
Data transfer	106 kbit/s
Data integrity	16 Bit CRC, parity, bit coding, bit counting
Typical ticketing transaction < 100 ms (including backup management)	

EEPROM	
MiFare 1K - 1 Kbyte, organized in 16 sectors with 4 blocks of 16 bytes each (one block consists of 16 bytes)	
MiFare 4k - 4 Kbyte, organised in 32 sectors with 4 blocks and 8 sectors with 16 blocks (one block consists of 16 bytes)	
User definable access conditions for each memory block	
Data retention of 10 years	
Write endurance 100.000 cycles	

Security	
Mutual three pass authentication (ISO/IEC DIS 9798-2)	
Individual set of two keys per sector (per application) to support multi-application with key hierarchy	
Unique serial number for each device	

Manufacturer: [NXP](#)

The MIFARE 1K label uses the same chip as the MIFARE Classic card, has similar operating distance and is also NFC Forum enabled. Details data can be found in the following figure:


Close

Mifare


MiFare, RF Interface (ISO/IEC 14443 A)	
Operating distance	Up to 100mm
Operating frequency	13.56 MHz
Data transfer	106 kbit/s
Data integrity	16 Bit CRC, parity, bit coding, bit counting
Typical ticketing transaction < 100 ms (including backup management)	

EEPROM
MiFare 1K - 1 Kbyte, organized in 16 sectors with 4 blocks of 16 bytes each (one block consists of 16 byte)
MiFare 4k - 4 Kbyte, organised in 32 sectors with 4 blocks and 8 sectors with 16 blocks (one block consists of 16 bytes)
User definable access conditions for each memory block
Data retention of 10 years
Write endurance 100.000 cycles

Security
Mutual three pass authentication (ISO/IEC DIS 9798-2)
Individual set of two keys per sector (per application) to support multi-application with key hierarchy
Unique serial number for each device

Manufacturer: NXP

The MIFARE UltraLight Label is also ISO/IEC14443A compliant, and also NFC Forum type 2 compliant [20]. It offers no security protection. Technical specifications are shown in the figure below.



Mifare UltraLight

MiFare, RF Interface (ISO/IEC 14443 A)	
Operating distance	Up to 100mm
Operating frequency	13.56 MHz
Data transfer	106 kbit/s
Data integrity	16 Bit CRC, parity, bit coding, bit counting
Typical ticketing transaction	<35 ms
Fast counter transaction	<10 ms

EEPROM
512 bit, organised in 16 pages with 4 byte each
Field programmable read-only locking function per page
32 bit user definable One Time Programmable (OTP) area
384 bit user r/w area (12 pages)
Data retention of 5 years
Write endurance 100.000 cycles

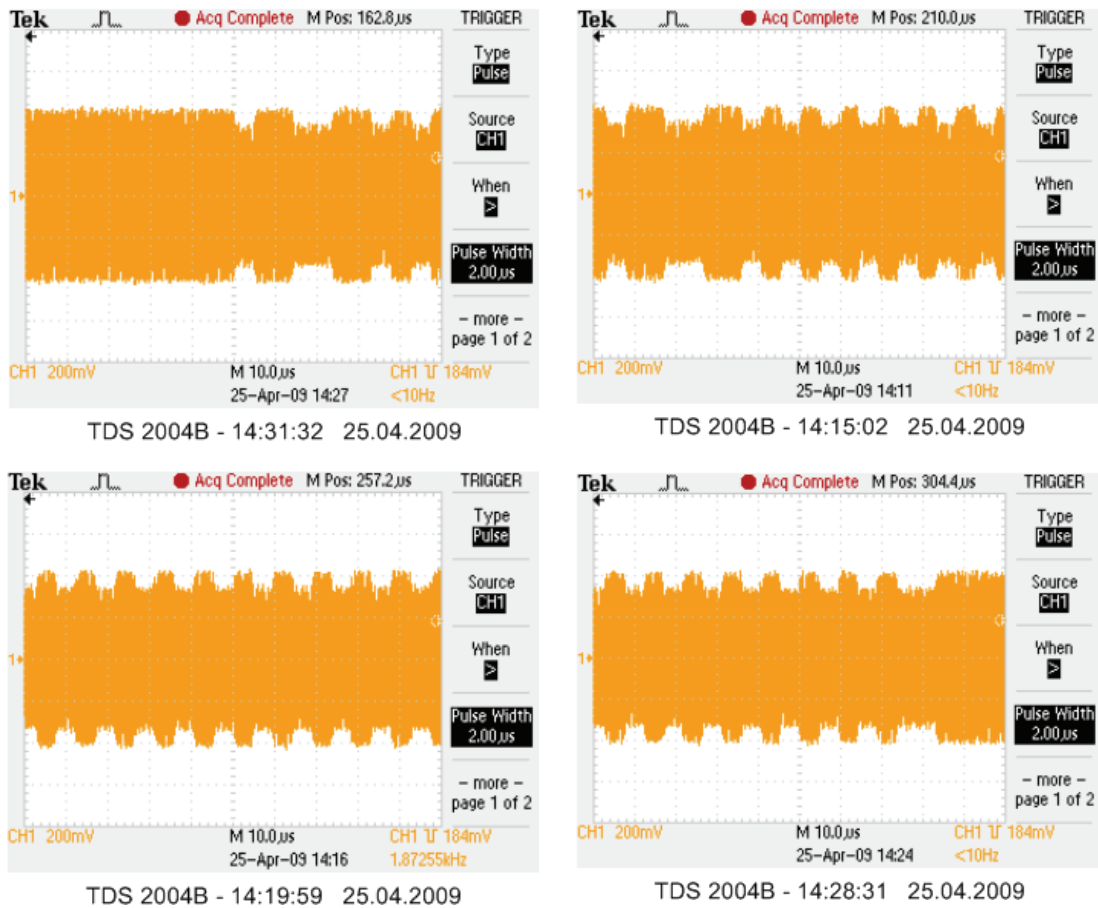
Security
Anti-cloning support by unique 7 Byte serial number for each device
32 Bit user programmable OTP area
Field programmable read-only locking function per page

Manufacturer: [NXP](#)

Technical specifications the NOKIA Tags does not seem to be available. Reading of the tags does however state that they are ISO/IEC 14443 A compliant.

A2. TEST RESULT DOCUMENTATION

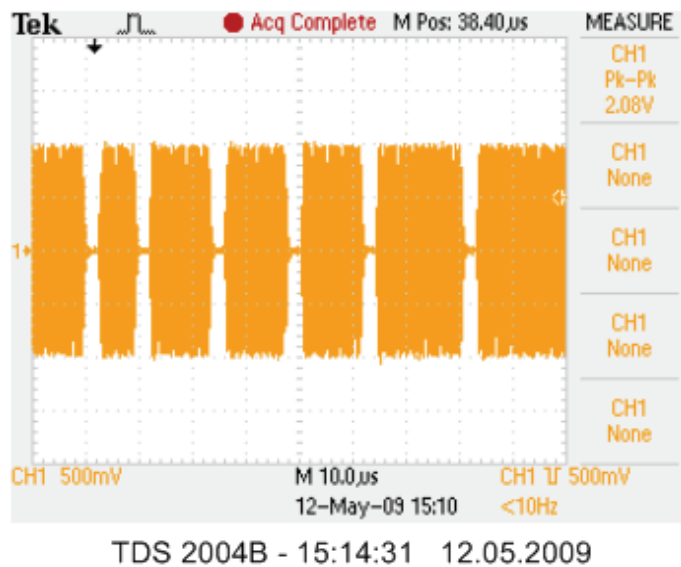
This section shows complete screenshot sections of all commands explained in Chapter 3.5. The picture order is arranged from left to right and then top down.



Complete SENS-RES command for NOKIA 6212 Classic.

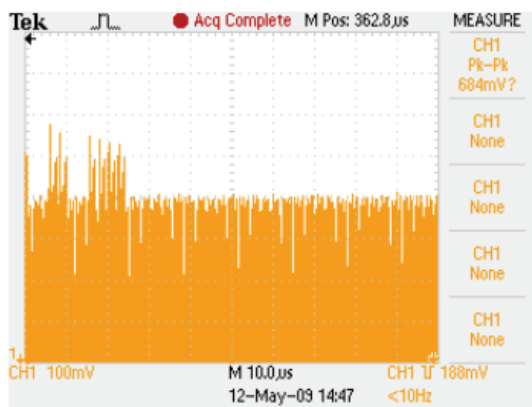
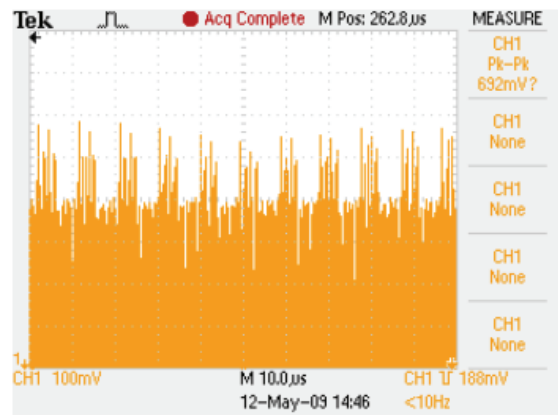
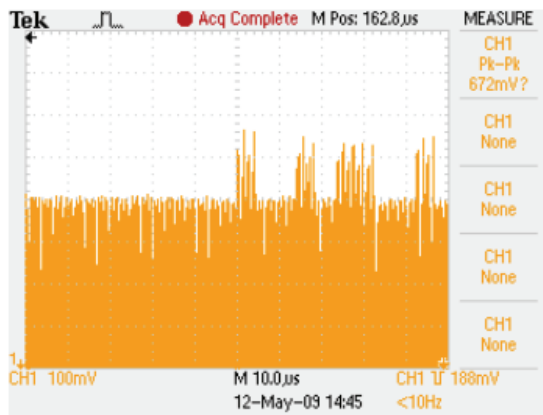
A2.2. Communication sequence recognition

Command 1:



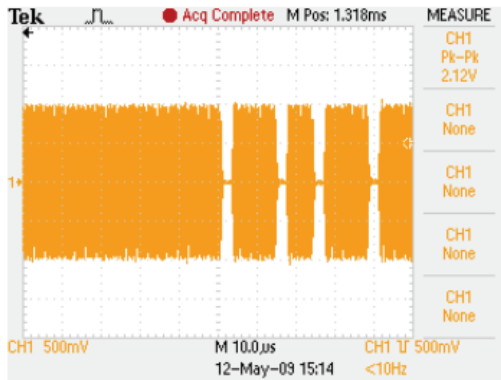
Complete ALL_REQ command.

Command 2:

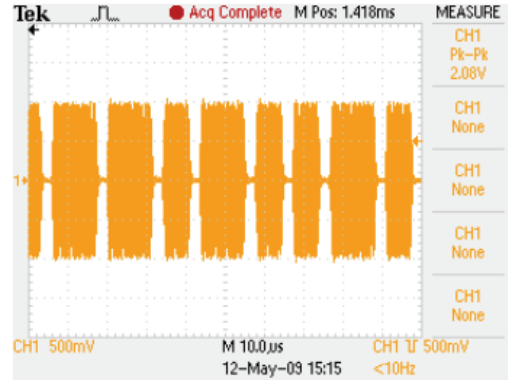


Complete SENS_RES command.

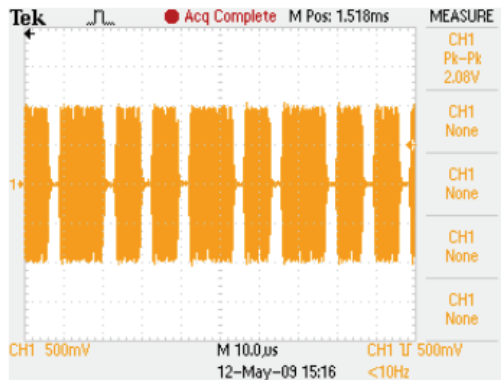
Command 3:



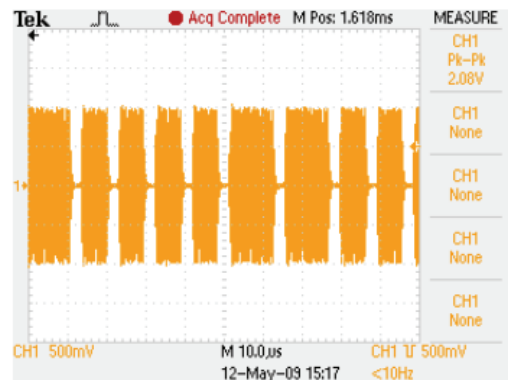
TDS 2004B - 15:18:47 12.05.2009



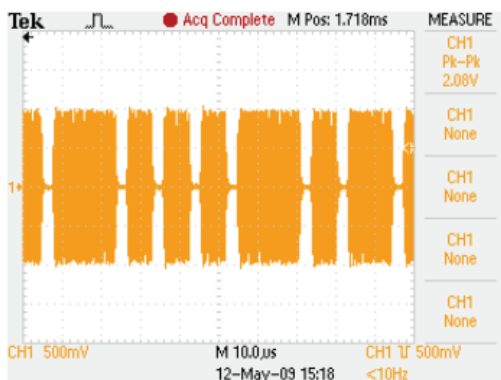
TDS 2004B - 15:19:46 12.05.2009



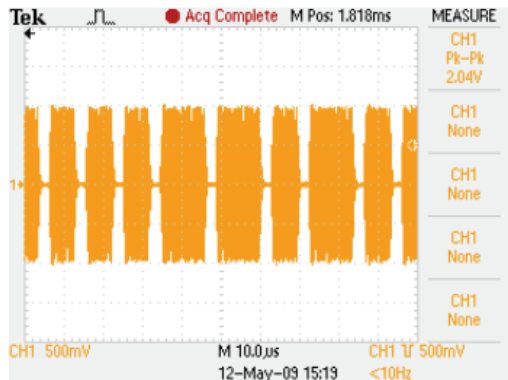
TDS 2004B - 15:20:38 12.05.2009



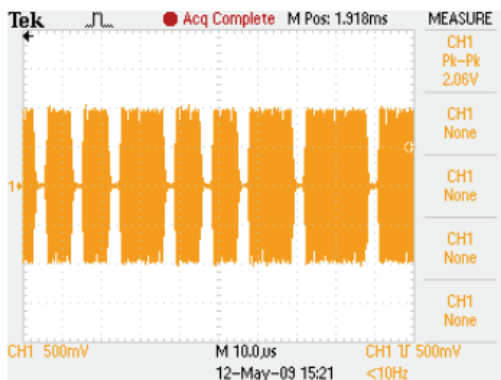
TDS 2004B - 15:21:38 12.05.2009



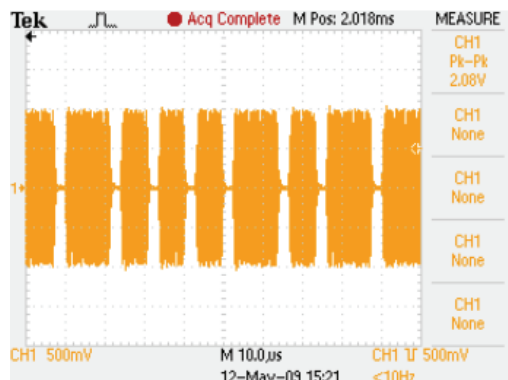
TDS 2004B - 15:22:39 12.05.2009



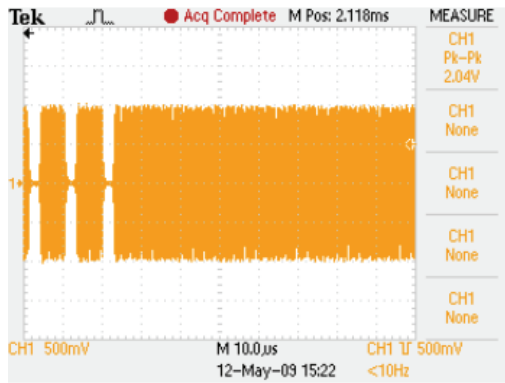
TDS 2004B - 15:23:39 12.05.2009



TDS 2004B - 15:25:02 12.05.2009



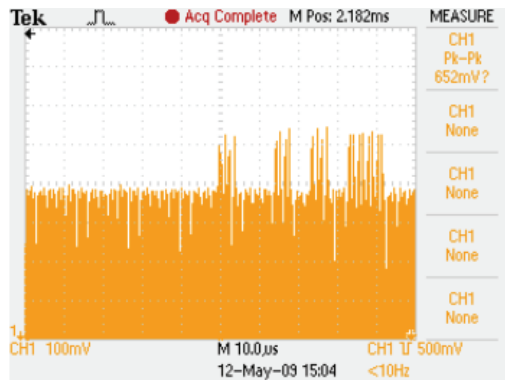
TDS 2004B - 15:25:57 12.05.2009



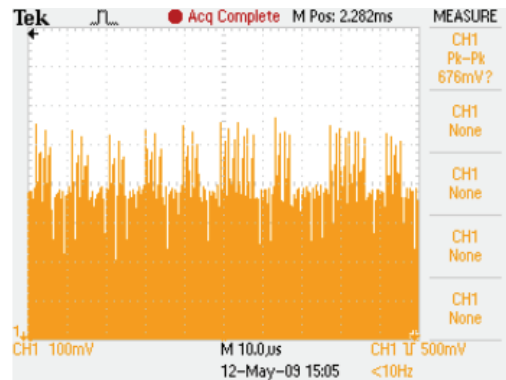
TDS 2004B - 15:27:01 12.05.2009

Complete SEL_REQ command.

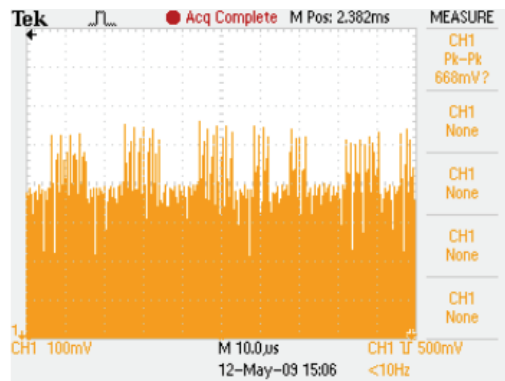
Command 4:



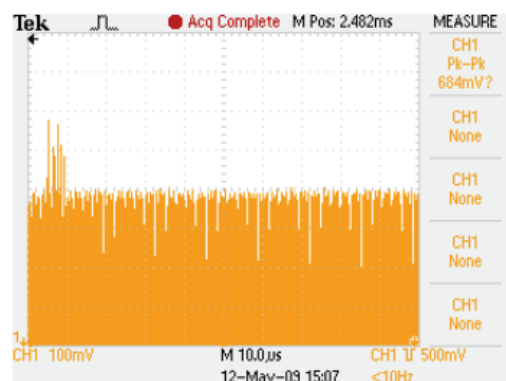
TDS 2004B - 15:08:54 12.05.2009



TDS 2004B - 15:09:46 12.05.2009



TDS 2004B - 15:10:50 12.05.2009



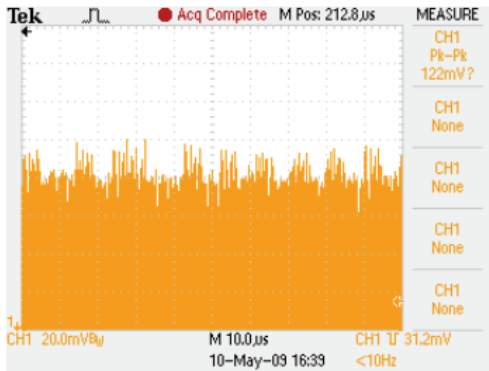
TDS 2004B - 15:11:45 12.05.2009

Complete SEL_RES command.

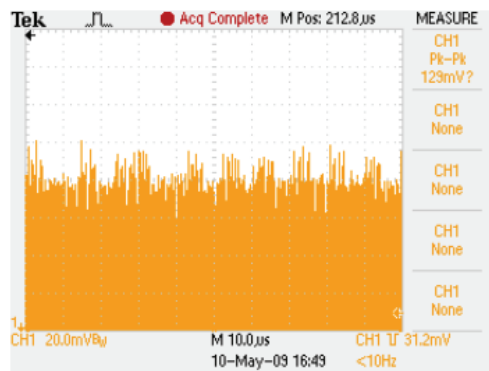
A2.3. Eavesdropping reading range test

Although I have stored three pictures of each measurement in this section, I find it sufficient to present one picture. Each picture presents one test position from 1 until 8, with the pictures arranged from left to right and then top down.

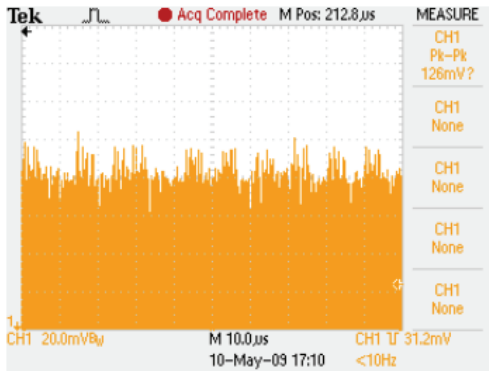
Leveled antenna test:



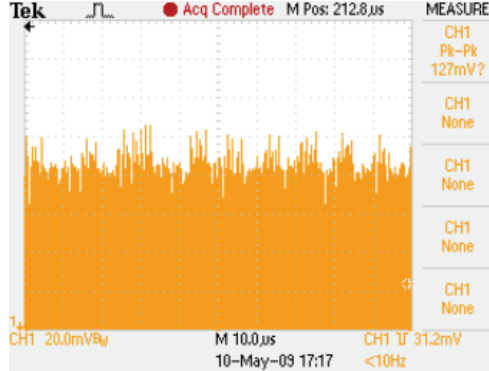
TDS 2004B - 16:43:37 10.05.2009



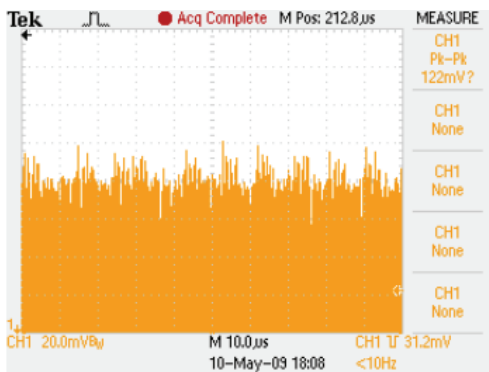
TDS 2004B - 16:53:55 10.05.2009



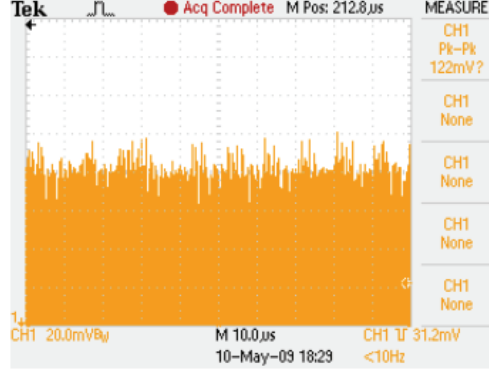
TDS 2004B - 17:14:28 10.05.2009



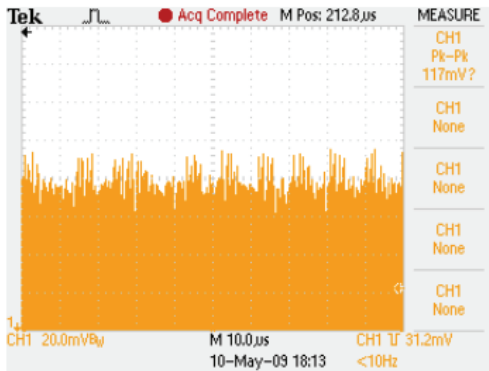
TDS 2004B - 17:21:08 10.05.2009



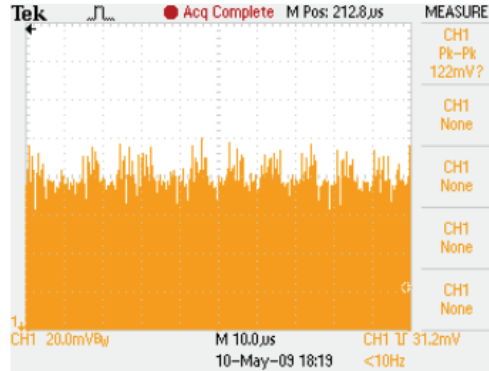
TDS 2004B - 18:12:37 10.05.2009



TDS 2004B - 18:33:31 10.05.2009



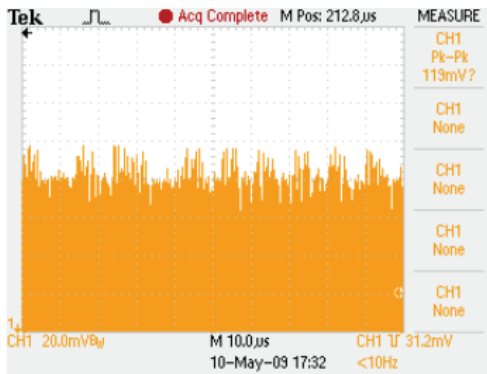
TDS 2004B - 18:17:08 10.05.2009



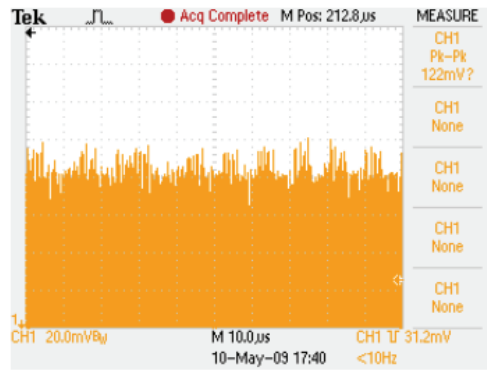
TDS 2004B - 18:23:00 10.05.2009

Test position results for leveled antenna test.

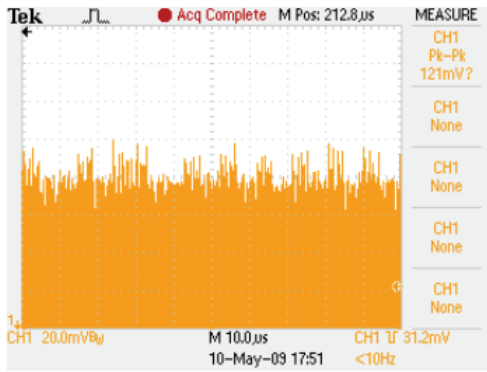
Test with antenna perpendicular to the magnetic field:



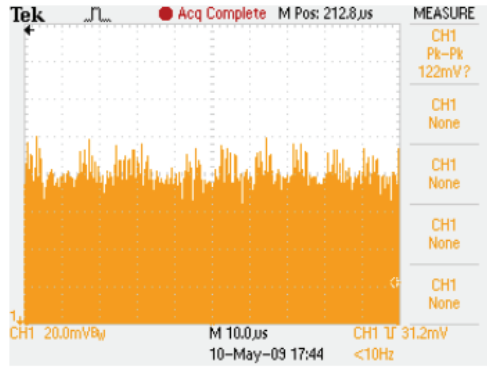
TDS 2004B - 17:36:01 10.05.2009



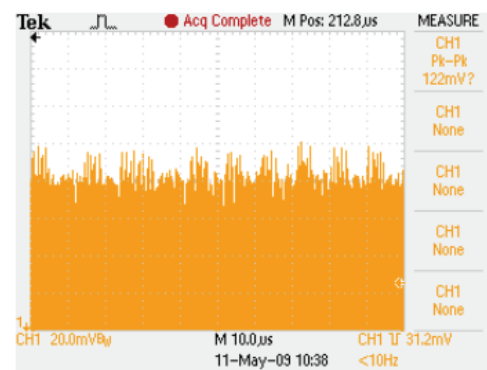
TDS 2004B - 17:44:29 10.05.2009



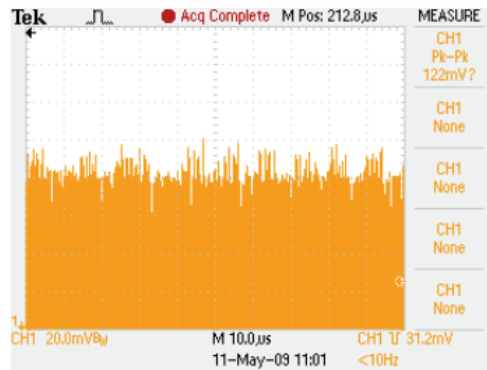
TDS 2004B - 17:55:46 10.05.2009



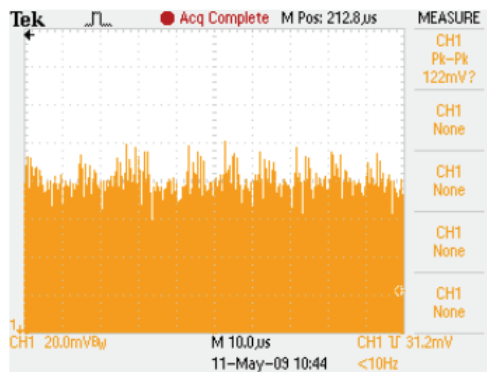
TDS 2004B - 17:48:56 10.05.2009



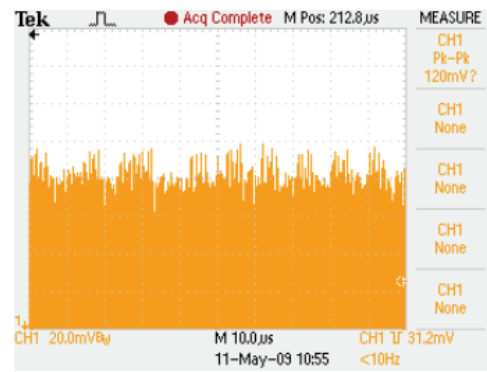
TDS 2004B - 10:42:17 11.05.2009



TDS 2004B - 11:06:40 11.05.2009



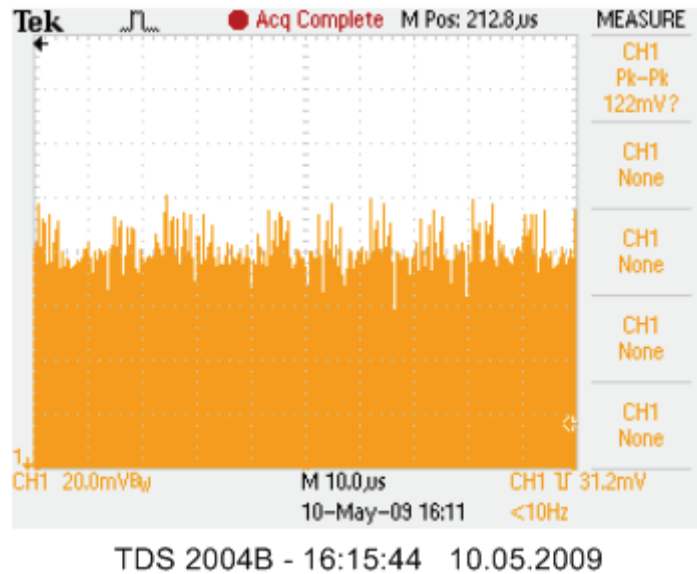
TDS 2004B - 10:48:30 11.05.2009



TDS 2004B - 10:59:02 11.05.2009

Test position results for perpendicular antenna.

Test position number 9 is common for both tests, and is located directly above the centre of the reader. A reading from this position is shown in the next figure.



Test result for position 9.

The pictures shown above are captures of the first 100µs of a SENS_RES command at the maximum eavesdropping range for each test position. If the range is increased from this point, the load modulated signal is impossible to identify using an oscilloscope and the specific antenna made for this thesis.

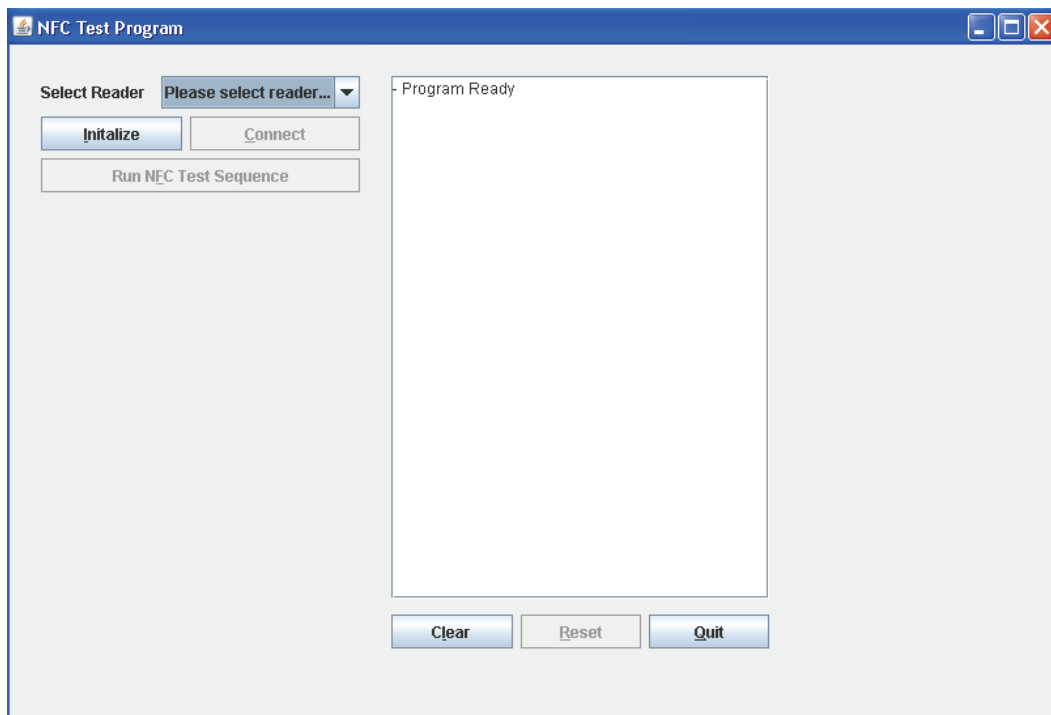
APPENDIX B: TEST-SOFTWARE DOCUMENTATION

B1. NFC Test Program

This program was designed to enable a repetitive and stable communication sequence between the reader and a tag. To be able to trig the oscilloscope, the RF-field had to be turned off whenever no communication sequence is taking place. The program is made to turn on the RF-field, poll the tag, get its UID and finally turn the RF-field off again. The code is written reusing some the example code provided in the SDK together with own commands based on studies of the ACR122U API [28].

B1.1. User manual

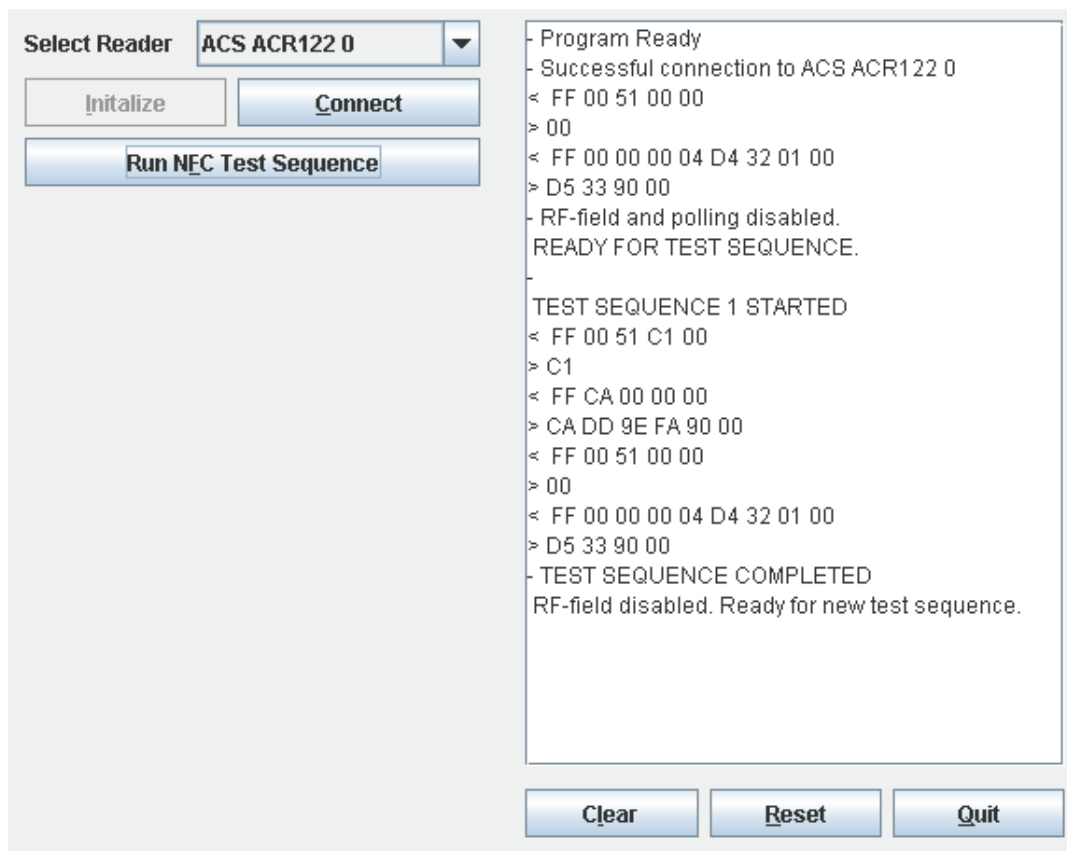
To be able to use the ACR 122U reader, the interface drivers (PC/SC-drivers) have to be installed. These are provided in the SDK, and can also be downloaded at the ACS home page [28]. When this is accomplished, the “NFC Test Program.jar” can be executed. The following GUI will appear:



NFC Test Program GUI

- To initiate contact with the reader, press “Initialize” and the connected reader(s) will appear in the drop down box. Choose the wanted reader if you have connected more than one.

- The next step is to put a tag within the reading range of the reader. The diode will turn green when the tag is readable.
- Press “Connect” to connect to the tag. Now the reader performs a select procedure, and then turns off the RF field. The tag is in sleep mode and the program is ready to perform a test sequence.
- Configure the oscilloscope to capture the desired sequence
- Press “Run NFC Test Sequence”, and one single test sequence is performed. Test sequence number, sent commands, received data and reader status will be displayed in the text window:



GUI information after one performed test sequence

- The test sequence can be repeated as long as the same tag is present within the readers reading range when the “Run”-button is pressed. If the connection is lost, you have to “Reset” and connect to the tag again or restart the whole program.
- The “Clear” button is used to clear the text window.
- Details of the commands shown in the text window can be found in the readers API [28].

B1.2. Source code StartTest.java

All code related to GUI configuration is omitted as it has no relevance for this task. The most important ads made by me can be found from within line 207 and line 330.

```

1. // Based on class files provided in the ACR 122U SDK Starter Kit
2. // Modified by Henning S. Kortvedt, The Norwegian University of Science
and Technology
3. // This class is a modified combination of DevProg.java and
Polling.java. These are example code
4. // provided in the ACR122U SDK.
5. // For educational purposes only.
6. // Their respective Copyrights are listed below:
7.
8. /*=====
=
9. *   Copyright (C) :      Advanced Card Systems Ltd
10.  *
11.  * File:                devProg.java
12.  *
13.  * Description:         This sample program outlines the steps on how
to
14.  *                     set the LED?Buzzer and antenna of the ACR122
NFC reader
15.  *
16.  * Author:              M.J.E.C. Castillo
17.  *
18.  * Date:                June 24, 2008
19.  *
20.  * Revision Trail:     (Date/Author/Description)
21.  *
22.
23.  *=====*/
====
24.   Copyright (C) :      Advanced Card Systems Ltd
25.
26.   File:                Polling.java
27.
28.   Description:         This sample program outlines the steps on how to
29.   execute card detection polling functions using
ACR1222
30.
31.   Author:              M.J.E.C. Castillo
32.
33.   Date:                June 24, 2008
34.
35.   Revision Trail:     (Date/Author/Description)
36.
37.  =====
= */
38.
39.
40.  import java.io.*;
41.  import java.awt.*;
42.  import java.awt.event.*;
43.
44.  import javax.swing.*;
45.  import javax.swing.filechooser.*;

```

```

46.  import javax.swing.filechooser.FileFilter;
47.
48.  public class StartTest extends JFrame implements ActionListener {
49.
50.      //JPCSC Variables
51.      int testNumber;
52.      int retCode;
53.      boolean connActive;
54.
55.      //All variables that requires pass-by-reference calls to
functions are
56.      //declared as 'Array of int' with length 1
57.      //Java does not process pass-by-ref to int-type variables, thus
Array of int was used.
58.      int [] ATRLen = new int[1];
59.      int [] hContext = new int[1];
60.      int [] cchReaders = new int[1];
61.      int [] hCard = new int[1];
62.      int [] PrefProtocols = new int[1];
63.      int [] RecvLen = new int[1];
64.      int SendLen = 0;
65.      int [] nBytesRet =new int[1];
66.      byte [] SendBuff = new byte[262];
67.      byte [] RecvBuff = new byte[262];
68.      byte [] szReaders = new byte[1024];
69.
70.      static String VALIDCHARSHEX = "ABCDEFabcdef0123456789";
71.
72.      //GUI Variables
73.      private JButton bClear, bConn, bInit, bReset, bQuit, bRunTest;
//Added bRunTest
74.      private JComboBox cbReader;
75.      private JLabel lblReader, lblToggle;
76.      private JTextArea mMsg;
77.      private JPanel readerPanel, msgPanel;
78.      private JScrollPane scrPaneMsg;
79.
80.      static JacspcscLoader jacs = new JacspcscLoader();
81.
82.      public StartTest() {
83.          this.setTitle("NFC Test Program");
84.          initComponents();
85.          initMenu();
86.      }
87.
88.      private void initComponents() {
89.          setSize(810,550);
90.          bClear = new JButton();
91.          bConn = new JButton();
92.          bInit = new JButton();
93.          bReset = new JButton();
94.          bQuit = new JButton();
95.          bRunTest = new JButton();
96.          lblReader = new JLabel();
97.          lblToggle = new JLabel();
98.          mMsg = new JTextArea();
99.          readerPanel = new JPanel();
100.         msgPanel = new JPanel();
101.         scrPaneMsg = new JScrollPane();
102.
103.         lblReader.setText("Select Reader");

```



```

104.
105.         String[] rdrNameDef = {"Please select reader
106.     "};
107.         cbReader = new JComboBox(rdrNameDef);
108.         cbReader.setSelectedIndex(0);
109.
110.         bInit.setText("Initalize");
111.         bConn.setText("Connect");
112.         bRunTest.setText("Run NFC Test Sequence");
113.         bInit.setMnemonic(KeyEvent.VK_I);
114.         bConn.setMnemonic(KeyEvent.VK_C);
115.         bRunTest.setMnemonic(KeyEvent.VK_F);
116.         bReset.setMnemonic(KeyEvent.VK_R);
117.         bClear.setMnemonic(KeyEvent.VK_L);
118.         bQuit.setMnemonic(KeyEvent.VK_Q);
119.
120.         bInit.addActionListener(this);
121.         bConn.addActionListener(this);
122.         bRunTest.addActionListener(this);
123.         bReset.addActionListener(this);
124.         bClear.addActionListener(this);
125.         bQuit.addActionListener(this);
126.     }
127.     public void actionPerformed(ActionEvent e) {
128.         if(bInit == e.getSource())
129.         {
130.             //1. Establish context and obtain hContext handle
131.             retCode =
132.             jacs.jSCardEstablishContext(ACSMModule.SCARD_SCOPE_USER, 0, 0, hContext);
133.             if (retCode != ACSModule.SCARD_S_SUCCESS)
134.             {
135.                 mMsg.append("Calling
136.             SCardEstablishContext...FAILED\n");
137.                 displayOut(1, retCode, "");
138.             }
139.             //2. List PC/SC card readers installed in the
140.             system
141.             retCode = jacs.jSCardListReaders(hContext, 0,
142.             szReaders, cchReaders);
143.             int offset = 0;
144.             cbReader.removeAllItems();
145.             for (int i = 0; i < cchReaders[0]-1; i++)
146.             {
147.                 if (szReaders[i] == 0x00)
148.                 {
149.                     cbReader.addItem(new String(szReaders,
150.             offset, i - offset));
151.                     offset = i+1;
152.                 }
153.             }
154.             if (cbReader.getItemCount() == 0)
155.                 cbReader.addItem("No PC/SC reader detected");
156.
157.             cbReader.setSelectedIndex(0);
158.             bConn.setEnabled(true);

```

```

159.             bInit.setEnabled(false);
160.             bClear.setEnabled(true);
161.             bReset.setEnabled(true);
162.
163.             //Look for ACR128 SAM and make it the default
reader in the combobox
164.             for (int i = 0; i < cchReaders[0]; i++)
165.             {
166.                 cbReader.setSelectedIndex(i);
167.
168.                 if (((String)
cbReader.getSelectedItem()).lastIndexOf("ACR122") > -1)
169.                     break;
170.                 else
171.                     cbReader.setSelectedIndex(0);
172.             }
173.         }
174.
175.         if(bConn == e.getSource())
176.         {
177.             if(connActive)
178.             {
179.                 retCode = jacs.jSCardDisconnect(hCard,
ACSMModule.SCARD_UNPOWER_CARD);
180.             }
181.
182.             String rdrcon = (String)cbReader.getSelectedItem();
183.
184.             retCode = jacs.jSCardConnect(hContext,
185.                                         rdrcon,
186.                                         ACSModule.SCARD_SHARE_SHARED,
187.                                         ACSModule.SCARD_PROTOCOL_T1 | ACSModule.SCARD_PROTOCOL_T0,
188.                                         hCard,
189.                                         PrefProtocols);
190.
191.             if (retCode != ACSModule.SCARD_S_SUCCESS)
192.             {
193.                 displayOut(1, retCode, "");
194.                 connActive = false;
195.                 return;
196.             }
197.             else
198.             {
199.
200.                 displayOut(0, 0, "Successful connection to " +
(String)cbReader.getSelectedItem());
201.                 testNumber = 1;
202.             }
203.
204.             String tmpStr="", tmpHex="";
205.
206.
207.             //Set operating parameter
208.             //Disable all polling (command = FF00510000)
209.
210.             clearBuffers();
211.             SendBuff[0] = (byte) 0xFF;

```

```

212.         SendBuff[1] = (byte) 0x00;
213.         SendBuff[2] = (byte) 0x51;
214.         SendBuff[3] = (byte) 0x00;
215.         SendBuff[4] = (byte) 0x00;
216.
217.         SendLen = 5;
218.         RecvLen[0] = 1;
219.
220.         retCode = transmit();
221.
222.         // Disable FR field (command = FF00000004D4320100)
223.
224.         clearBuffers();
225.         SendBuff[0] = (byte) 0xFF;
226.         SendBuff[1] = (byte) 0x00;
227.         SendBuff[2] = (byte) 0x00;
228.         SendBuff[3] = (byte) 0x00;
229.         SendBuff[4] = (byte) 0x04;
230.         SendBuff[5] = (byte) 0xD4;
231.         SendBuff[6] = (byte) 0x32;
232.         SendBuff[7] = (byte) 0x01;
233.         SendBuff[8] = (byte) 0x00;
234.
235.         SendLen = 9;
236.
237.         retCode = transmit();
238.
239.         if (retCode != ACSModule.SCARD_S_SUCCESS)
240.             return;
241.
242.         // Detect successful RF disable : (response = D5 33
243.         // 90 00)
244.         if (RecvBuff[0] == (byte) 0xD5 && RecvBuff[1] ==
245.             (byte) 0x33 && RecvBuff[2] == (byte) 0x90 && RecvBuff[3] == (byte) 0x00){
246.             displayOut(0, 0, "RF-field and polling
247. disabled. \n READY FOR TEST SEQUENCE.");
248.         }
249.         else{
250.             displayOut(0, 0, "FAILED to disable RF-field!
251. \n Restart reader and program");
252.         }
253.
254.         //add buttons
255.         connActive=true;
256.         bRunTest.setEnabled(true);
257.     }
258.
259.     // Run test sequence
260.     if(bRunTest == e.getSource())
261.     {
262.         displayOut(0, 0, "\n TEST SEQUENCE " + testNumber
263. + " STARTED");
264.         testNumber ++;
265.
266.         // Activate polling for MIFARE Type A
267.         clearBuffers();
268.         SendBuff[0] = (byte) 0xFF;
269.         SendBuff[1] = (byte) 0x00;
270.         SendBuff[2] = (byte) 0x51;
271.         SendBuff[3] = (byte) 0xC1;

```

```

268.         SendBuff[4] = (byte) 0x00;
269.
270.         SendLen = 5;
271.         RecvLen[0] = 1;
272.
273.         retCode = transmit();
274.
275.         //Send "Get UID"
276.         clearBuffers();
277.         SendBuff[0] = (byte) 0xFF;
278.         SendBuff[1] = (byte) 0xCA;
279.         SendBuff[2] = (byte) 0x00;
280.         SendBuff[3] = (byte) 0x00;
281.         SendBuff[4] = (byte) 0x00;
282.
283.         SendLen = 5;
284.         RecvLen[0] = 1;
285.
286.         retCode = transmit();
287.
288.         //Start deactivation of reader after successful
test sequence
289.         //Set operating parameter
290.         //Disable all polling
291.         clearBuffers();
292.         SendBuff[0] = (byte) 0xFF;
293.         SendBuff[1] = (byte) 0x00;
294.         SendBuff[2] = (byte) 0x51;
295.         SendBuff[3] = (byte) 0x00;
296.         SendBuff[4] = (byte) 0x00;
297.
298.         SendLen = 5;
299.         RecvLen[0] = 1;
300.
301.         retCode = transmit();
302.
303.         // Disable FR field
304.
305.         clearBuffers();
306.         SendBuff[0] = (byte) 0xFF;
307.         SendBuff[1] = (byte) 0x00;
308.         SendBuff[2] = (byte) 0x00;
309.         SendBuff[3] = (byte) 0x00;
310.         SendBuff[4] = (byte) 0x04;
311.         SendBuff[5] = (byte) 0xD4;
312.         SendBuff[6] = (byte) 0x32;
313.         SendBuff[7] = (byte) 0x01;
314.         SendBuff[8] = (byte) 0x00;
315.
316.         SendLen = 9;
317.
318.         retCode = transmit();
319.
320.         if (retCode != ACSModule.SCARD_S_SUCCESS)
321.             return;
322.
323.         // Detect successful disable : (response = D5 33 90
00)
324.         if (RecvBuff[0] == (byte) 0xD5 && RecvBuff[1] ==
(byte) 0x33 && RecvBuff[2] == (byte) 0x90 && RecvBuff[3] == (byte) 0x00){

```

```

325.             displayOut(0, 0, "TEST SEQUENCE COMPLETED \n
RF-field disabled. Ready for new test sequence.");
326.         }
327.
328.             else{
329.                 displayOut(0, 0, "FAILED to disable RF-field!
/n Restart reader and program");
330.             }
331.         }
332.         if(bClear == e.getSource())
333.         {
334.             mMsg.setText("");
335.         }
336.
337.         if(bQuit == e.getSource())
338.         {
339.             this.dispose();
340.         }
341.
342.         if(bReset==e.getSource())
343.         {
344.             //disconnect
345.             if (connActive){
346.                 retCode = jacs.jSCardDisconnect(hCard,
ACSMModule.SCARD_UNPOWER_CARD);
347.                 connActive= false;
348.             }
349.
350.             //release context
351.             retCode = jacs.jSCardReleaseContext(hContext);
352.             //System.exit(0);
353.
354.             mMsg.setText("");
355.             initMenu();
356.             cbReader.removeAllItems();
357.             cbReader.addItem("Please select reader      ");
358.         }
359.     }
360.
361.     public int transmit()
362.     {
363.         ACSModule.SCARD_IO_REQUEST IO_REQ = new
ACSMModule.SCARD_IO_REQUEST();
364.         ACSModule.SCARD_IO_REQUEST IO_REQ_Recv = new
ACSMModule.SCARD_IO_REQUEST();
365.         IO_REQ.dwProtocol = PrefProtocols[0];
366.         IO_REQ.cbPciLength = 8;
367.         IO_REQ_Recv.dwProtocol = PrefProtocols[0];
368.         IO_REQ_Recv.cbPciLength = 8;
369.         RecvLen[0] = 262;
370.
371.         String tmpStr, tmpHex="";
372.         tmpStr = "";
373.
374.         for(int i=0; i<SendLen; i++)
375.         {
376.             tmpHex =
Integer.toHexString(((Byte)SendBuff[i]).intValue() & 0xFF).toUpperCase();
377.
378.             //For single character hex
379.             if (tmpHex.length() == 1)

```

```

380.             tmpHex = "0" + tmpHex;
381.
382.             tmpStr += " " + tmpHex;
383.         }
384.
385.         displayOut(2, 0, tmpStr);
386.
387.         retCode = jacs.jSCardTransmit(hCard,
388.                                     IO_REQ,
389.                                     SendBuff,
390.                                     SendLen,
391.                                     null,
392.                                     RecvBuff,
393.                                     RecvLen);
394.
395.         if (retCode != ACSModule.SCARD_S_SUCCESS)
396.         {
397.             displayOut(1, retCode, "");
398.         }
399.         else
400.         {
401.             tmpStr = "";
402.
403.             for(int i =0; i<RecvLen[0]; i++)
404.             {
405.                 tmpHex =
Integer.toHexString(((Byte)RecvBuff[i]).intValue() & 0xFF).toUpperCase();
406.
407.                 //For single character hex
408.                 if (tmpHex.length() == 1)
409.                     tmpHex = "0" + tmpHex;
410.
411.                 tmpStr += " " + tmpHex;
412.             }
413.             displayOut(3, 0, tmpStr.trim());
414.         }
415.
416.         return retCode;
417.     }
418.
419.     public void displayOut(int mType, int msgCode, String
printText)
420.     {
421.         switch(mType)
422.         {
423.             case 1:
424.                 {
425.                     mMsg.append("! " + printText);
426.
mMsg.append(ACSModule.GetScardErrMsg(msgCode) + "\n");
427.                     break;
428.                 }
429.             case 2: mMsg.append("< " + printText + "\n");break;
430.             case 3: mMsg.append("> " + printText + "\n");break;
431.             default: mMsg.append("- " + printText + "\n");
432.         }
433.
434.     }
435.
436.     public void clearBuffers()
437.     {

```

```

438.         for(int i=0; i<262; i++)
439.         {
440.             SendBuff[i]=(byte) 0x00;
441.             RecvBuff[i]= (byte) 0x00;
442.         }
443.     }
444.
445.     public void keyReleased(KeyEvent ke) {
446.     }
447.     public void keyPressed(KeyEvent ke) {
448.         //restrict paste actions
449.         if (ke.getKeyCode() == KeyEvent.VK_V )
450.             ke.setKeyCode(KeyEvent.VK_UNDO );
451.     }
452.
453.     public void initMenu()
454.     {
455.         connActive = false;
456.         bConn.setEnabled(false);
457.         bInit.setEnabled(true);
458.         bRunTest.setEnabled(false);
459.         bReset.setEnabled(false);
460.
461.         displayOut(0, 0, "Program Ready");
462.     }
463.
464.     public static void main(String args[]) {
465.         EventQueue.invokeLater(new Runnable() {
466.             public void run() {
467.                 new StartTest().setVisible(true);
468.             }
469.         });
470.     }
471. }

```

B2. CRC Calculation Program

This program is designed to check the CRC of one or two data bytes, according to the CRC computations standardized in ISO/IEC 18092 [1]. The purpose is to be able to check that the eavesdropped CRC bytes agree to the CRC computed on the eavesdropped data bytes. This will contribute in verifying that the eavesdropping is accurate.

B2.1. User manual

- Initiate the program by running the CRCCalculationProgram.exe in a command window.
- Enter the desired one or two data bytes with lsb first per byte. Remember to exclude all parity bits and start/end delimiters.

```

C:\Documents and Settings\bruker>CRCCalculationProgram.exe
Enter bitstring to calculate (lsb first):00010000
CRC of [00010000] is [011011011011011] with P=[10001000000100001]

```

- The program shows the inserted data, the 16 bit CRC byte wise with lsb per byte. In the end it indicates the polynomial used in the calculation, where each '1' indicates an x in the power of the bit position. In this case it is permanently set to $X^{15}+X^{11}+X^5+1$.
- The .exe-file has to be executed for each calculation.

B2.2. Source code for CRCCalculationProgram.c

```
//
=====
// CRC Generation Unit - Linear Feedback Shift Register implementation
// (c) Kay Gorontzi, GHSi.de, distributed under the terms of LGPL
//
=====

//
=====
// Modified by Henning S. Kortvedt spring 2009 for use in my Masters Thesis
// at NTNU, Norway. For educational purposes only.
//
=====

char *MakeCRC(char *BitString)
{
    static char Res[17];           // CRC Result
    char CRC[16];
    char ff[16];                  // Added by Henning S. Kortvedt
    int i;
    char DoInvert;

    ff[0] = 0;                   // Added by Henning S. Kortvedt
    ff[1] = 1;                   // Set initial value of the sift register
    ff[2] = 1;
    ff[3] = 0;
    ff[4] = 0;
    ff[5] = 0;
    ff[6] = 1;
    ff[7] = 1;
    ff[8] = 0;
    ff[9] = 1;
    ff[10] = 1;
    ff[11] = 0;
    ff[12] = 0;
    ff[13] = 0;
    ff[14] = 1;
    ff[15] = 1;

    for (i=0; i<16; ++i) CRC[i] = ff[i];           // Init before calculation
}
```



```

// Modified By henning S. Kortvedt to
// set initial value instead of 15 zeros
// "crc[i] = 0" replaced by "CRC[i] = ff[i]"

for (i=0; i<strlen(BitString); ++i)
{
    DoInvert = ('1'==BitString[i]) ^ CRC[15];    // XOR required?

    CRC[15] = CRC[14];
    CRC[14] = CRC[13];
    CRC[13] = CRC[12];
    CRC[12] = CRC[11] ^ DoInvert;
    CRC[11] = CRC[10];
    CRC[10] = CRC[9];
    CRC[9] = CRC[8];
    CRC[8] = CRC[7];
    CRC[7] = CRC[6];
    CRC[6] = CRC[5];
    CRC[5] = CRC[4] ^ DoInvert;
    CRC[4] = CRC[3];
    CRC[3] = CRC[2];
    CRC[2] = CRC[1];
    CRC[1] = CRC[0];
    CRC[0] = DoInvert;
}

for (i=0; i<16; ++i) Res[15-i] = CRC[i] ? '1' : '0'; // Convert binary to ASCII
Res[16] = 0;                                     // Set string terminator

return(Res);
}

// A simple test driver:

#include <stdio.h>

int main()
{
    char *Data, *Result, *Input;                // Declare two strings - added *Input
    printf("Enter bitstring to calculate (lsb first):"); // Added by Hening S. Kortvedt. Input
    scanf("%s", Input );                       // of data bits to calculate

    Data = Input;                               // Data to compute - added "= Input"
    Result = MakeCRC(Data);                     // Calculate CRC

    printf("CRC of [%s] is [%s] with P=[10001000000100001]\n", Data, Result);

    return(0);
}

```


APPENDIX C: PROJECT PLAN

C1. INTRODUCTION

C1.1. Background

NFC is a new technology which aims at integrated services for mobile phones. The development of SIM-cards and mobile phones has made them powerful enough to store and run third party applications. NFC is a short range communication channel based on magnetic field induction in the receiving antenna.

C1.1. Project goal

The goal for this thesis is to shape a security protocol for NFC, so that it can be used in applications exchanging sensitive information such as replacement of credit cards and public transport payment. I will use physical experiments to prove that the channel can be eavesdropped, in order to visualize the need for a secure channel solution.

C1.2. Limitations

The task will be solved within the constraints given by NTNU and the responsible professor, together with the evaluations the student should find necessary in order to complete the task in the best possible manner.

C2. SCOPE AND DELIMITATIONS

The outline for this task is presented by Prof. Stig Frode Mjøl̄snes, but the student will be relatively free in deciding the direction of the work. Although, frequent meetings between the professor and the student are supposed to assure that the work stays within the intentions of the thesis suggestion.

I have chosen to focus on a communication model where an NFC mobile phone acts as a passive tag, to assure that the work load doesn't exceed the time available. This is also what I believe will be the most used form of communication, as it's compatible with previous systems like MIFARE.

A crucial point of this task is to succeed in proving the easiness of listening into the communication channel. If this part of the work fails I'll either have to assume that eavesdropping is possible and refer to this assumption in the security solution proposal, or try to prove that eavesdropping is impossible and if so state that the properties of NFC itself offers sufficient security. This implies that a lot of work will be put into the experiment until it's successful.

C3. PROJECT ORGANIZING

C3.1. Project process

The project shall present a security protocol for NFC communication, so that sensitive applications can safely exchange data between NFC devices. To accomplish this, relevant literature and standards must be studied. To eliminate any doubts around the security of NFC today, I will try to prove that a passive antenna for eavesdropping easily can be made.

C3.2. Project management

This project is managed through the IME-faculty at NTNU, and is composed by Prof. Stig Frode Mjøl̄snes. He will acts as both teaching supervisor and responsible professor representing the faculty.

C3.3. Other roles

Roles and responsibility	Description
Project responsible(PR)	Henning S. Kortvedt
Project manager(PM)	Stig Frode Mjøl̄snes
Project responsible, contractor	Stig Frode Mjøl̄snes
Project associates	None
Functional responsible	Henning S. Kortvedt

C4. DECISION STAGES, FOLLOW-UP AND MILESTONES

C4.1. Decision stages

Decision stages	Approver	Execution responsible	Deadline	Accomplished
Submission of Master's thesis contract	PM	PR	15/1	13/1
Submission of problem text	PM	PM	15/2	13/2
Submission of Master's thesis	PM	PM	10/6	4/6

C4.2. Follow-up

Usually a project has high frequency meetings for work supervision and some meetings with PM to check the main direction for the project progress. As the professor acts both as PM and teaching supervisor, the two types of meetings will merge.

C4.2.1. Project meetings

- Purpose: Progress direction control, status reporting, academic quality assurance, consultation, presentation and final approval.
- Participants: PR and PM
- Frequency: Every Tuesday. (If necessary)
- Reports: None

C4.3. Milestones

Date	Description
13/1	Project start
15/2	Submission of problem text
16/3	Start of experiment
31/4	End of experiment
6/6	Report finished and ready for correction
8/6	Report ready for printing
8/6	Submission of report

C5. CARRYING OUT

C5.1. Main activities

This project will contain both theoretical work and physical experiments with NFC equipment. The work starts with a comprehensive literature study, followed by a presentation of theory relevant for this thesis. Then I will plan and carry out the experiment to prove that it is possible to listen into the communication channel. The biggest challenge here will be to make a compatible antenna. Finally I will present the best possible security solution for NFC communication based on the presented theory and the test results.

C5.2. Time and resource schedule

Activities	Time schedule (hours)						Hours totally
	Jan	Feb	Mar	Apr	May	Jun	
Forming of problem text	5	5					10
Literature study	100	100	50				250
Report writing		25	20	10	250	40	345
Experiment planning		30	30				60
Experiment work			80	150			230
Project meetings	2	2	2	1	2	1	10
Sum hours:							905

C5.3. Tools

Tools that will be used during the work are:

- Microsoft Office Word 2007 to write documentation
- Microsoft Office Visio to make illustrations
- ECLIPSE to write JAVA-code
- “LabVIEW SignalExpress Tektronix Edition” and “Tektronix OpenChoice Desktop Application” to document oscilloscope measurements
- Minimalist GNU for Windows (MinGW) to compile C-code